Microsoft

Power Automate Bootcamp: Basic

Hands-On Lab Guide

Published (Jan 2025)

Disclaimer

The content, code samples, and guidance provided in this lab document (the "Hands-On Lab Guide") are for informational and educational purposes only. The Materials are provided "as is" and reflect general practices and platform capabilities at the time of publication. While reasonable efforts have been made to ensure accuracy, no representation or warranty, express or implied, is made as to the completeness, accuracy, reliability, or suitability of the Materials for any purpose.

The Materials are not intended to be used as-is in production environments. It is the sole responsibility of the user to evaluate and test all solutions, configurations, and code examples in a controlled, non-production environment prior to any implementation in a live setting. Use of these Materials in a production environment is done entirely at the user's own risk.

No warranties or guarantees of any kind are provided, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or the absence of errors or defects. Under no circumstances shall the authors, contributors, or any affiliated parties be held liable for any damages, including but not limited to direct, indirect, incidental, consequential, or special damages, arising out of or in connection with the use of the Materials.

By using or adapting any portion of the lab document in any capacity, including in production environments, you expressly acknowledge and agree that you are assuming full responsibility and liability for such use, and that you release the authors and affiliated parties from any and all claims or liabilities that may arise as a result.

Microsoft Power Platform

Contents

Disclaimer 2

Exercise 1: Experimenting Advanced Power Automate Settings 1

- Objectives 1
- Estimated Time 1
- Task 1: Uploading Dummy Excel File to OneDrive 1
- Task 2: Creating a Manually Triggered Flow 2
- Task 3: Pagination and Threshold settings 3
- Task 4: Concurrency Settings 6
- Task 5: Run after Settings 9

Exercise 2: Create Approval Workflow to Run from Power Apps 13

- Objectives 13
- Estimated Time 13
- Task 1: Creating Environment Variables 14
- Task 2: Creating a New Flow to Run from Power App 18
- Task 3: Handling Approval Timeout 24
- Task 4: Adding Flow to App 27

Exercise 3: Deep Linking for Direct Access 30

- Objectives 30
- Estimated Time 30
- Task 1: Duplicate the existing Screen 31
- Task 2: Handle Deep Link ID in App 33
- Task 2: Include link in Power Automate Approval 36

Exercise 4: Utilising Adaptive Card and Child Flow 37

- Objectives 37
- Estimated Time 37
- Task 1: Change Response Options in Approval Workflow 38
- Task 2: Design Adaptive card to Get User Input 41
- Task 3: Create a Child Flow to Get More Information 45
- Task 4: Add Child Flow to the Main Flow 50

Exercise 1: Experimenting Advanced Power Automate Settings

In this hands-on exercise, participants will create an **instant cloud flow** to deal with large excel data and play with the stage settings to be able to deal with it.

Objectives

After completing this exercise, participants will be able to:

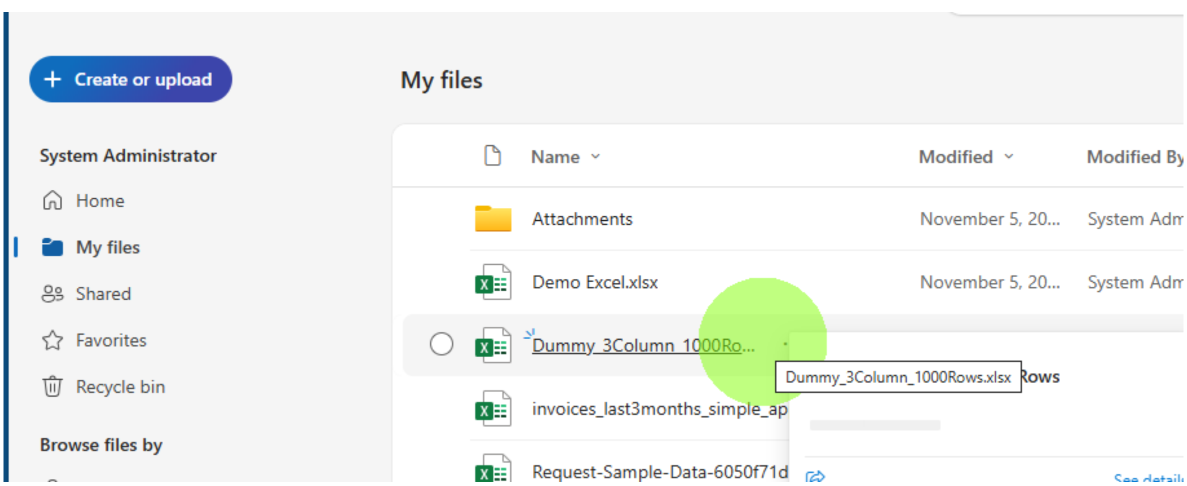
- Have a better understanding of Advanced Power Automate stage settings
- Perform Error Handling in Power Automate

Estimated Time

40–50 mins

Task 1: Uploading Dummy Excel File to OneDrive

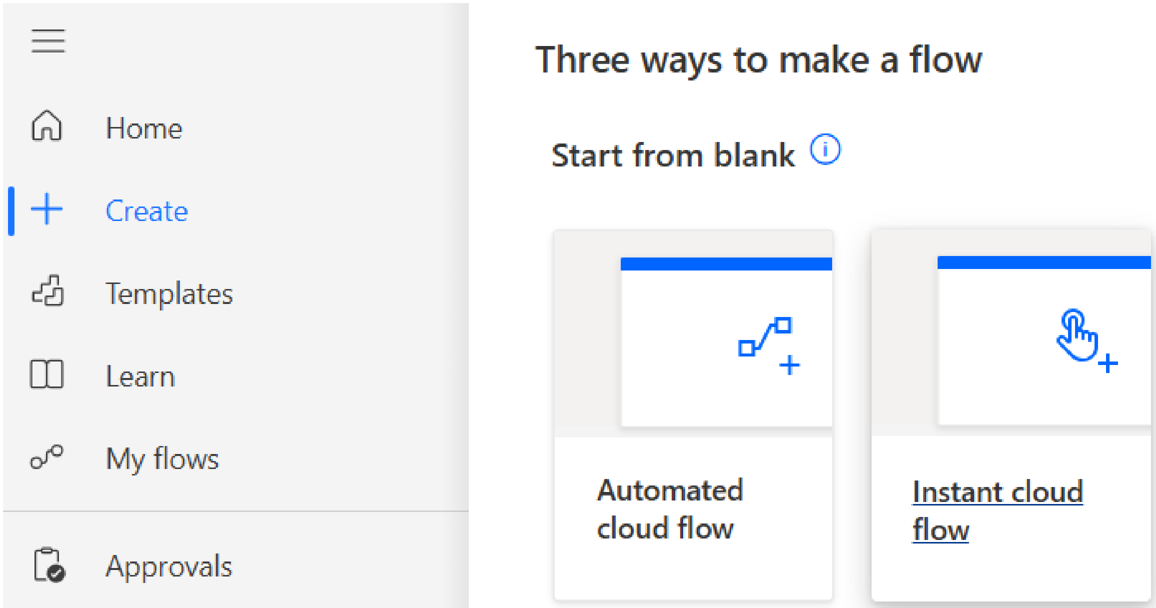
1. Go to **OneDrive for Business** and upload Dummy_3Column_1000Rows.xlsx file.



-- **End of Task** --

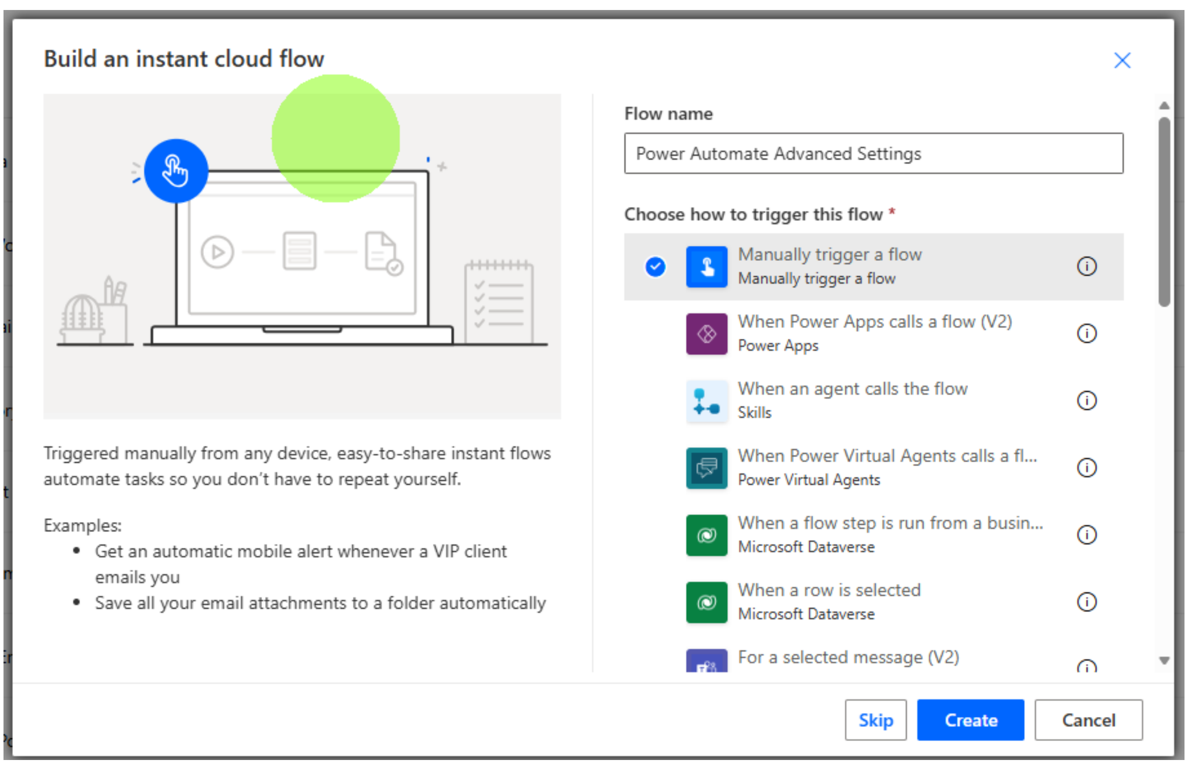
Task 2: Creating a Manually Triggered Flow

1. Go to Power Automate and click **Create > Instant cloud flow**



2. Name your flow: **Power Automate Advanced Settings**

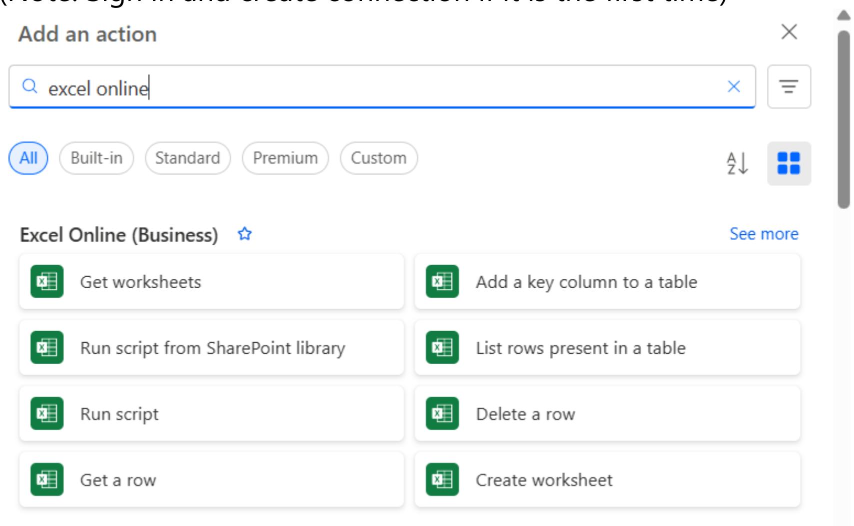
Choose the trigger: **Manually trigger a flow**, then click **Create**

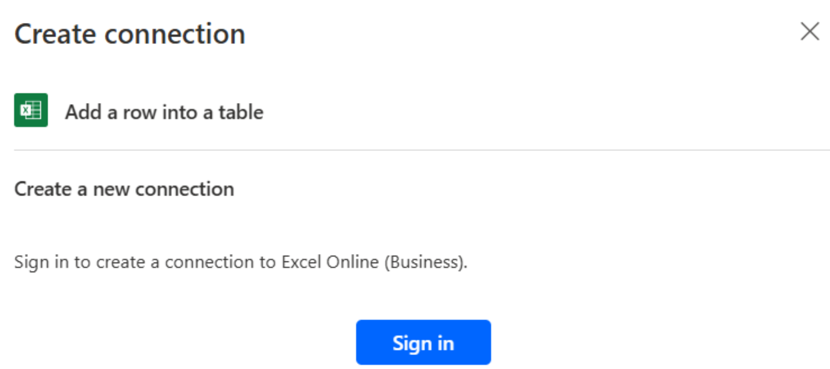


-- **End of Task** --

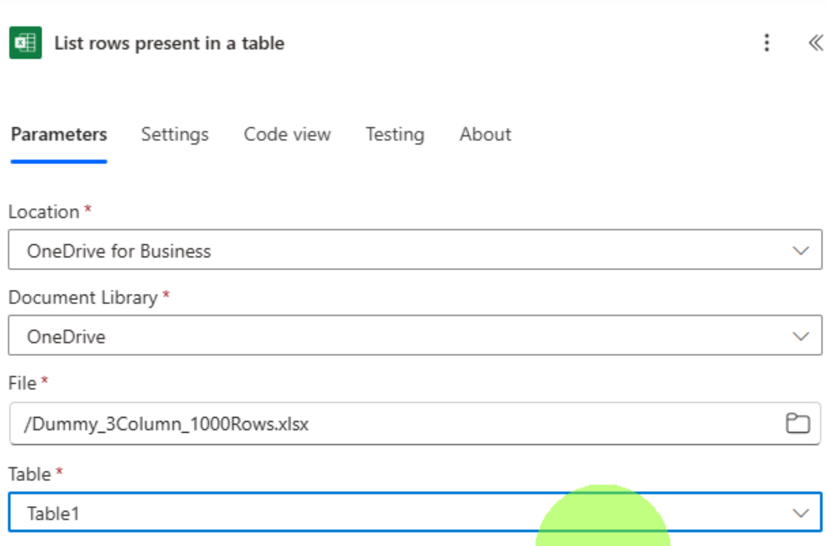
Task 3: Pagination and Threshold settings

1. Click **+ New Step**, search for **Excel Online (Business)**, and choose **List rows present in a table** (Note: Sign in and create connection if it is the first time)



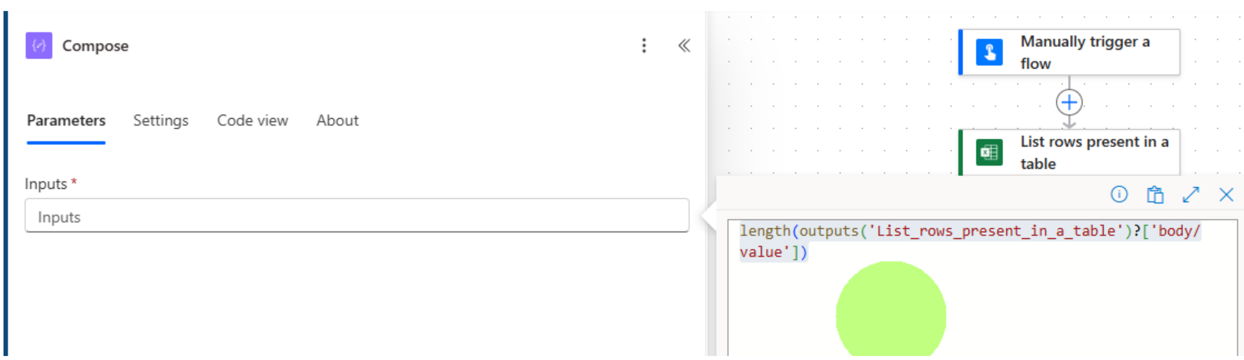


2. Select your OneDrive Excel file and the table name as shown below

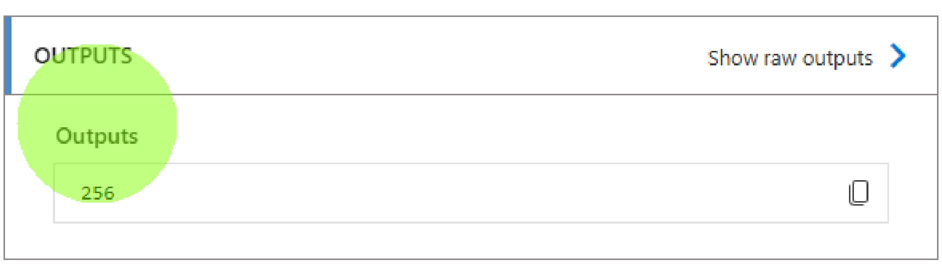


3. Add a compose stage with below formula to observe the number of rows read.

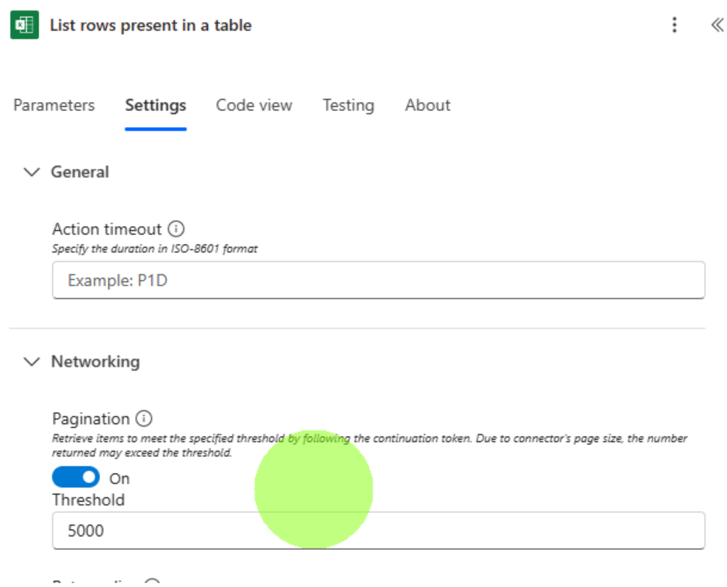
```
length(outputs('List_rows_present_in_a_table')?['body/value'])
```



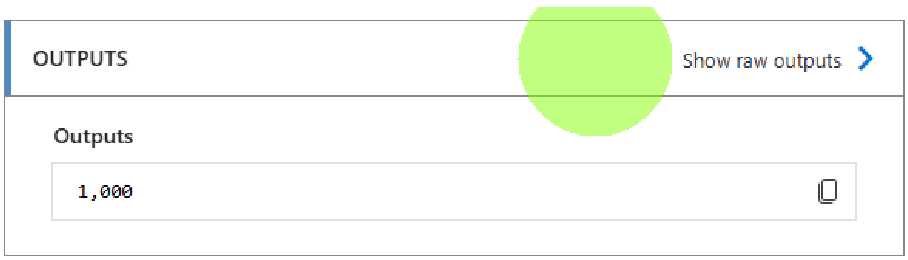
4. You can save and test the flow and notice in Compose stage output that it doesn't read all the rows from the file. That's due to Power Automate default settings to retrieve 256 rows.



5. Select List rows present in a table stage and switch to Settings tab. Turn on Pagination and set Threshold to 5000.



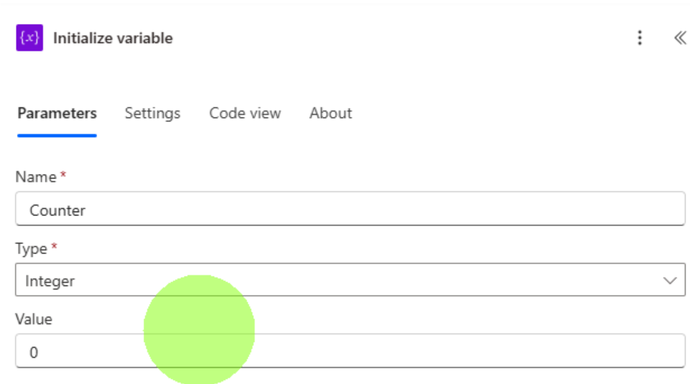
6. You can run again and will notice that it is now returning full list.



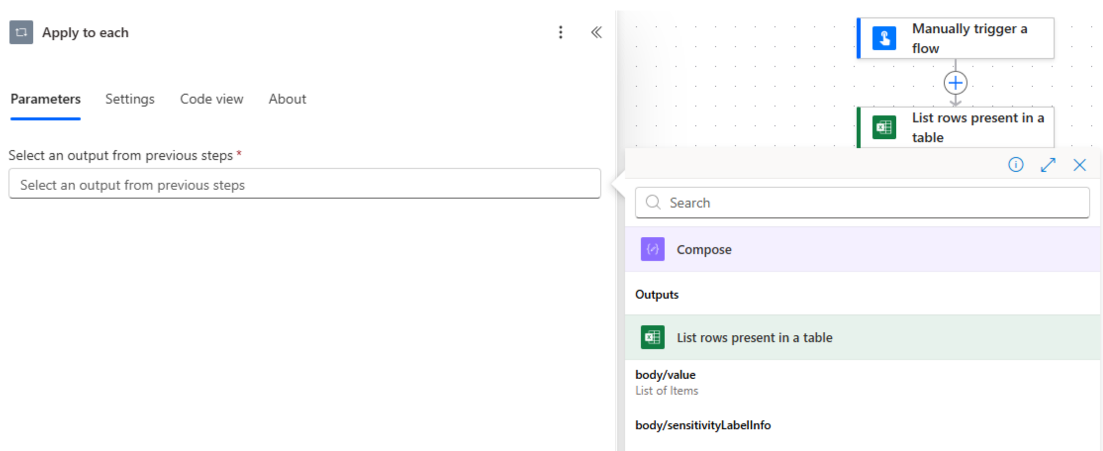
-- **End of Task** --

Task 4: Concurrency Settings

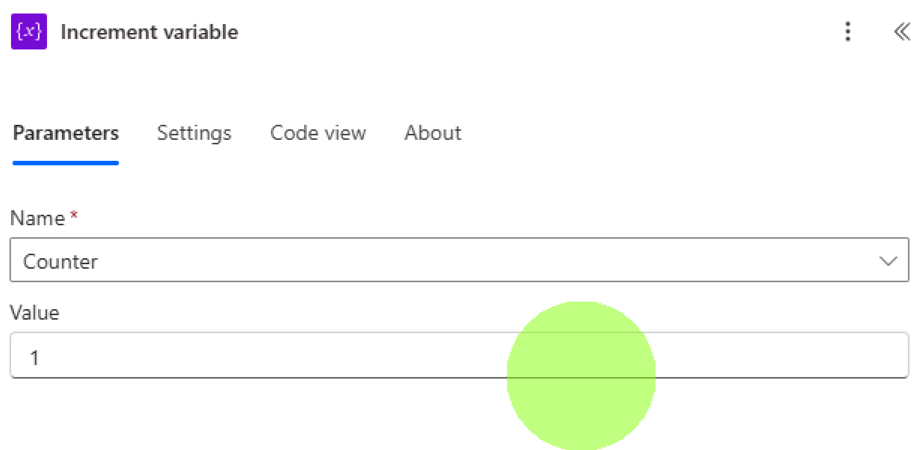
1. Add a new stage Initalize Variable and initialize a variable name Counter with integer as data type and 0 as value.



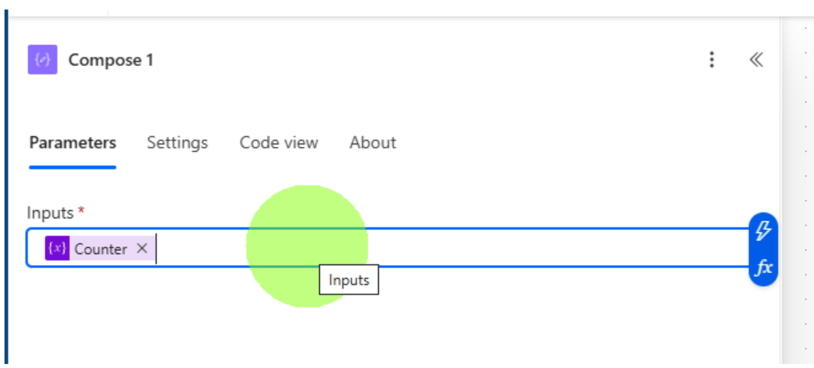
2. Add a new stage Apply to each and choose body/value output from List rows present in a table.



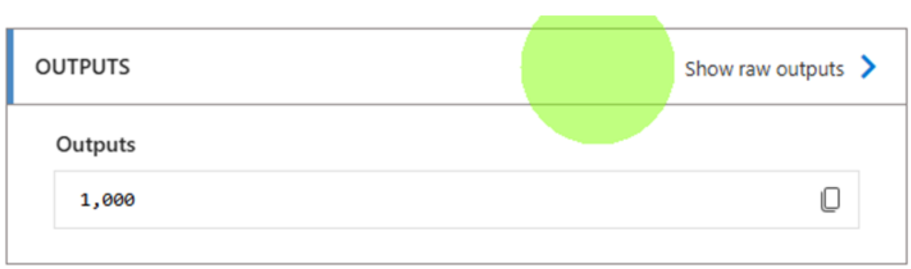
3. Add a new stage Increment variable inside Apply to each. Select Counter and set increment value to 1.



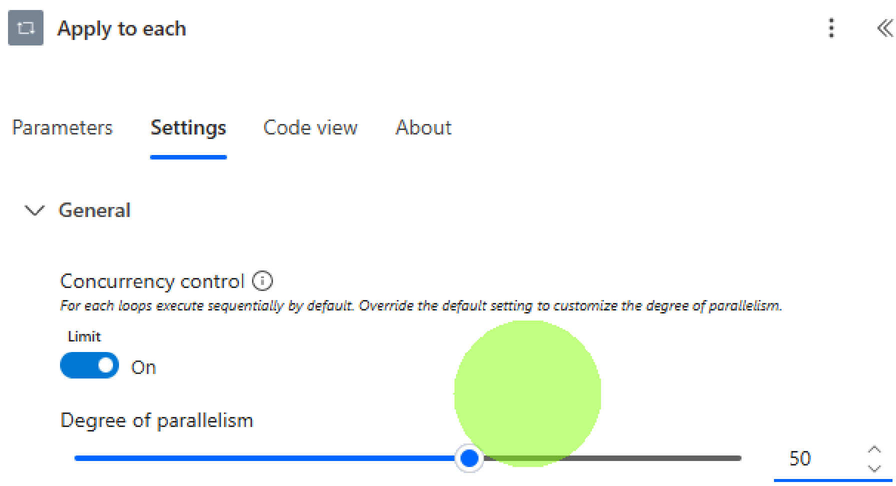
4. Add a compose stage after Apply to each with below formula to observe the final variable value.



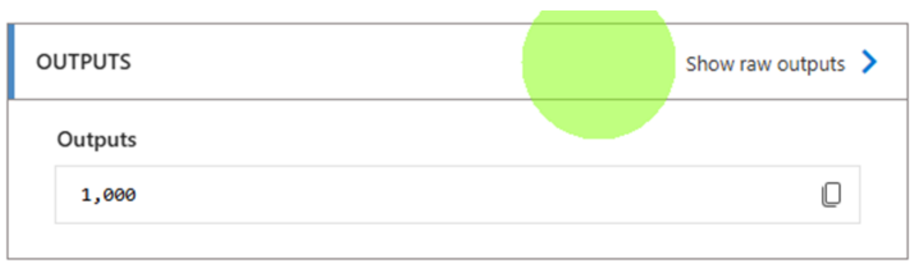
5. You can save and test the flow and notice in Compose 1 stage output that the final incremented value is 1000. And will notice that the flow runs for a long time. That's because Apply to each runs sequentially by default

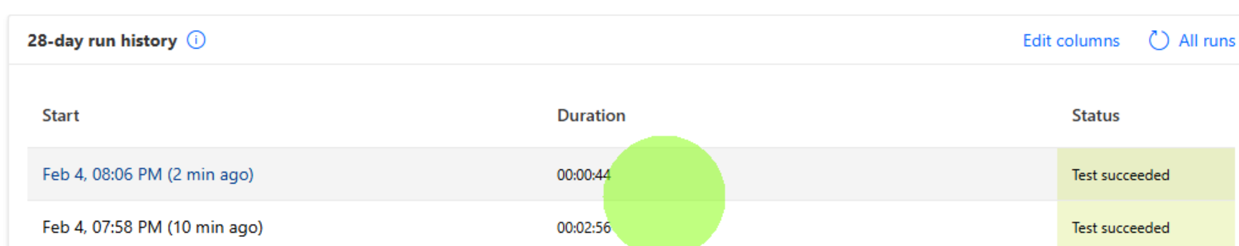


6. Select Apply to each stage and switch to Settings tab. Turn on Concurrency and set degree of parallelism to 50.



7. You can run again and will notice that the flow runs faster and incremented value is still 1000.

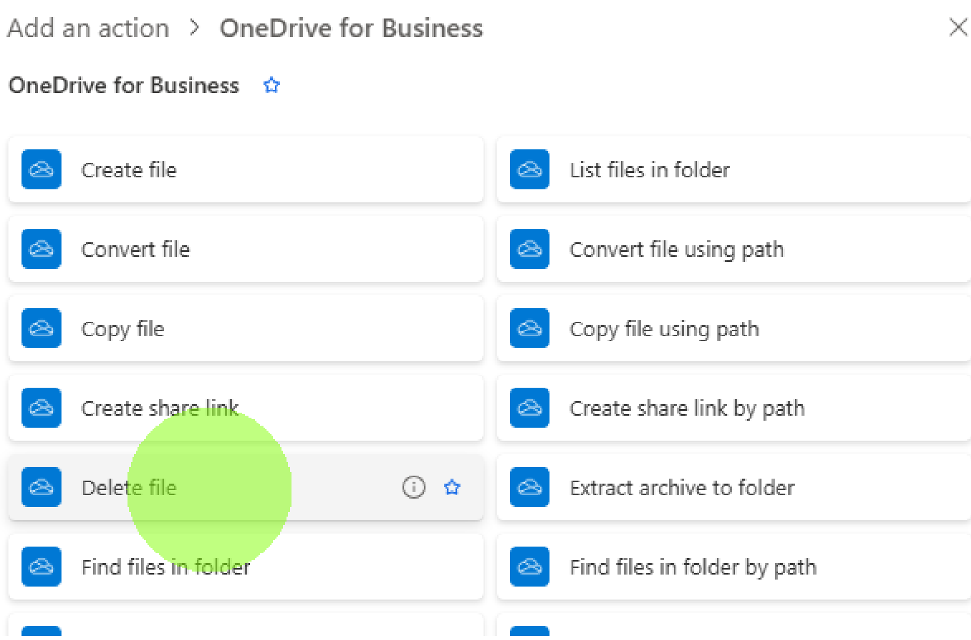




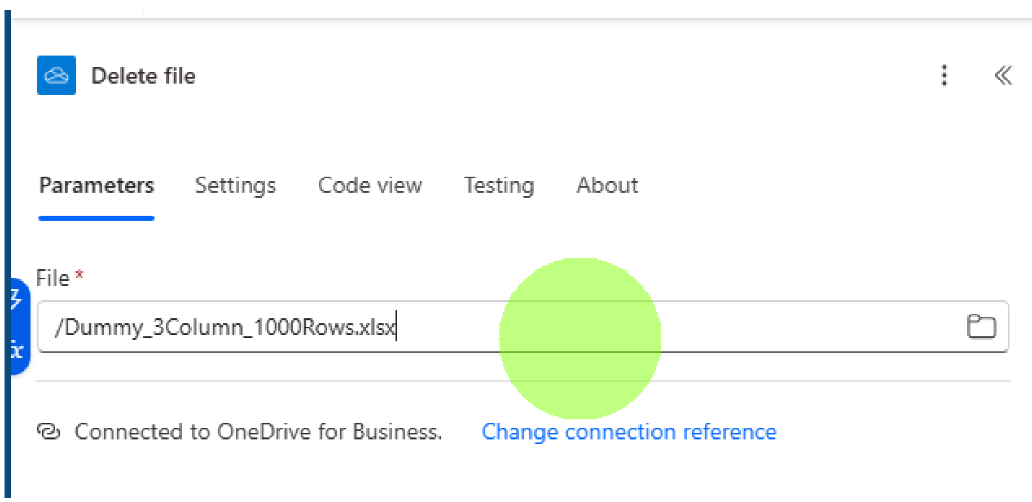
-- **End of Task** --

Task 5: Run after Settings

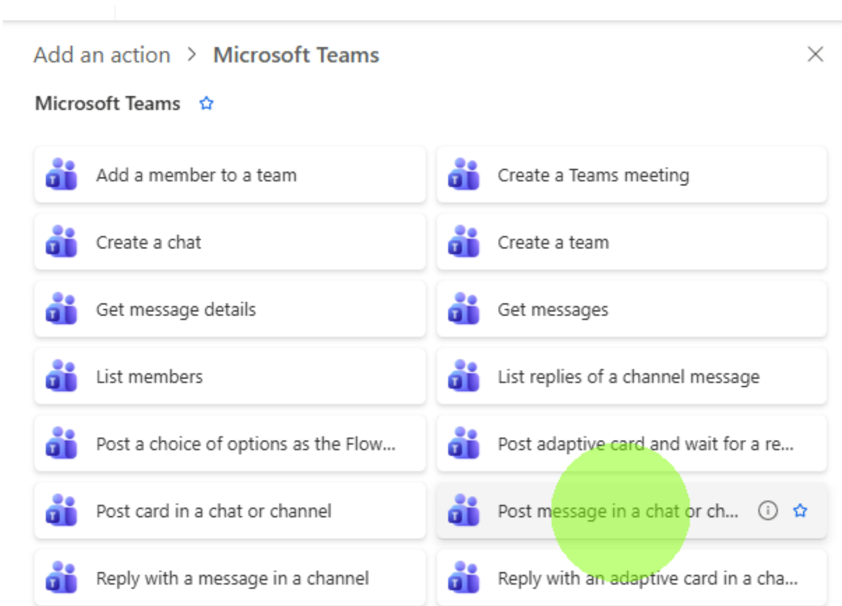
1. Add a new step with OneDrive for Business. Select Delete File.



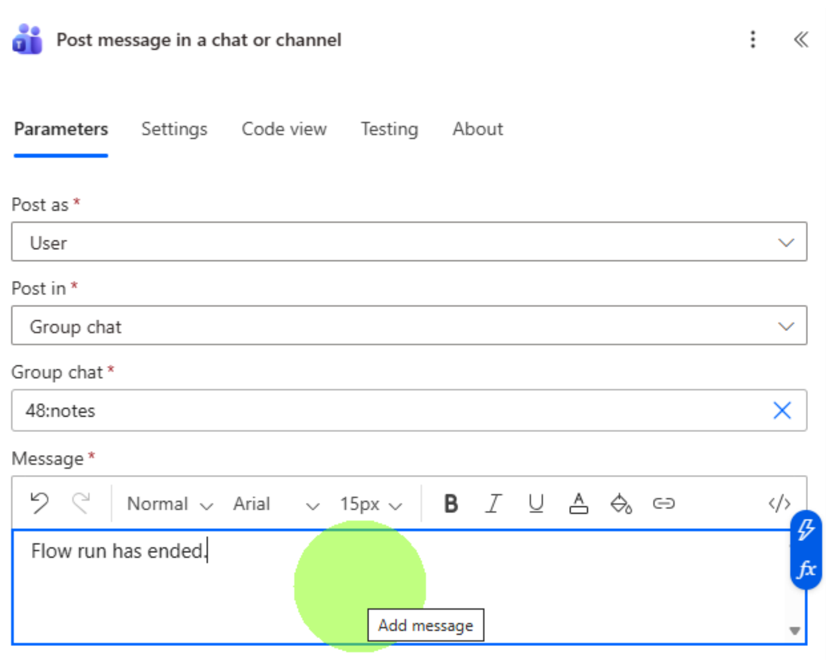
2. Select the excel file uploaded.



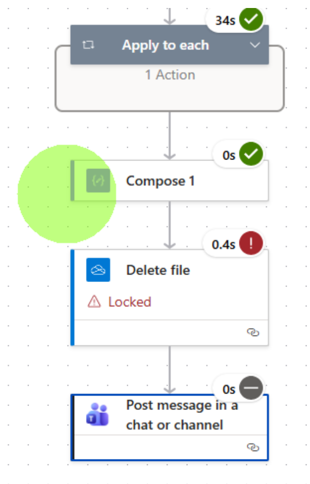
3. Add a new stage Post Message in a Chat or Channel under Teams.



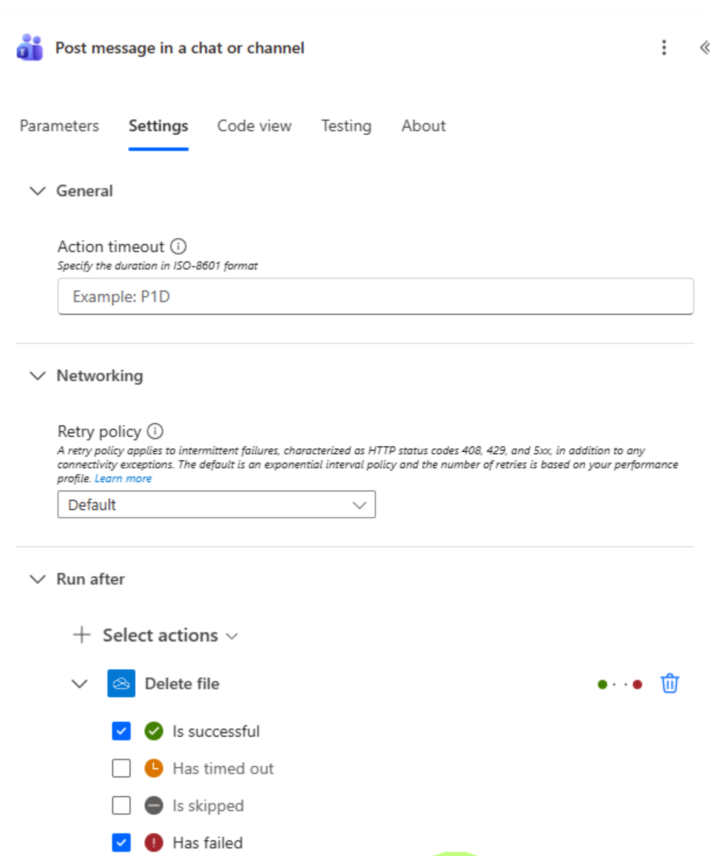
4. Set the parameters as below.



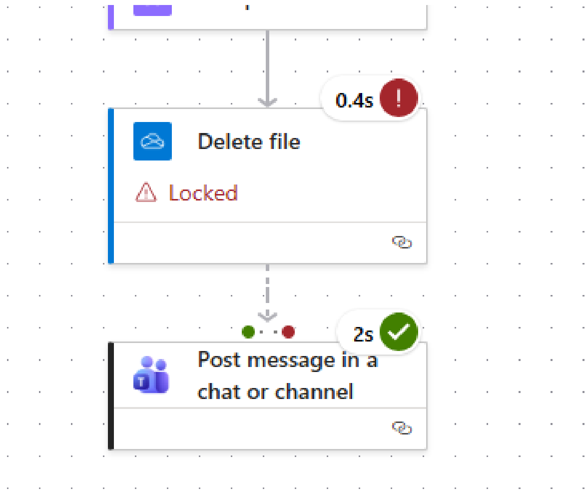
5. Save and run the flow. You will notice Delete file fails as the file is locked for reading and the step Post message in a chat or channel is skipped as the step above failed.



6. Select Post message in a chat or channel stage and switch to Settings tab.
   Check Has failed under Run after settings to make sure that the stage still runs even if the Delete file fails.



7. Save and run the flow again. You will notice the stage now runs regardless of Delete file status.



-- **End of Task** --

Exercise 2: Create Approval Workflow to Run from Power Apps

In this exercise, you'll create an Approval Workflow to be called from the Finance Invoice App created in Day 1 lab guide.

Objectives

After completing this exercise, participants will be able to:

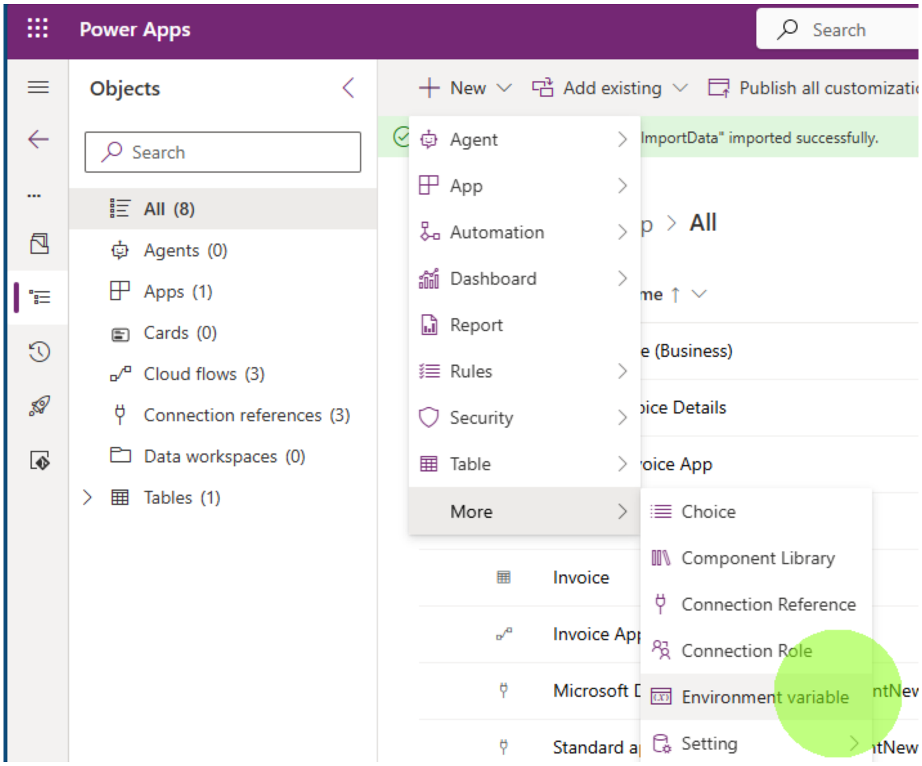
- Create workflows that integrate with Power Apps with input parameters
- Utilize environment variables in their solutions
- Send out approval request using Power Automate

Estimated Time

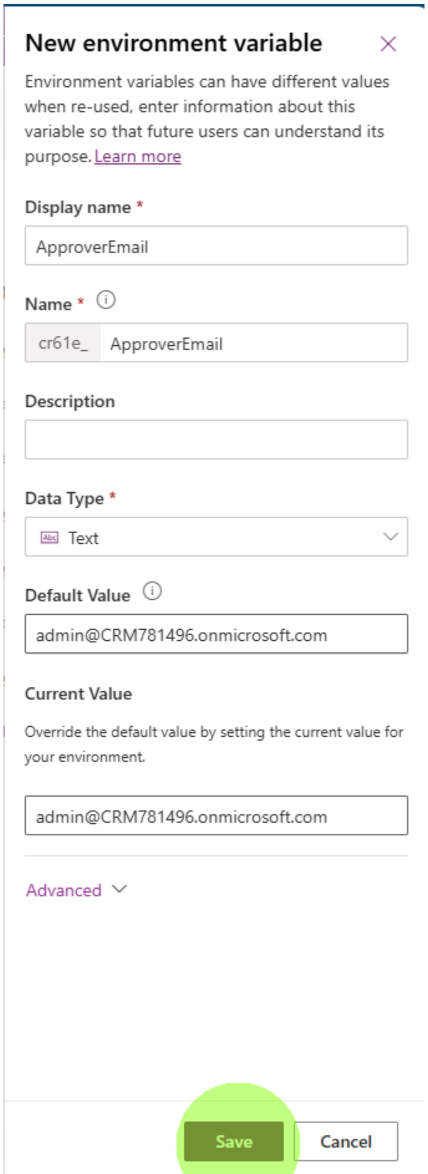
30–40 mins

Task 1: Creating Environment Variables

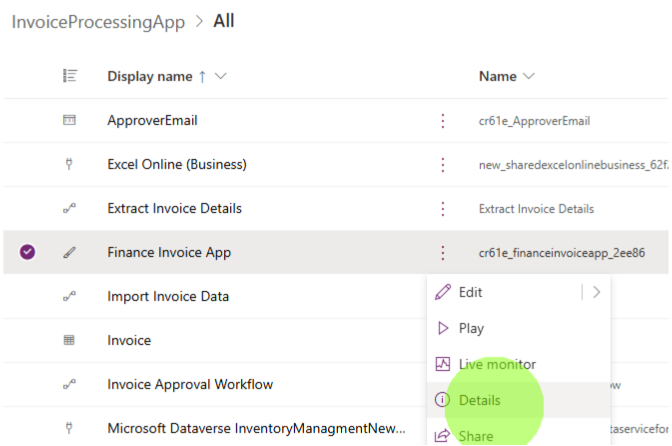
1. Go Invoice Processing App solution and a new Environment Variable in the solution.



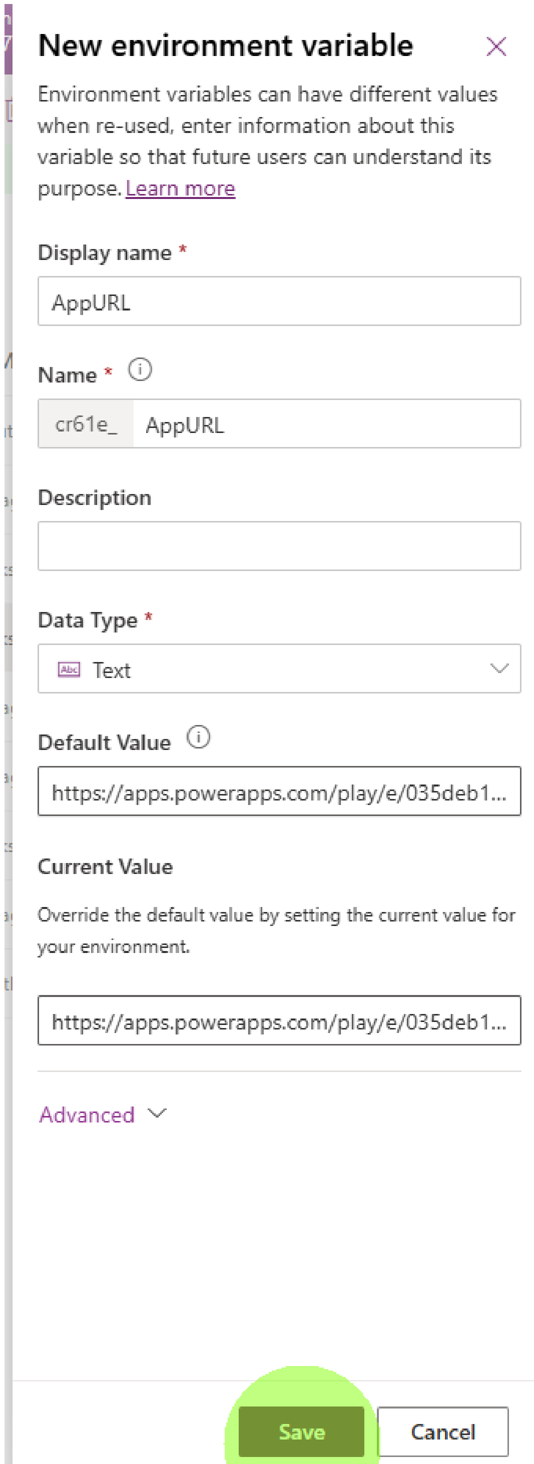
2. Call the variable name as ApproverEmail and set your own email address as the value.



3. Get Canvas App URL from details page.



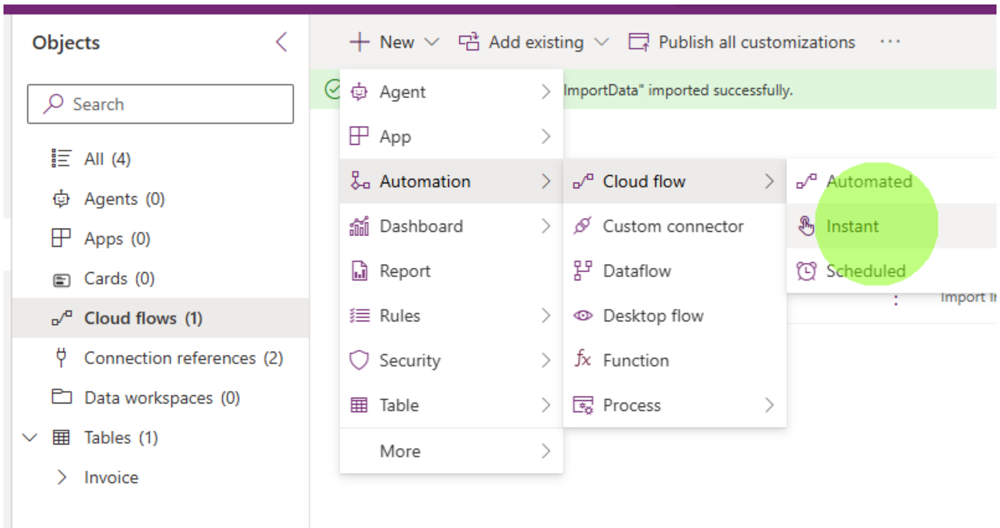
4. And create a new environment variable called AppURL.

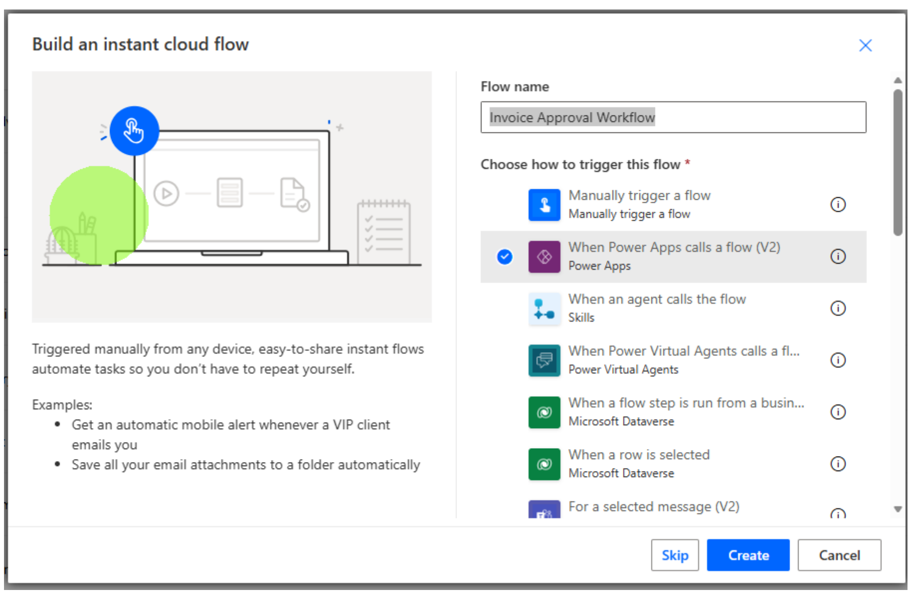


-- **End of Task** --

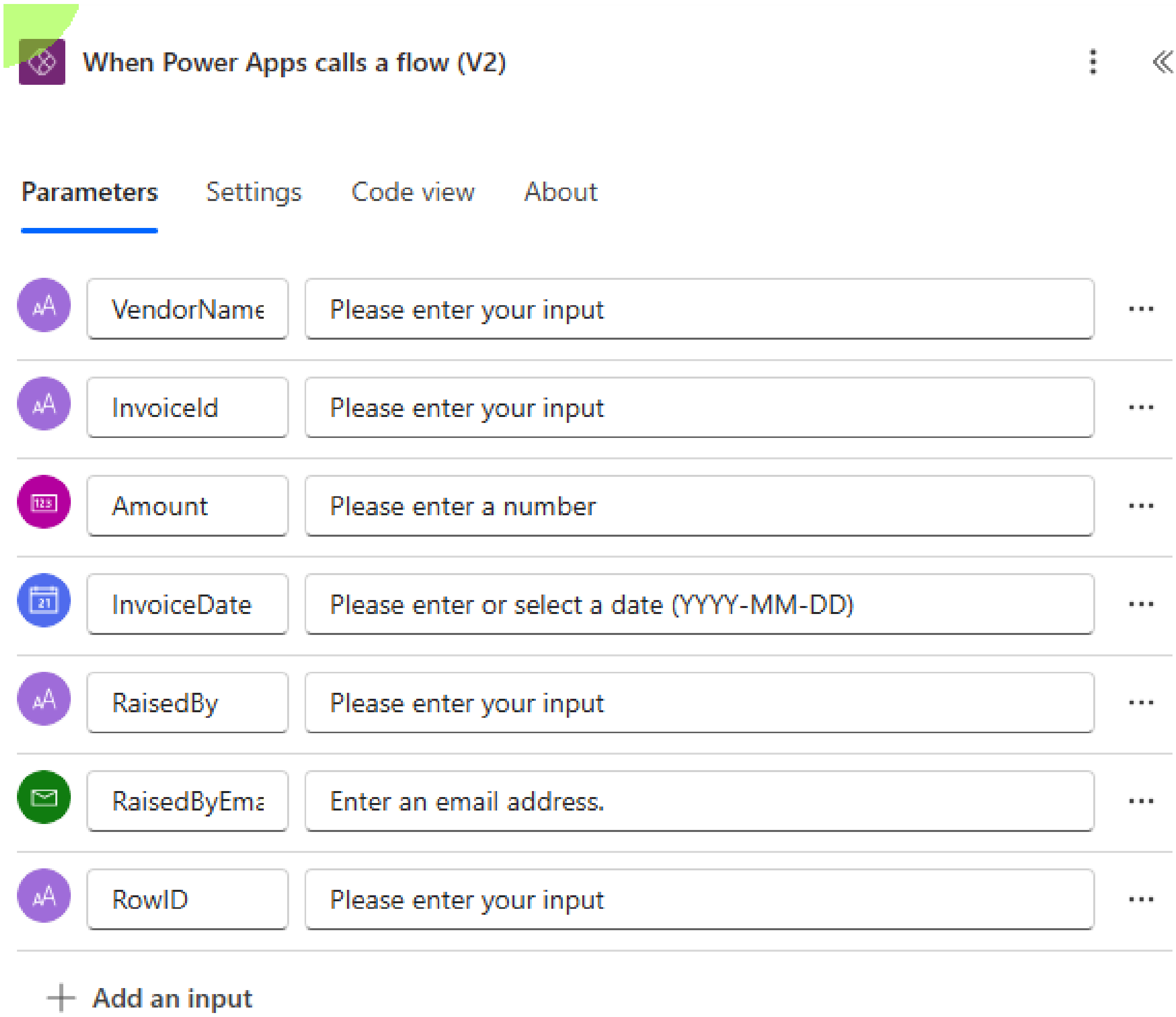
Task 2: Creating a New Flow to Run from Power App

1. Go Invoice Processing App solution and create a new **instant cloud flow,** Name it: **Invoice Approval Workflow**
   Choose **When Power Apps calls a flow (V2)**, then click **Create**

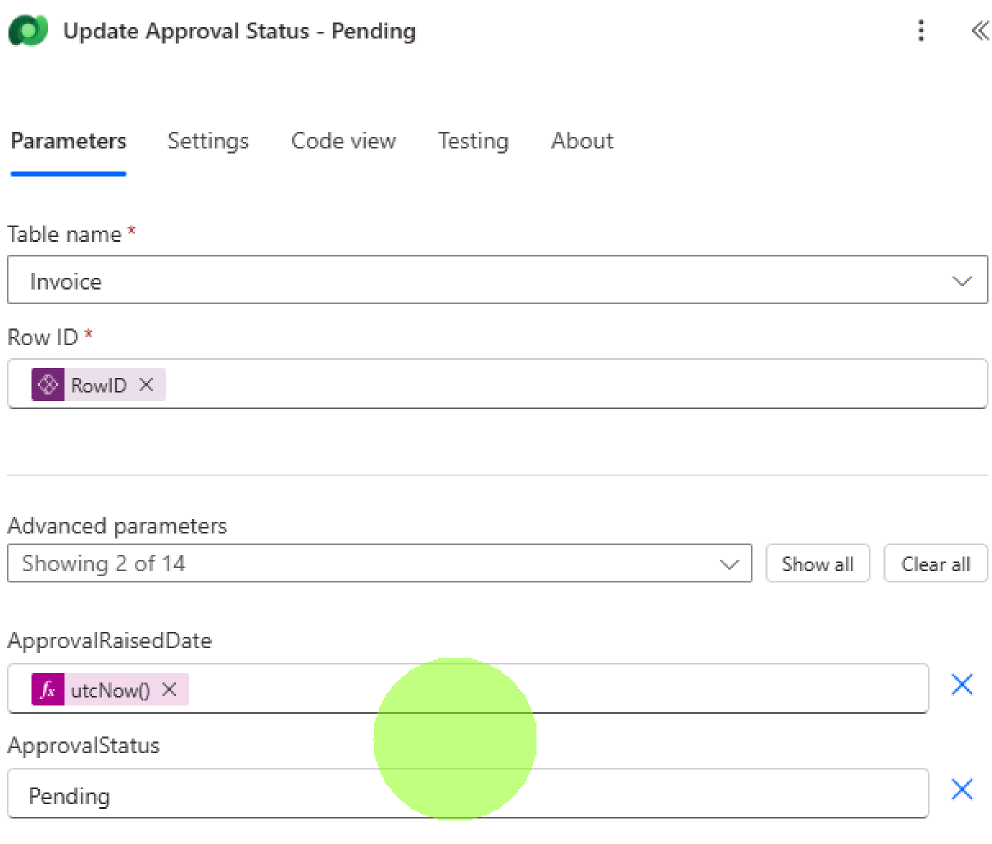




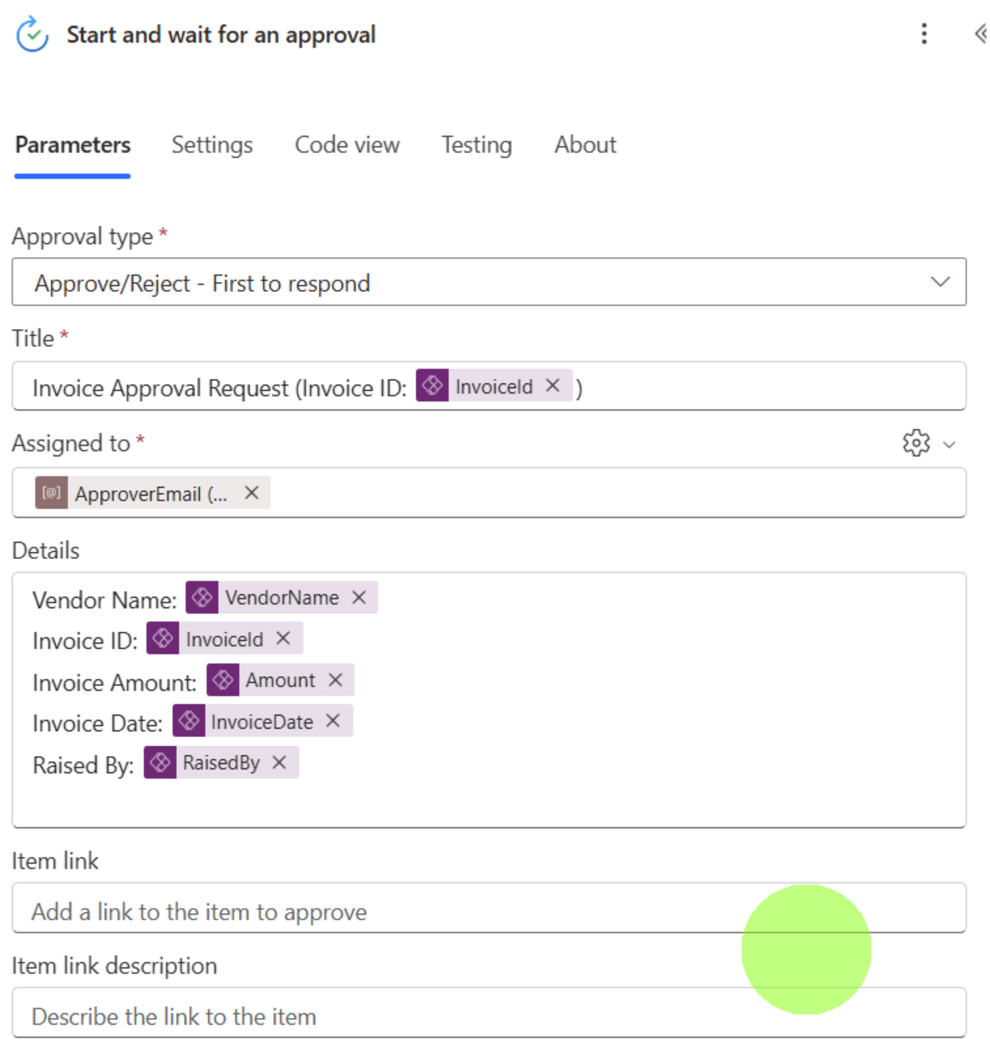
5. Create input parameters to the trigger as below.



6. Add a step **Update a row** from the **Dataverse** connector. And set parameters as below.

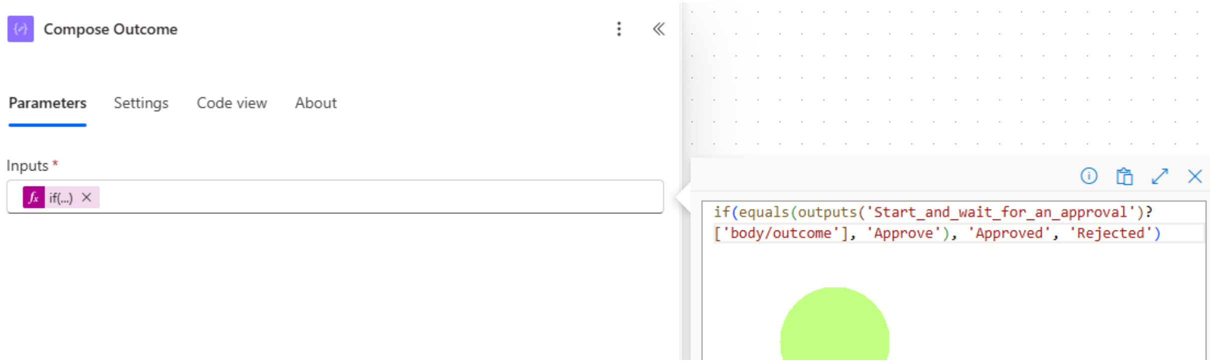


7. Add a step: **Start and Wait for an Approval** from the **Approval** connector. And set parameters as below.

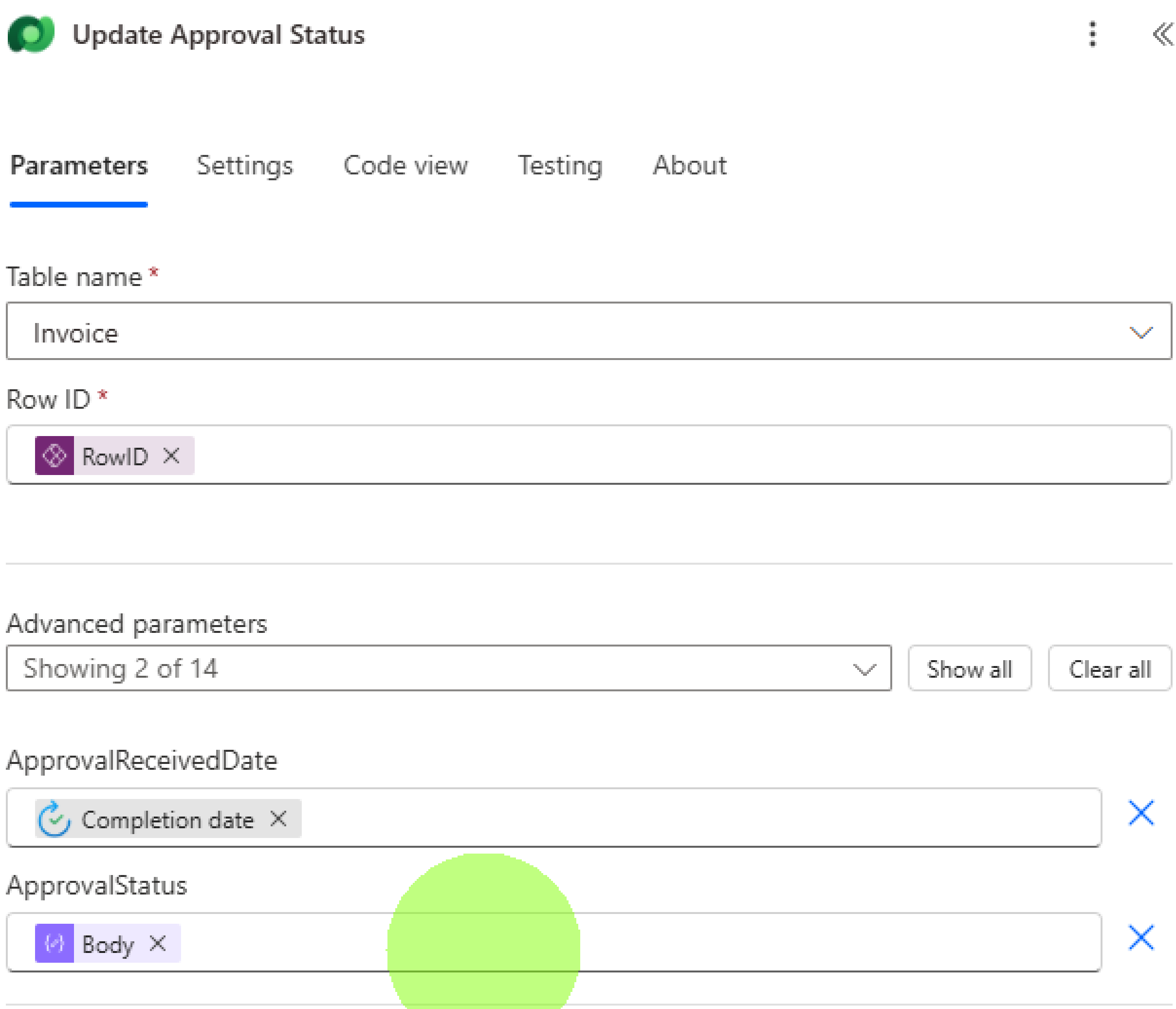


8. Add a Compose stage with the formula below to determine the Approval Outcome.

```
if(equals(outputs('Start_and_wait_for_an_approval')?['body/outcome'],
'Approve'), 'Approved', 'Rejected')
```



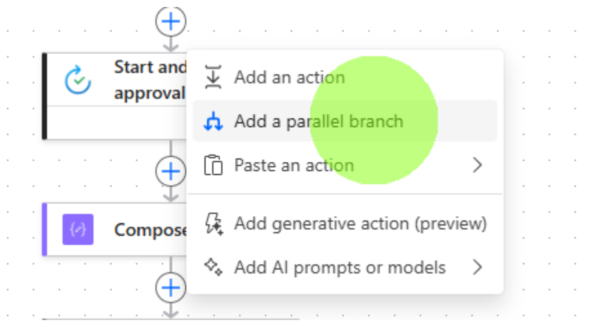
9. Add a step **Update a row** from the **Dataverse** connector. And set parameters as below.



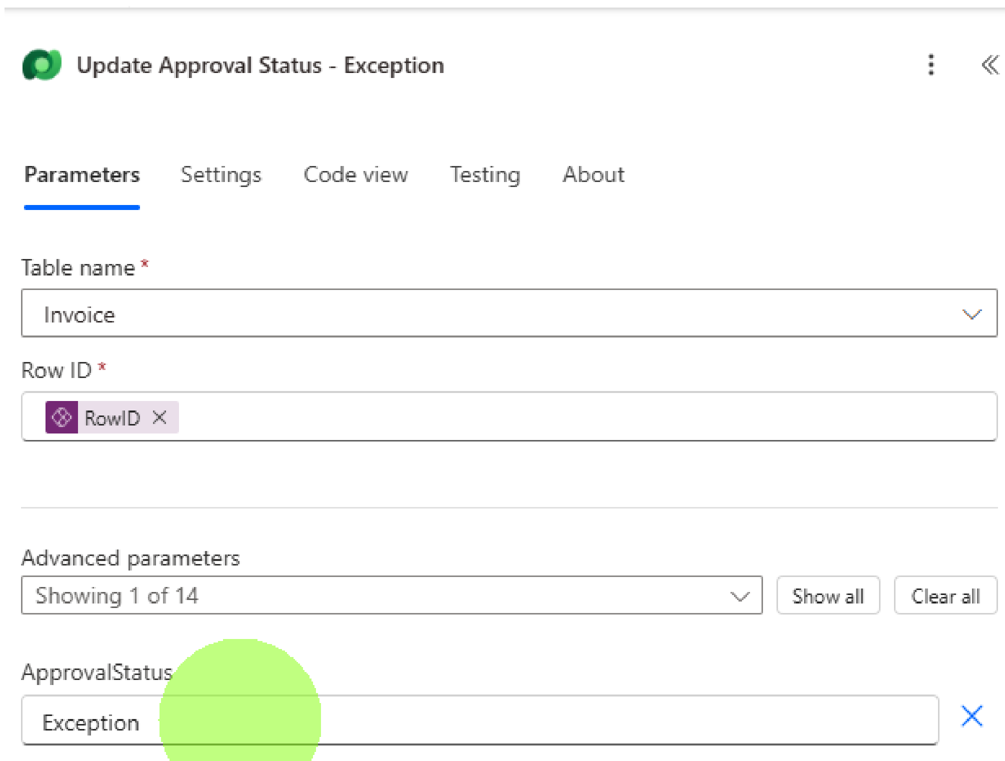
-- **End of Task** --

Task 3: Handling Approval Timeout

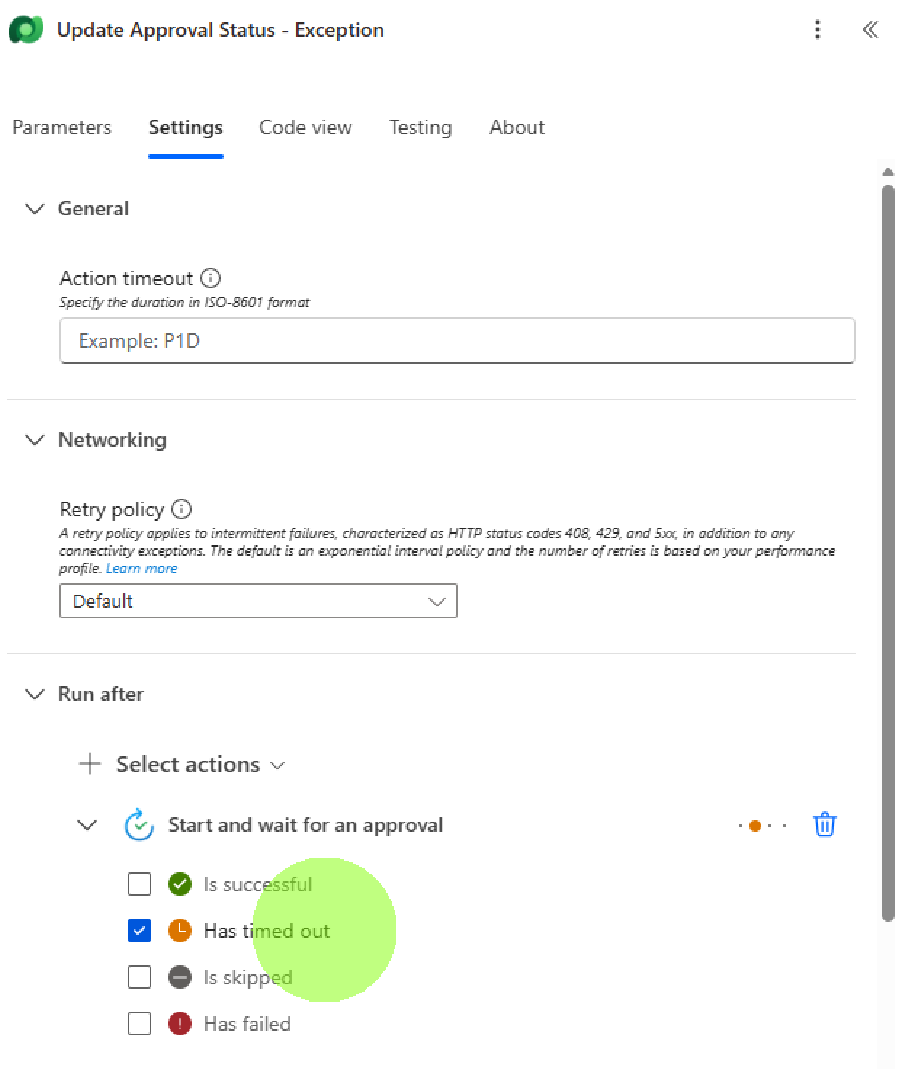
1. Add a parallel branch between Approval and Compose Step.



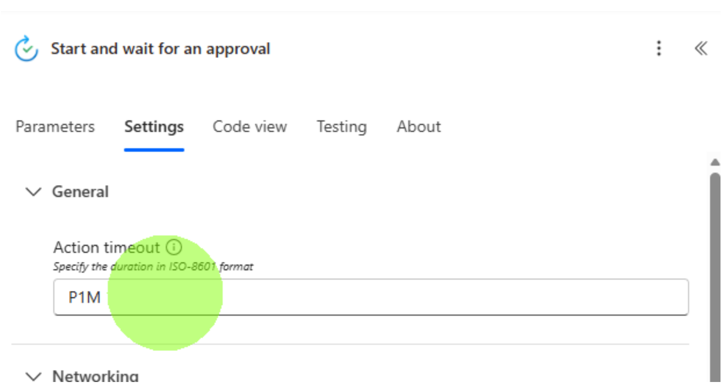
2. Add a step **Update a row** from the **Dataverse** connector. And set parameters as below.



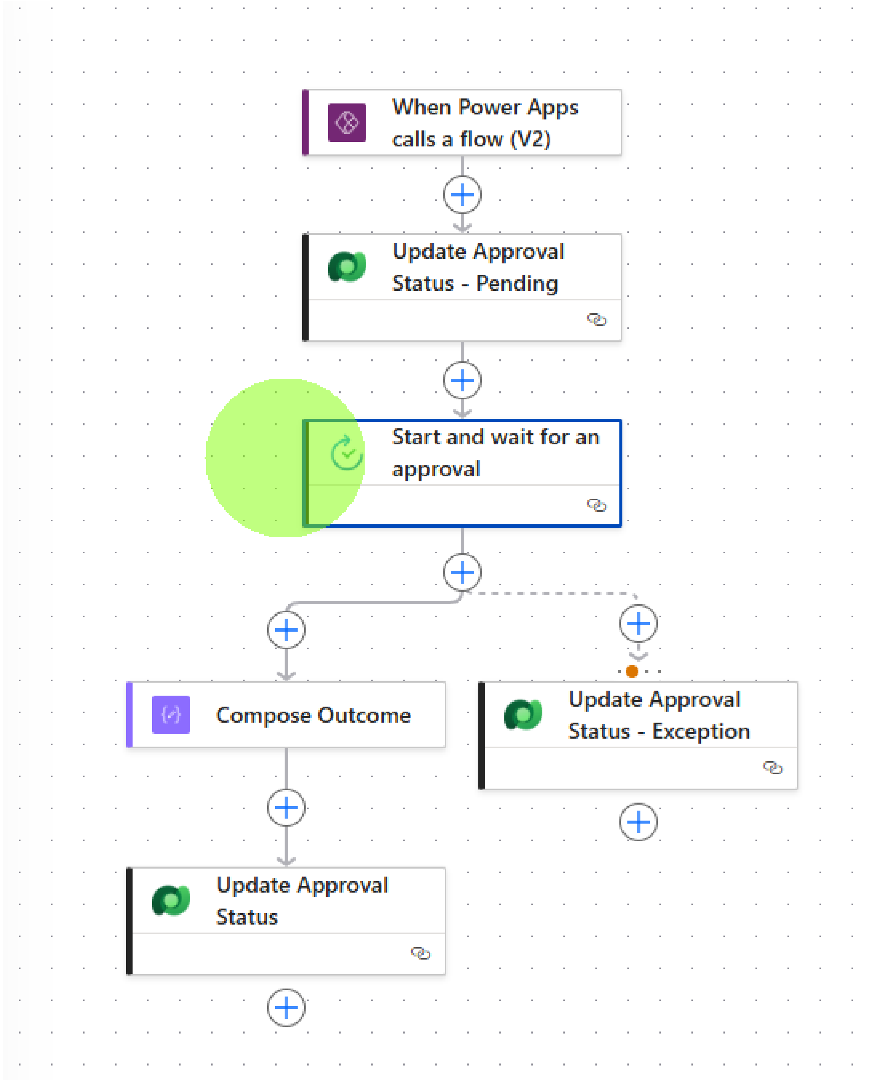
3. Open Settings and set Run after to Has timed out. So this step will only run when Approval stage time outs (i.e. 30 days by default)



4. For the purpose of testing in lab, we will change Action timeout of Approval stage to 1 minute (P1M in ISO-8601 format) as below.



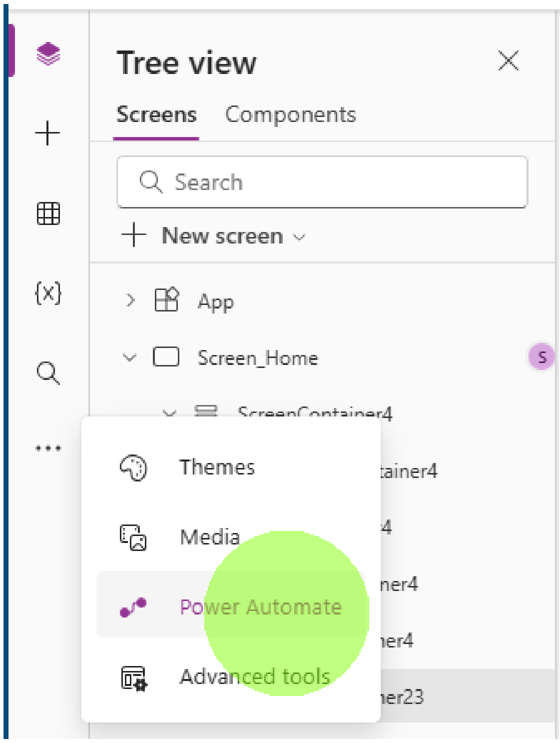
Completed flow should look like this.



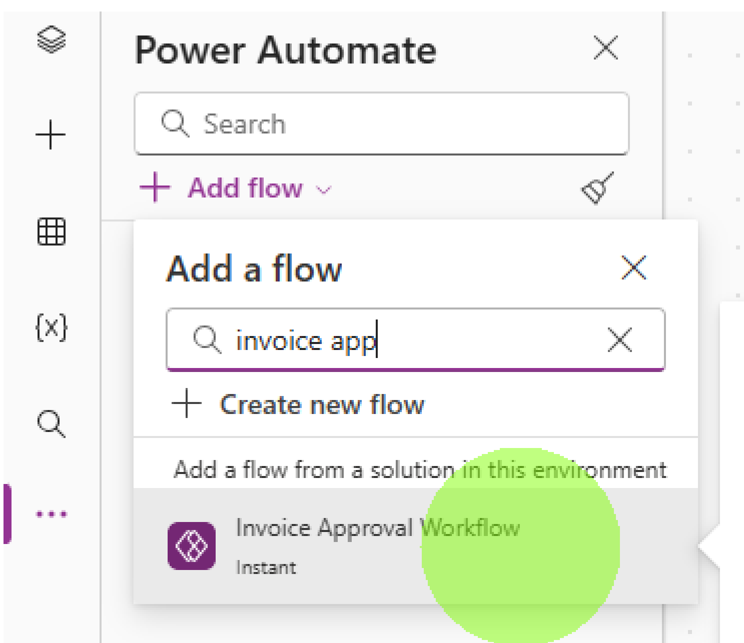
-- **End of Task** ---

Task 4: Adding Flow to App

1. Open Finance Invoice App and select Power Automate.

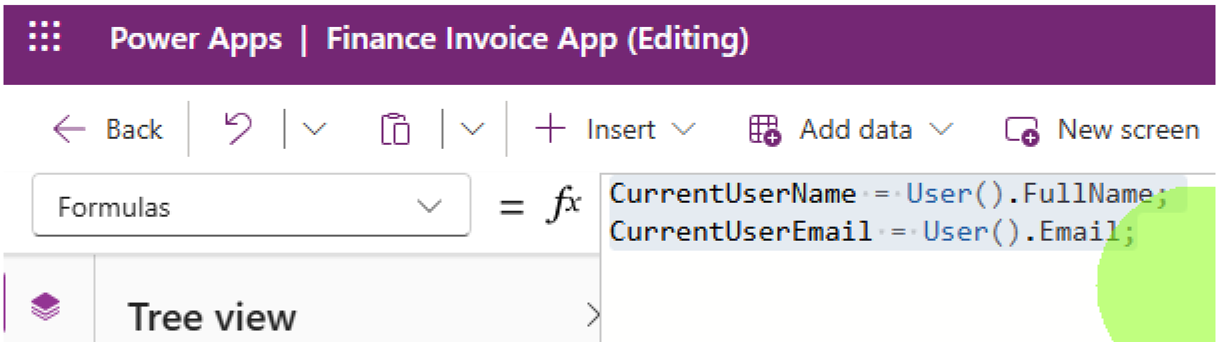


2. Under Add flow, look for Invoice Approval Workflow.



3. Go to App → Formulas property and insert these named formulas.

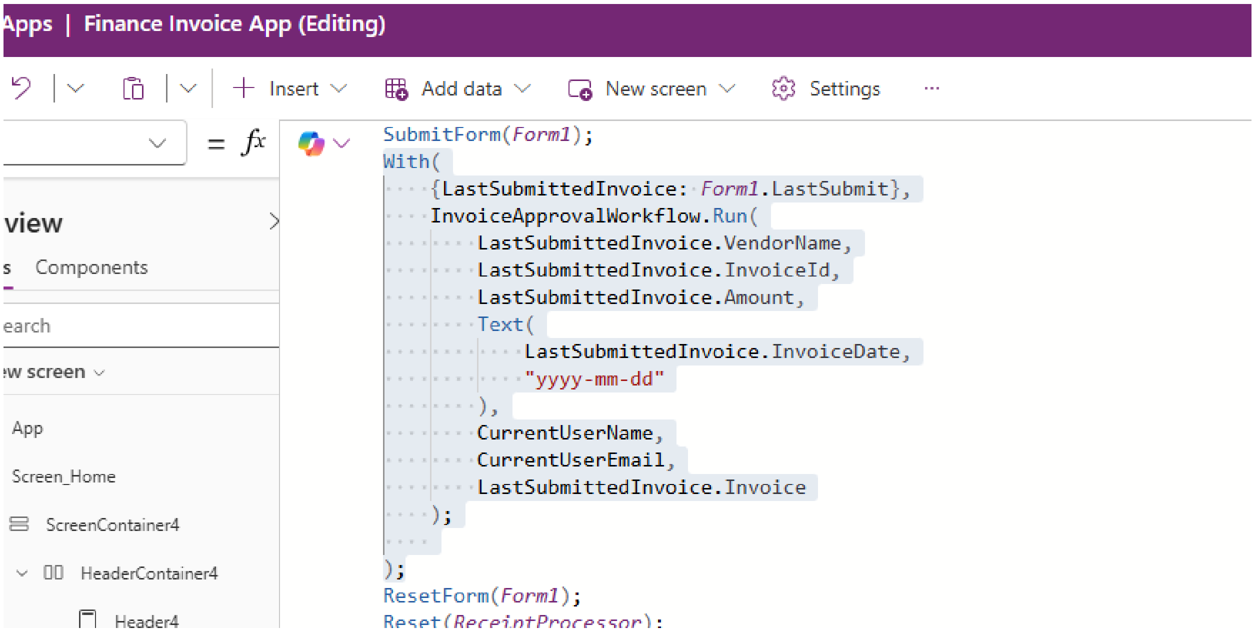
```
CurrentUserName = User().FullName;
CurrentUserEmail = User().Email;
```



4. Go to OnSelect property of Submit button and insert this formula between SubmitForm and ResetForm.

```
With(
    {LastSubmittedInvoice: Form1.LastSubmit},
    InvoiceApprovalWorkflow.Run(
        LastSubmittedInvoice.VendorName,
        LastSubmittedInvoice.InvoiceId,
        LastSubmittedInvoice.Amount,
        Text(
            LastSubmittedInvoice.InvoiceDate,
            "yyyy-mm-dd"
        ),
        CurrentUserName,
        CurrentUserEmail,
        LastSubmittedInvoice.Invoice
    );

);
```



4. You can now save the app and test Submit invoices to see if the approval flow is triggered.

-- **End of Task** --

Exercise 3: Deep Linking for Direct Access

In this exercise, participants will enhance the existing approval flow and app to link the approval request item in the message.

Objectives

After completing this exercise, participants will be able to:

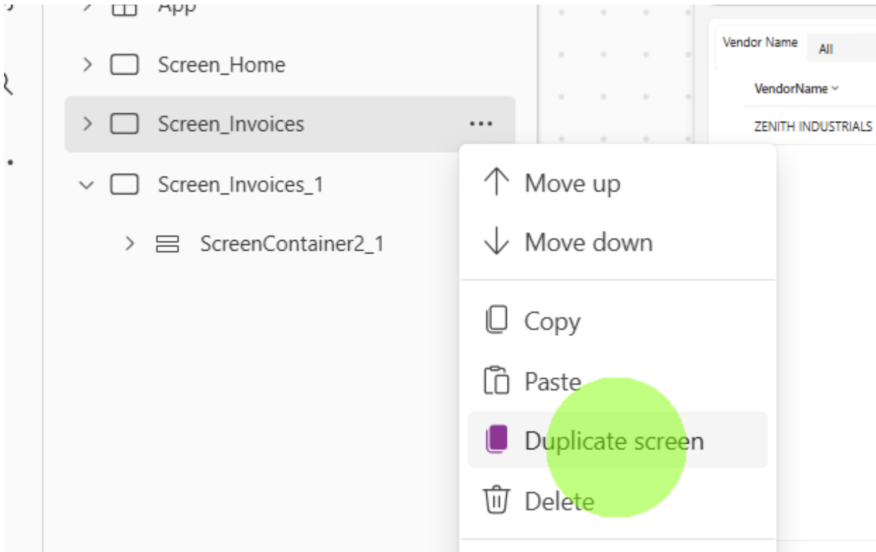
- Perform deep linking in Power App by URL

Estimated Time

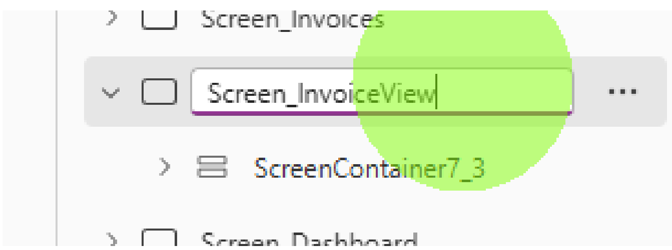
20-30 mins

Task 1: Duplicate the existing Screen

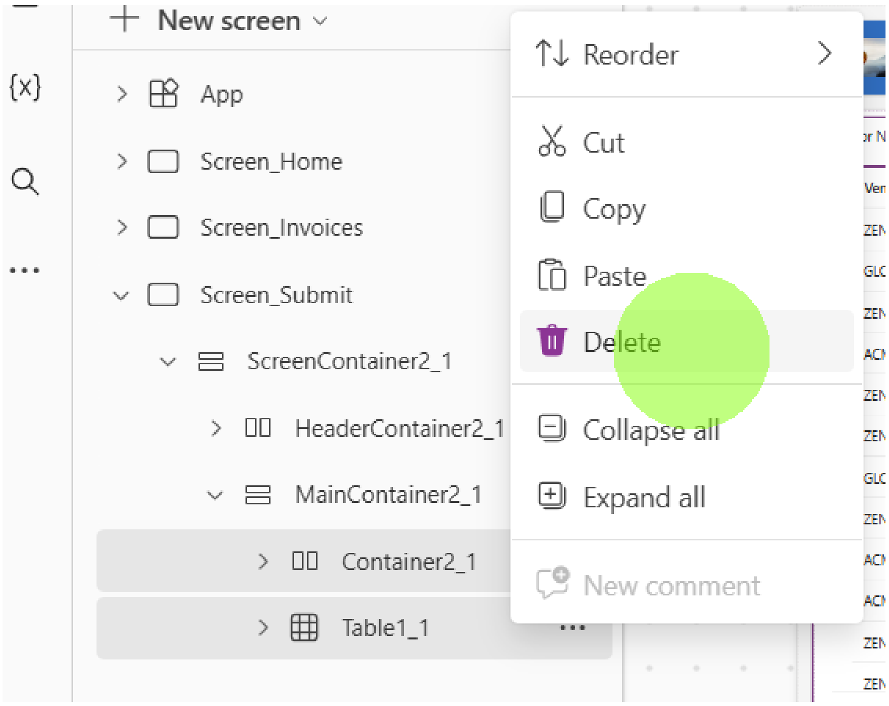
1. Duplicate Screen_Invoices by following selecting the screen, right clicking it, and then selecting the 'Duplicate Screen' option.



Rename the new screen to Screen_InvoiceView.



2. Expand the Main Container and delete the controls inside.

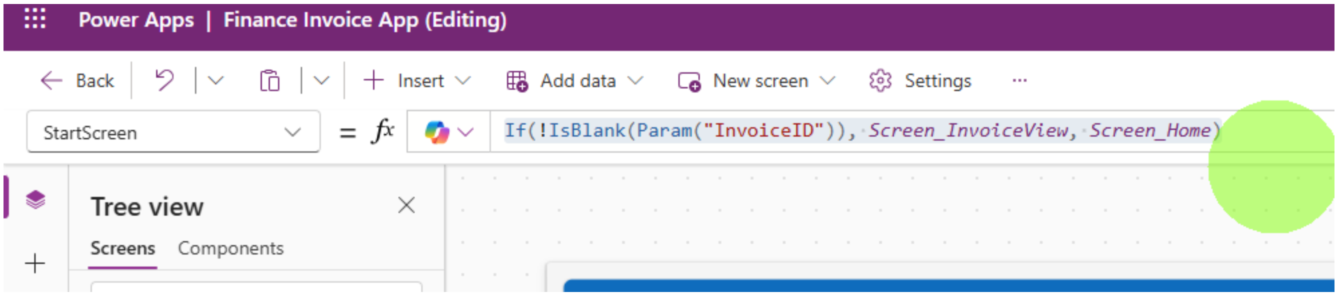


-- **End of Task** --

Task 2: Handle Deep Link ID in App

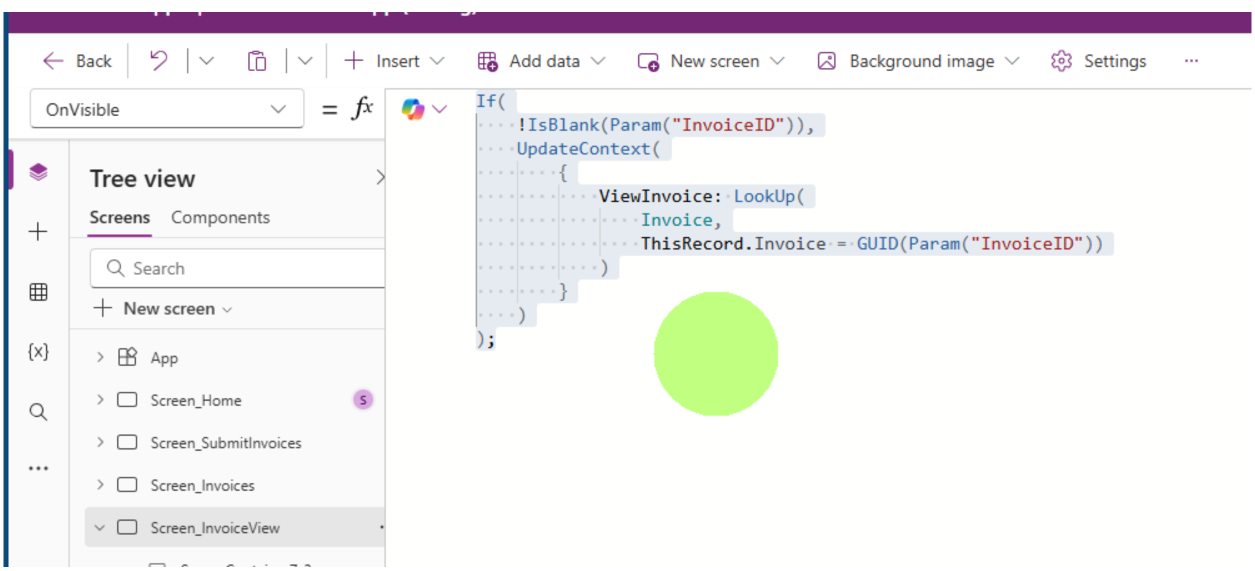
1. Go to App → StartScreen property and put the formula below.

```
If(!IsBlank(Param("InvoiceID")), Screen_InvoiceView, Screen_Home)
```

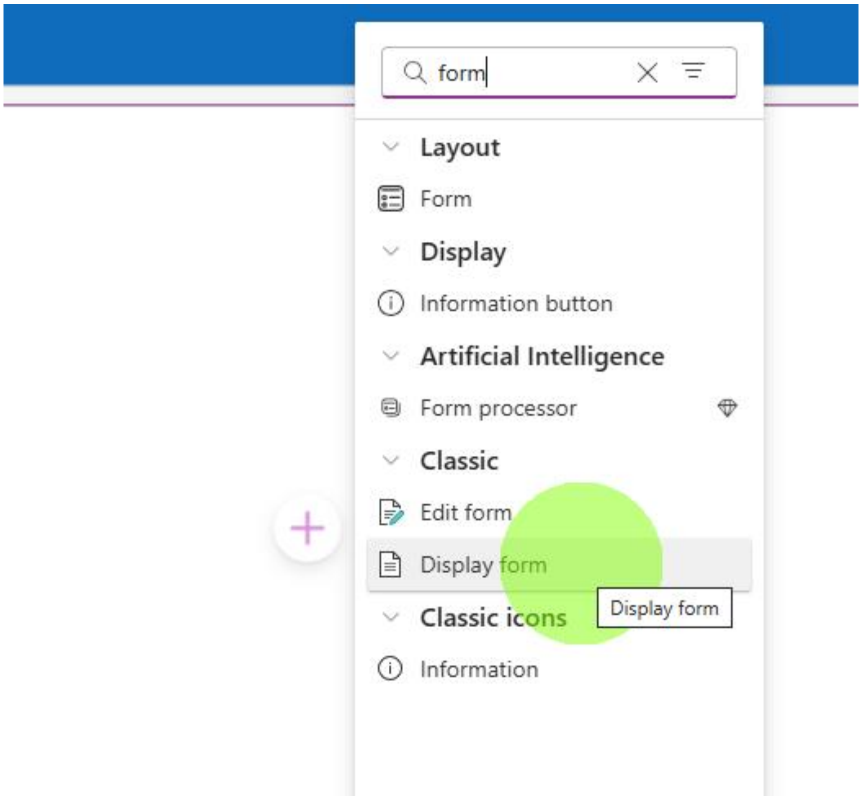


2. Go to OnVisible property under Screen_InvoiceView and put the formula below.

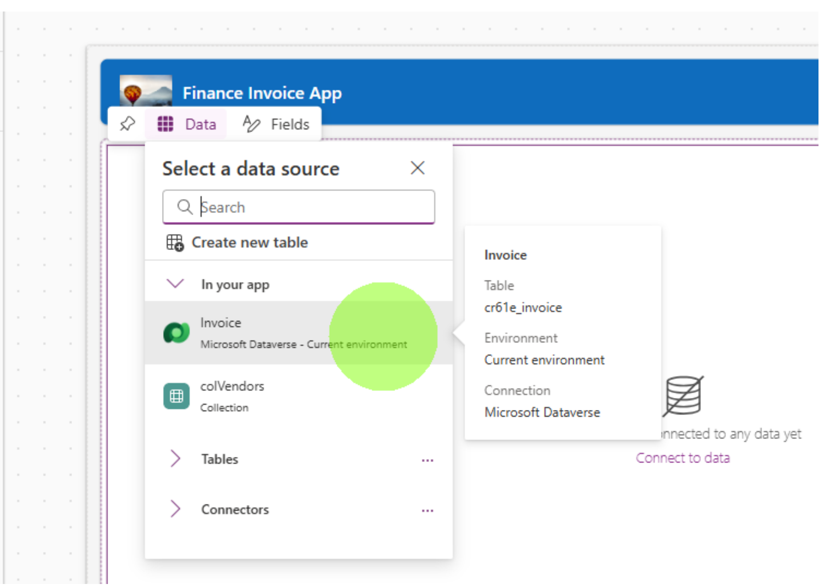
```
If(
    !IsBlank(Param("InvoiceID")),
    UpdateContext(
        {
            ViewInvoice: LookUp(
                Invoice,
                ThisRecord.Invoice = GUID(Param("InvoiceID"))
            )
        }
    )
);
```



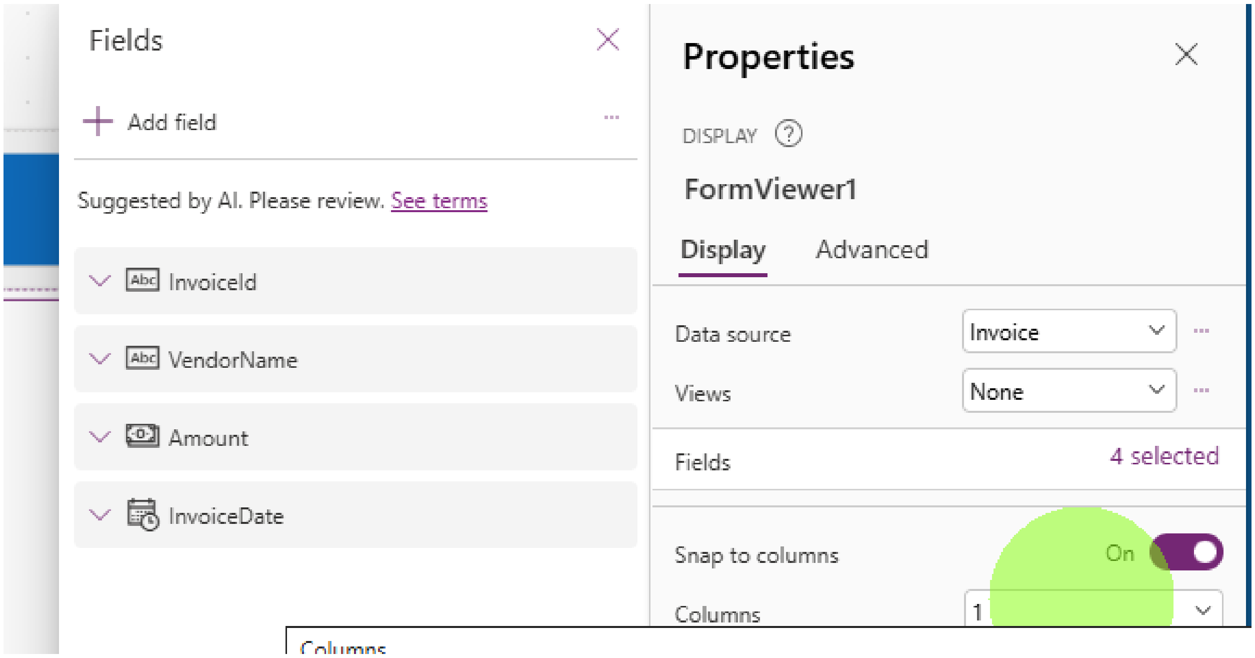
3. Add Display form control to the container.



4. Select Invoice as DataSource.

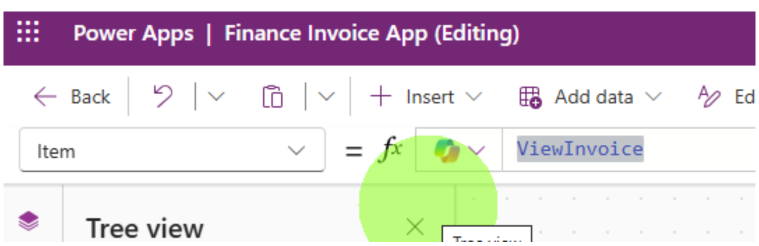


5. Set the Columns to 1 and Edit the fields to show only 4 columns.

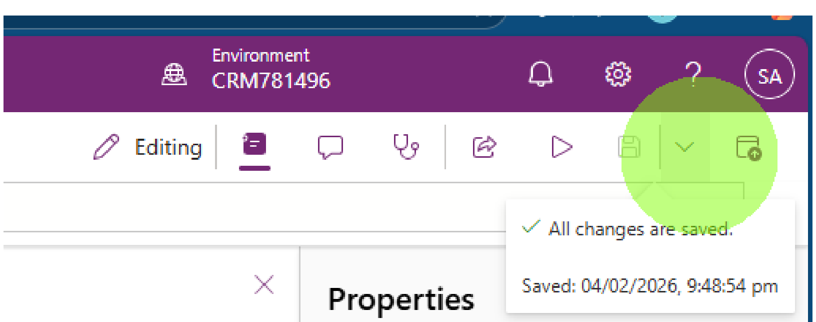


6. Set Item property of the form to below.

`ViewInvoice`



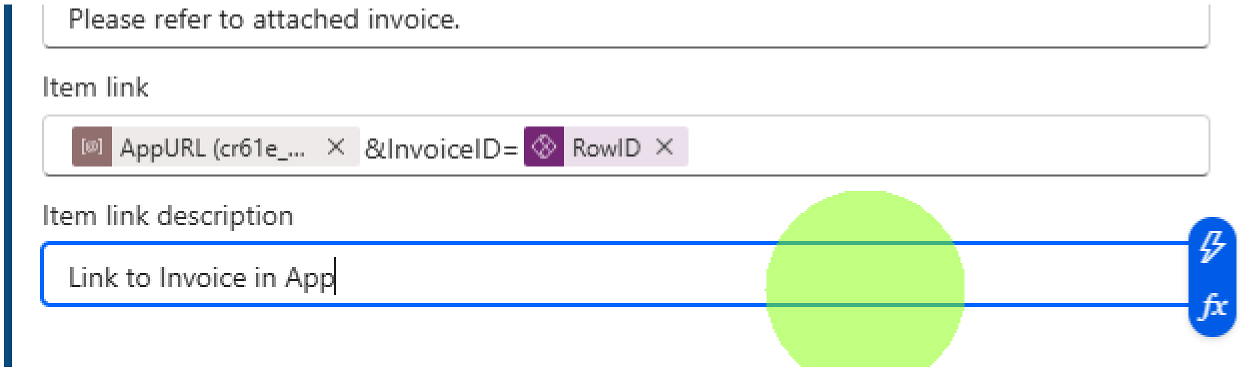
7. Save and publish the app.



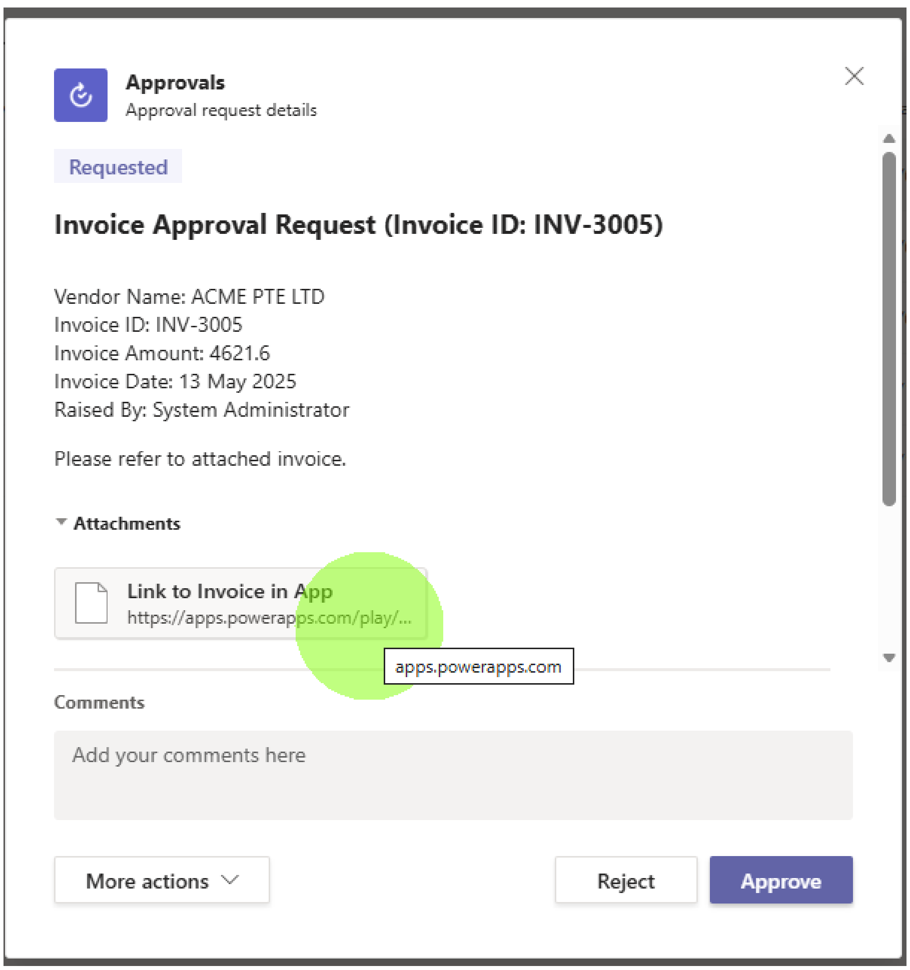
--- **End of Task** ---

Task 2: Include link in Power Automate Approval

1. In Invoice Approval Flow, select Start and wait for an approval stage.
2. Update Item Link and Item link description as below.



3. Save and try submitting a new invoice via app. The new approval request will contain a link.



-- **End of Task** --

Exercise 4: Utilising Adaptive Card and Child Flow

In this exercise, participants will explore sending Teams messages using Adaptive Card in a child flow.

Objectives

After completing this exercise, participants will be able to:

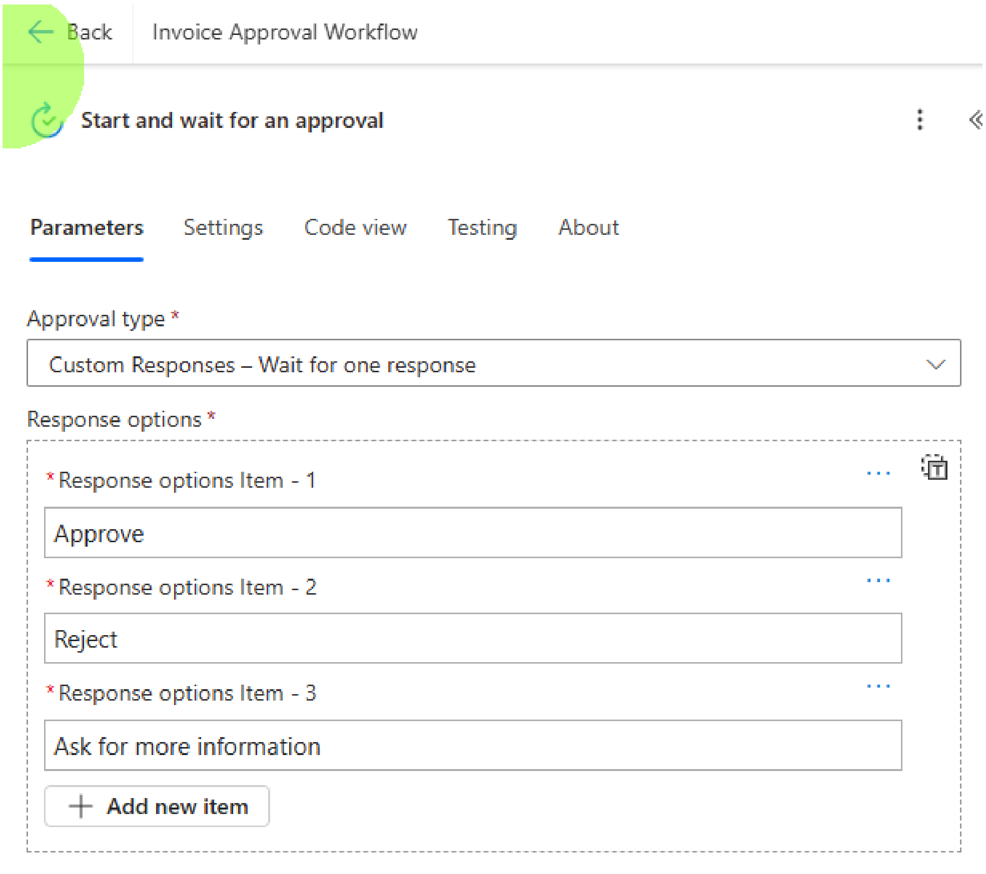
- Use adaptive cards to send out Teams notification and ask for user input
- Modularise their workflows by using Child Flows

Estimated Time

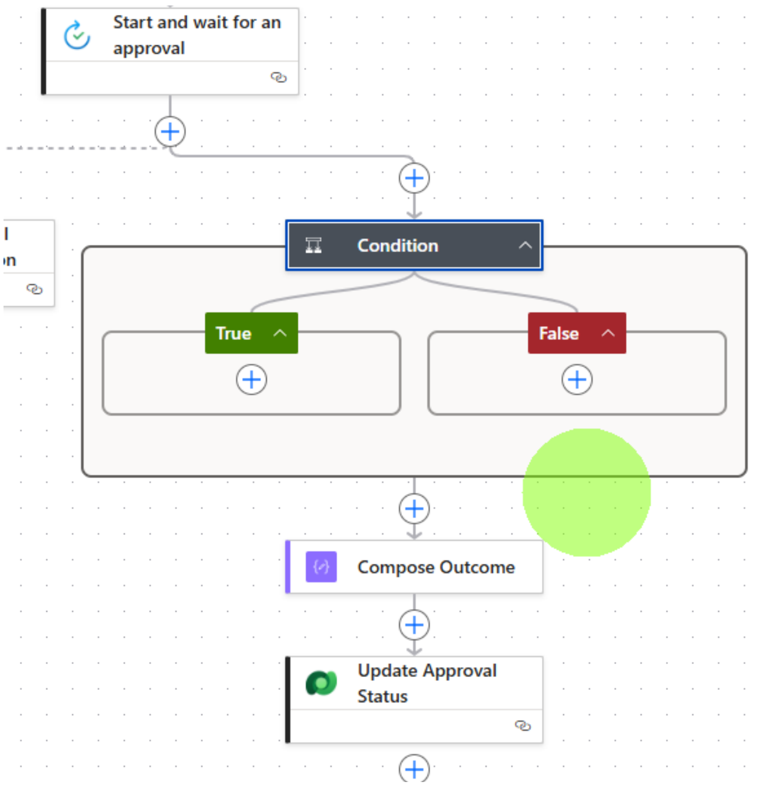
20-30 mins

Task 1: Change Response Options in Approval Workflow

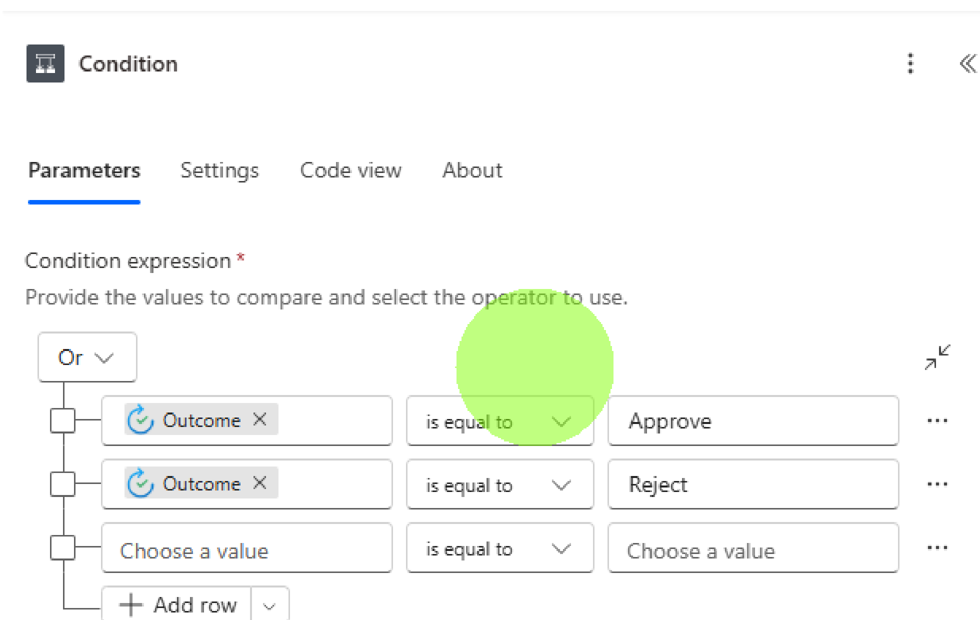
1. Select Start and wait for an approval and change Approval type and Response options as below.



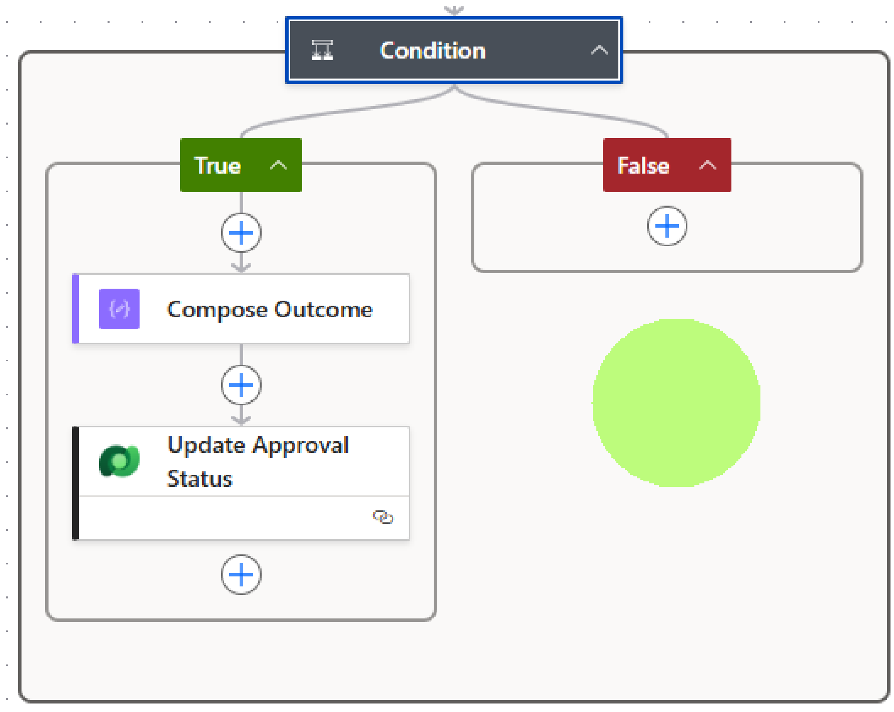
2. Add a Condition stage before Compose Outcome.



3. Set Condition as below.



4. Move Compose and Update Approval Status stages to True branch.

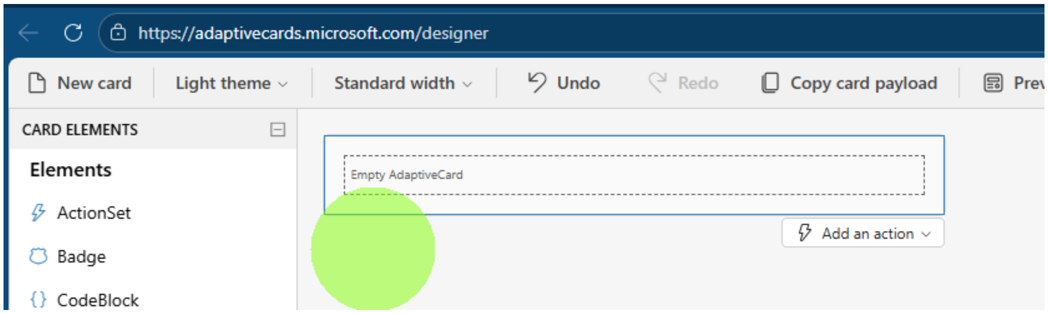


5. Save the flow first. We will create a child flow to put in False branch in the next task.

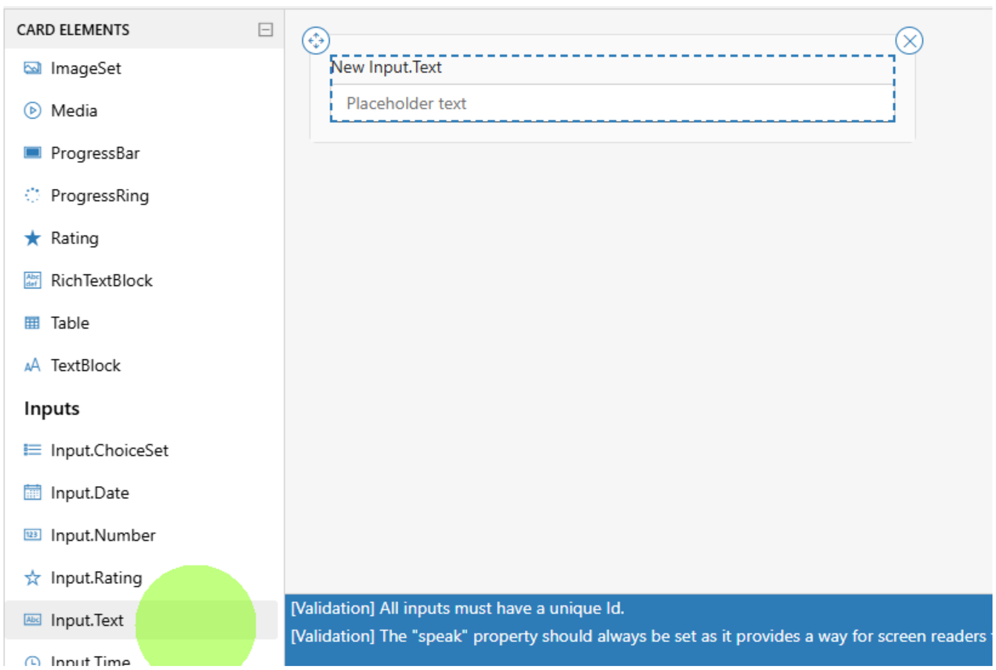
--- **End of Task** ---

Task 2: Design Adaptive card to Get User Input

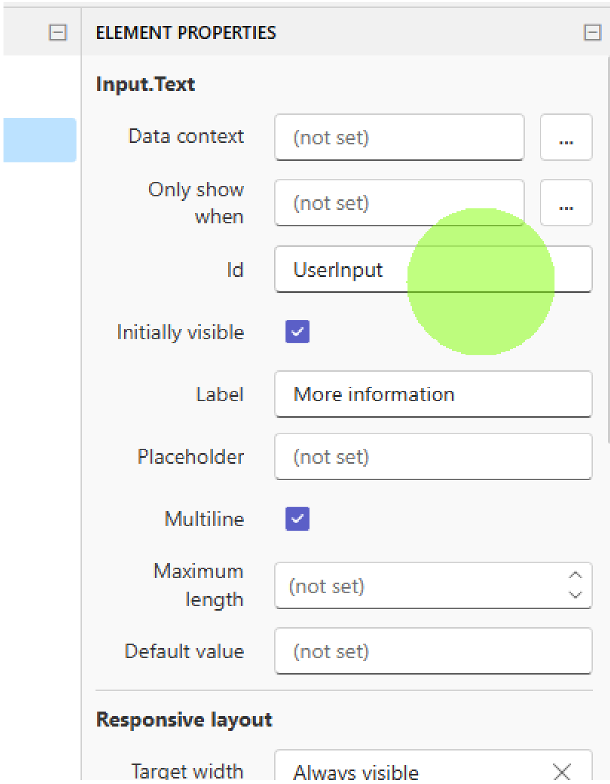
1. Go to https://adaptivecards.microsoft.com/designer.



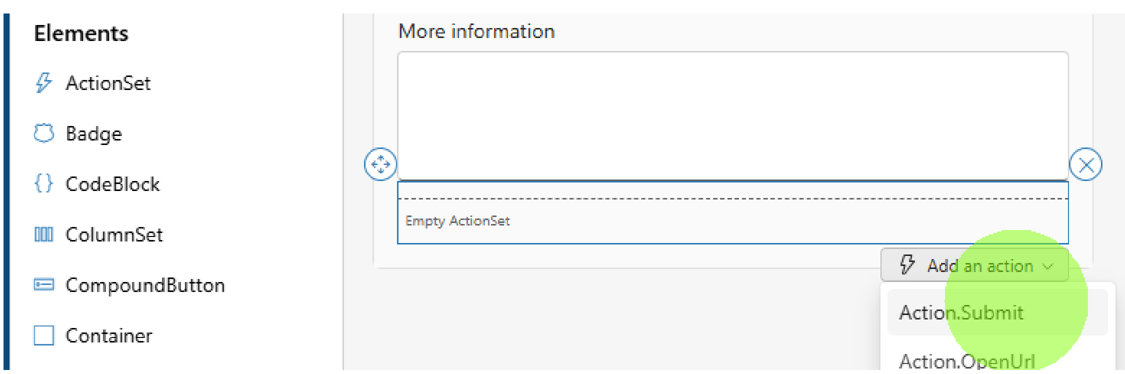
2. Add Input.Text to the empty card.



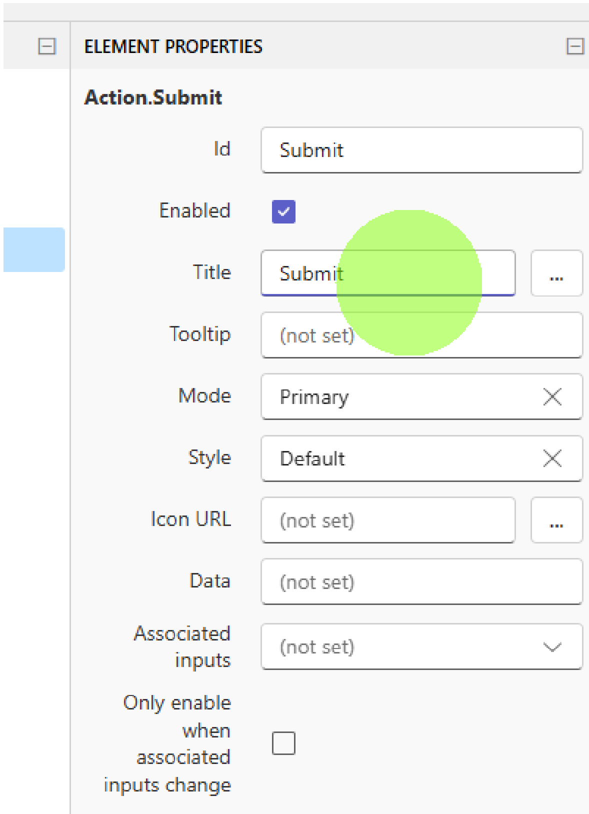
3. Set these values in Element Properties.



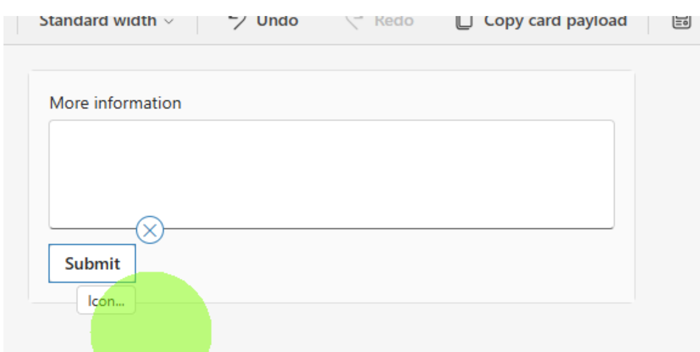
4. Add ActionSet and select Action.Submit.



5. Set these values in Element Properties.



6. Your card should look like below. Copy the text in Card Payload Editor below.



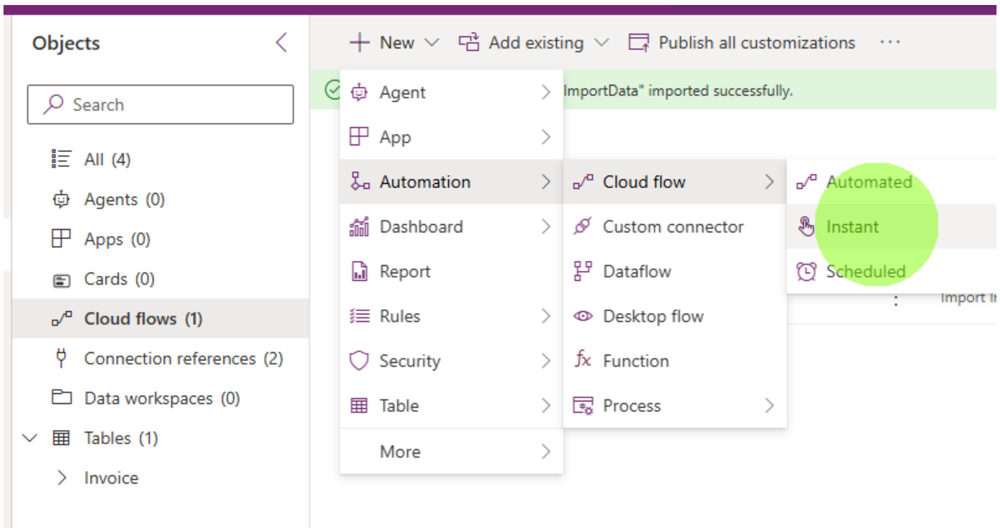
CARD PAYLOAD EDITOR

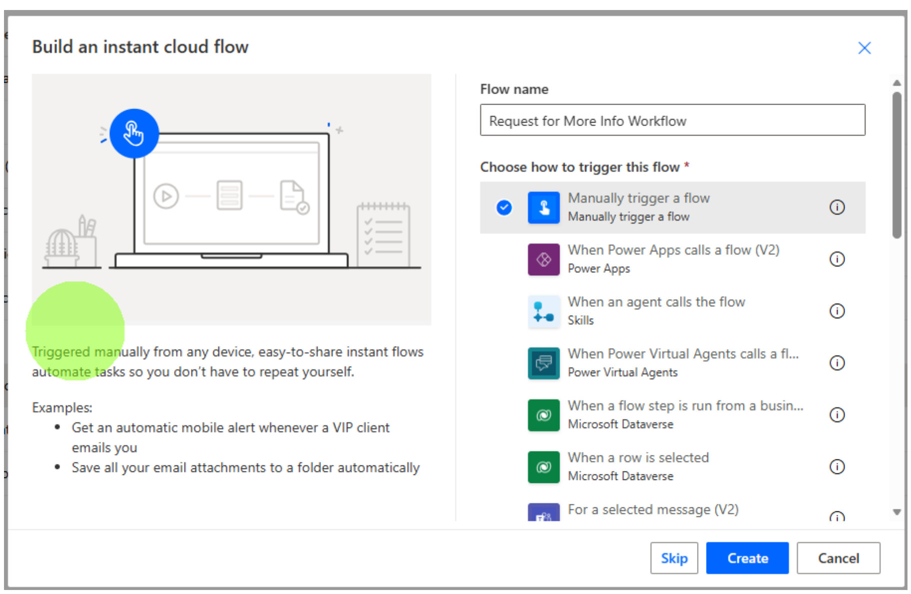
```
{
    "type": "AdaptiveCard",
    "$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
    "version": "1.6",
    "body": [
        {
            "type": "Input.Text",
            "id": "UserInput",
            "label": "More information",
            "isMultiline": true
        },
        {
            "type": "ActionSet",
            "actions": [
                {
                    "type": "Action.Submit",
                    "title": "Submit",
                    "id": "Submit"
```

-- **End of Task** --

Task 3: Create a Child Flow to Get More Information

1. Go Invoice Processing App solution and create a new **instant cloud flow,** Name it:
   Request for More Info Workflow
   Choose **Manually trigger a flow,** then click **Create**

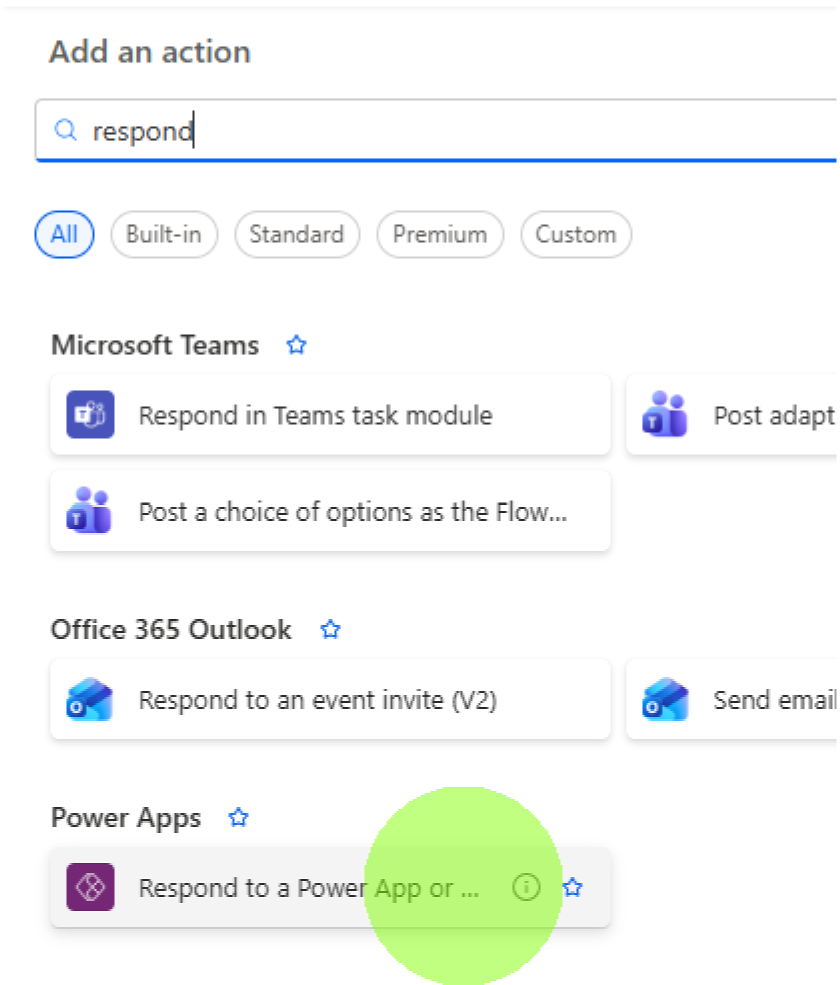




2. Add a new stage Post adaptive card and wait for a response under Teams connector. Message will come from Adaptive Card payload body we copied in Task 2.
   Change version to “1.5” manually as Teams cannot display with “1.6”.

3. Do a test run to see how the response body looks like.

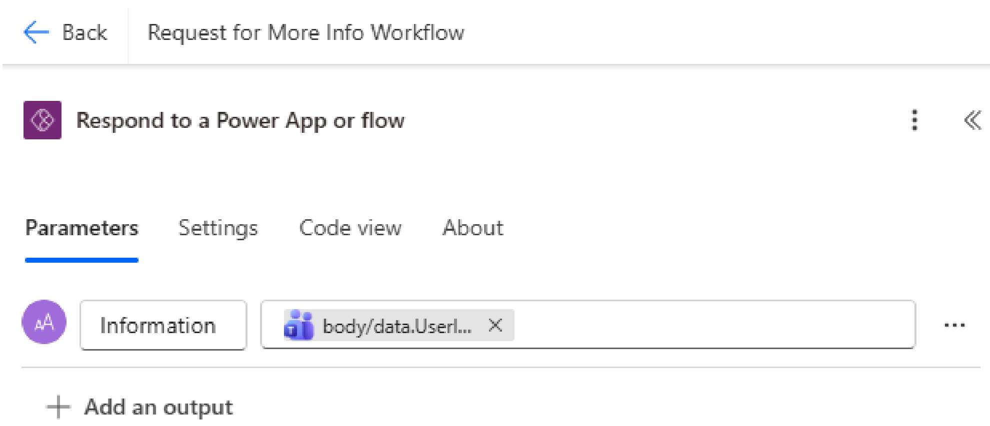
4. Add one more step Respond to a Power App or flow.



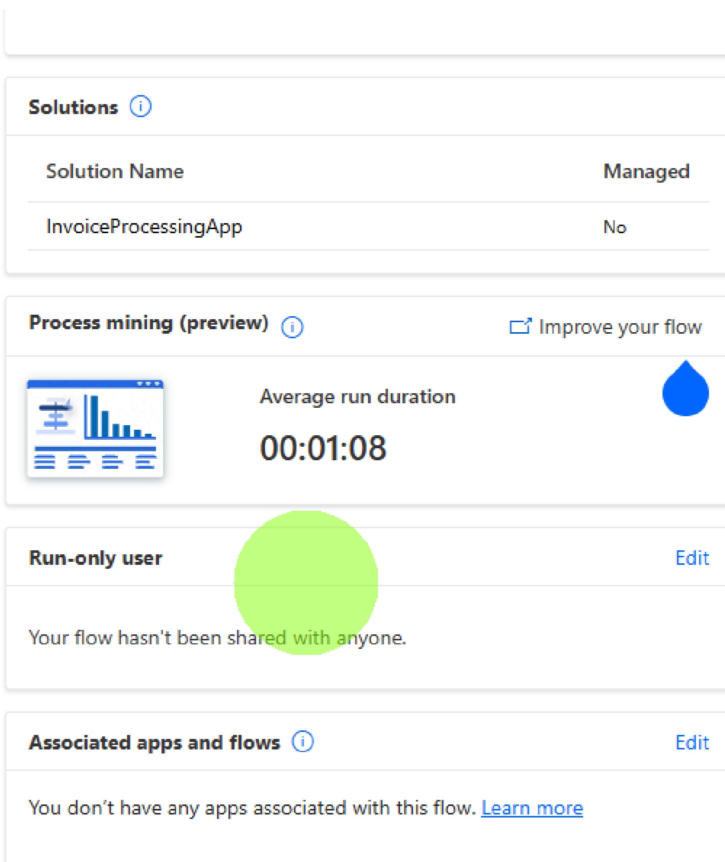
5. Create an output variable named Information and paste in the formula below.

outputs('Post_adaptive_card_and_wait_for_a_response')?['body/data']?['UserInput']

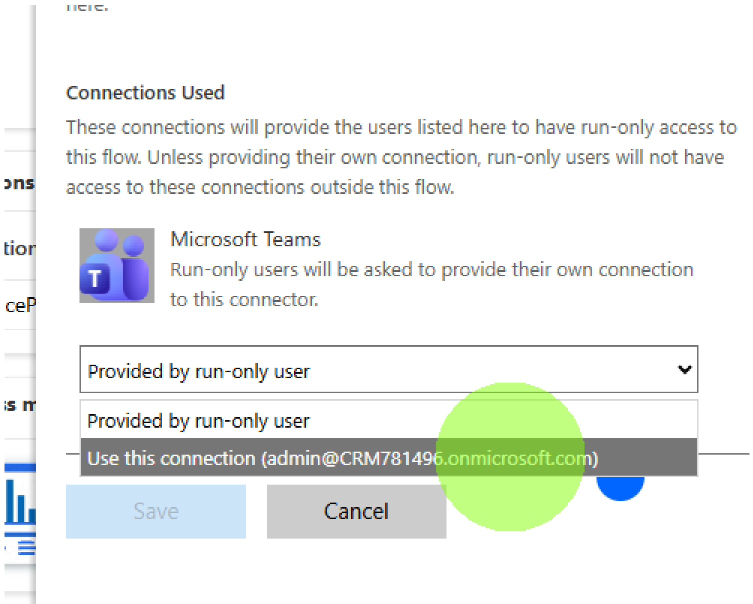
Note ?['UserInput'] requires manual typing as Power Automate doesn't provide the value automatically.



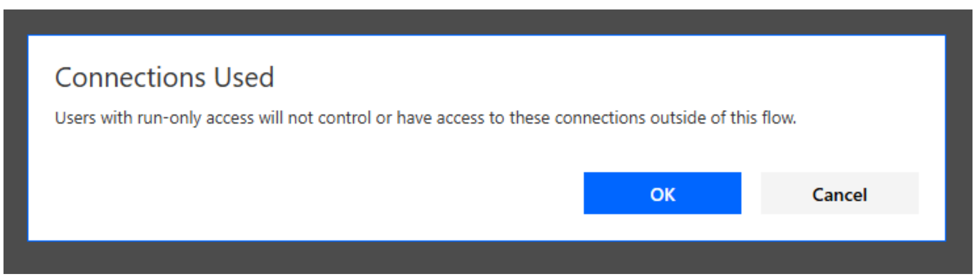
6. Go back to Details page of the flow. Scroll down to look for Run-only user on the right-hand side of the screen.



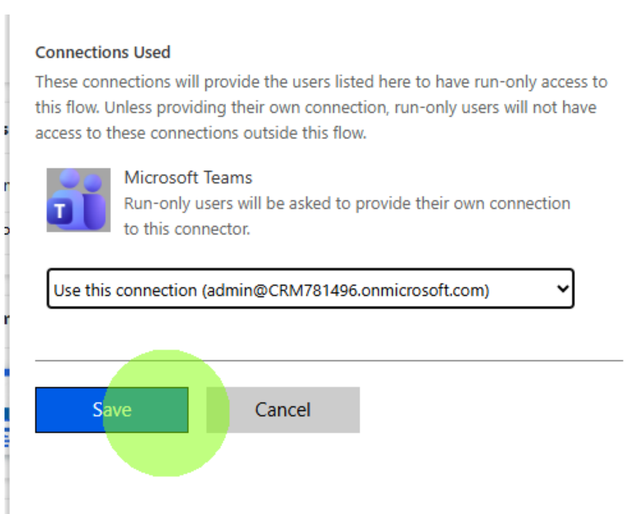
7. Click edit and change the connection used for Microsoft Teams dropdown value to a connection (Use this connection .....).



8. Click OK when prompted.



9. Click Save.

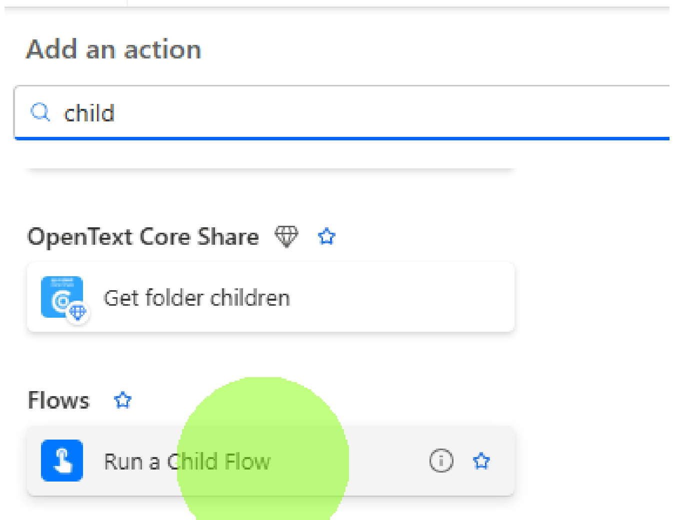


-- **End of Task** --

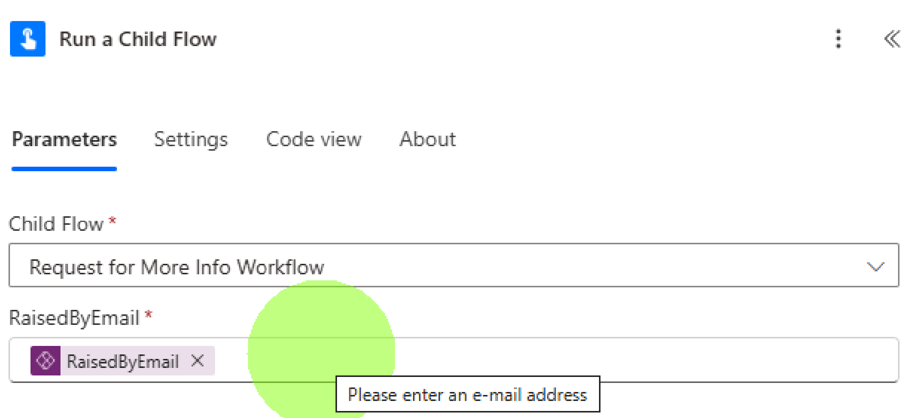
Microsoft Power Platform

Task 4: Add Child Flow to the Main Flow

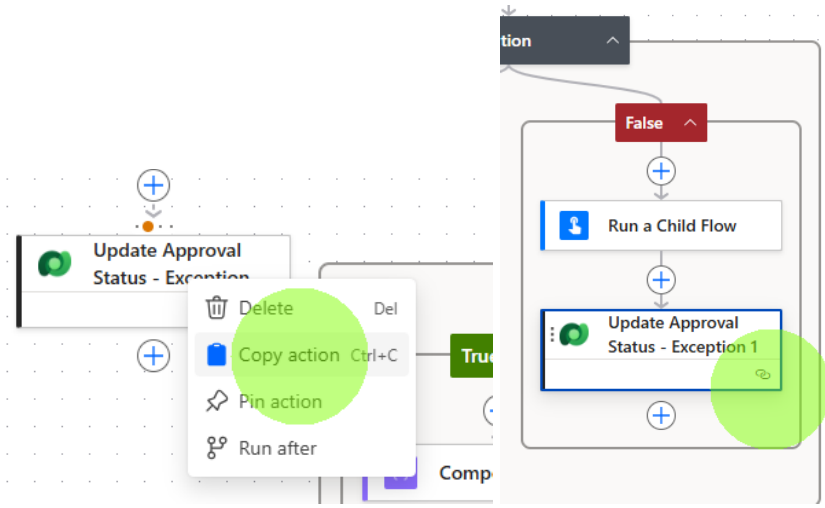
1. Go back to Invoice Approval Workflow.
2. Under False condition, add **Run a child flow**.



3. Select the child flow we have created in Task 3 and provide input parameter.

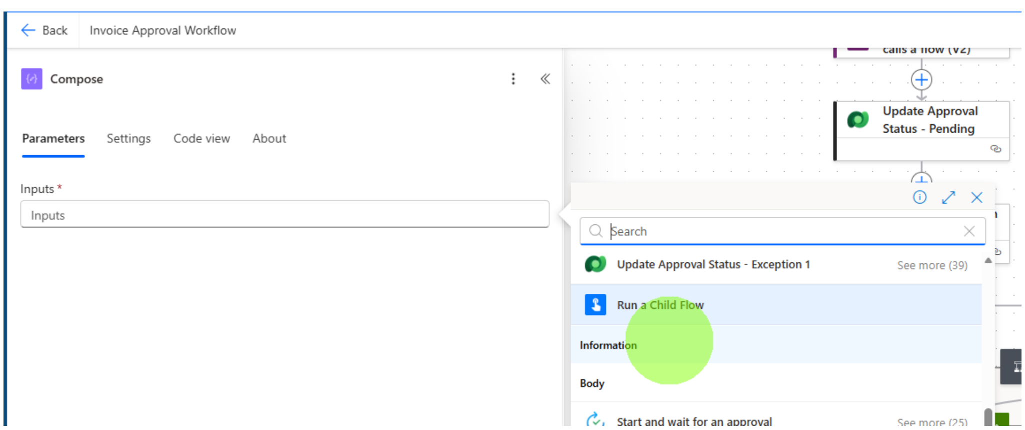


4. Copy Update Approval Status – Exception and paste after child flow stage.



Note: For simplicity of the lab exercise, it's just updating back the Dataverse record with exception status. In real life, you can design the workflow to trigger another round of approval along with the more information provided by the user.

5. Add a compose stage to view the information returned from the child flow.



End of Task