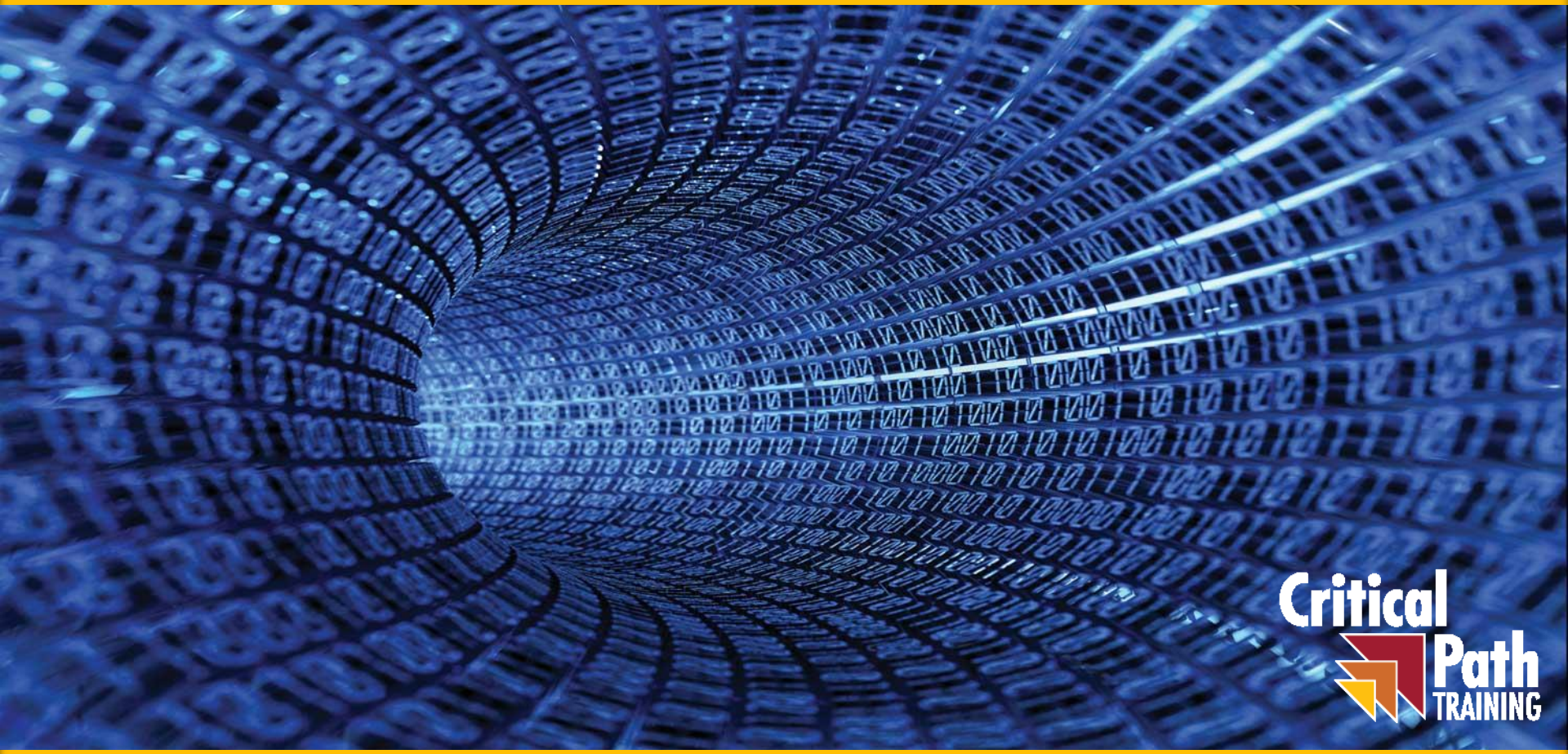
Building Flows to Manage Content and Approvals

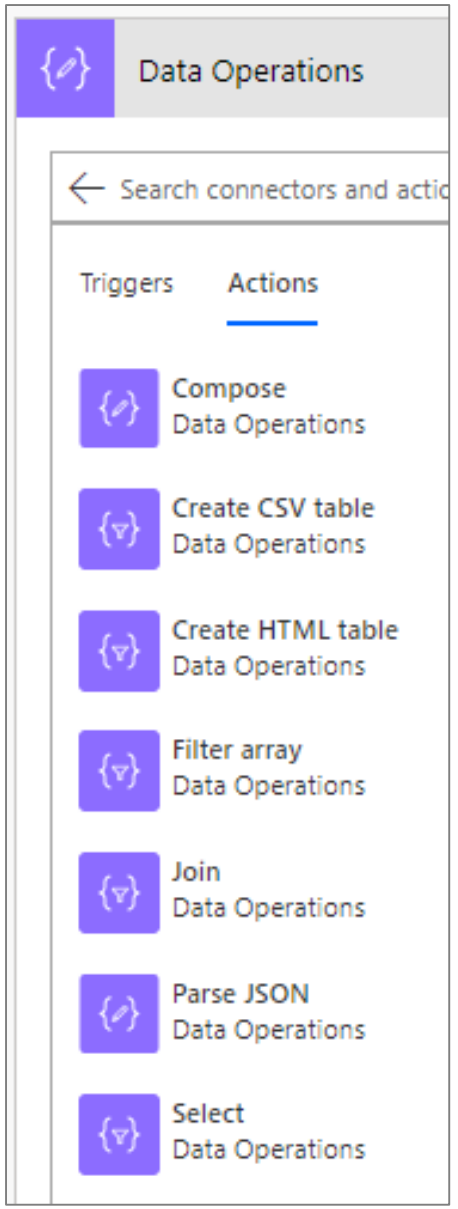


Agenda

- Converting and Reshaping Data
- Uploading Photos to SharePoint
- Automating Approval Processes
- Integrating Flow with Microsoft Forms
- Handling Runtime Errors
- Understanding Parallel Execution

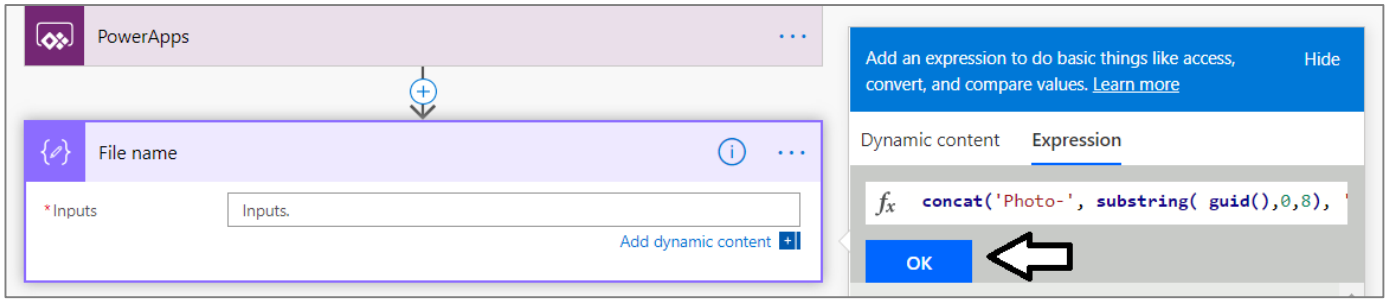
Actions with Data Operations

- Compose
  - Used to parse together text, HTML, JSON, etcs.
- Create CSV Table
  - Used to export data files with CSV format
- Create HTML Table
  - Used to create HTML tables for use in email messages
- Filter Array
  - Filter the contents of an existing array
- Join
  - Parse email address array into single string
- Parse JSON
  - Parse JSON returned from HTTP call into an object
- Select
  - Remap the columns in an array

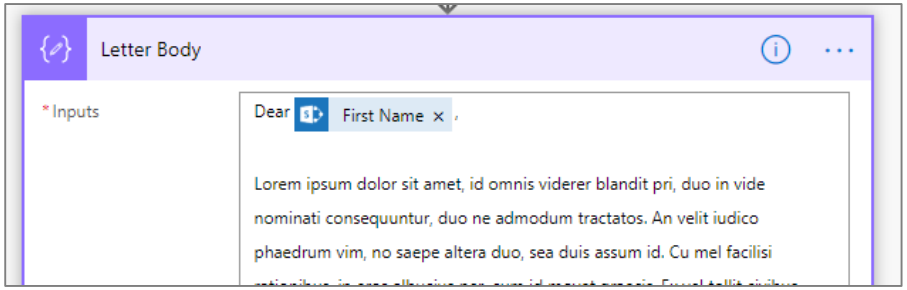


Making Flows Maintainable with Compose

- Compose actions used to simplify flow building
  - Separates parsing content into its own named action
  - Makes flows easier to read and understand
  - Example 1: use Compose step to create a file name

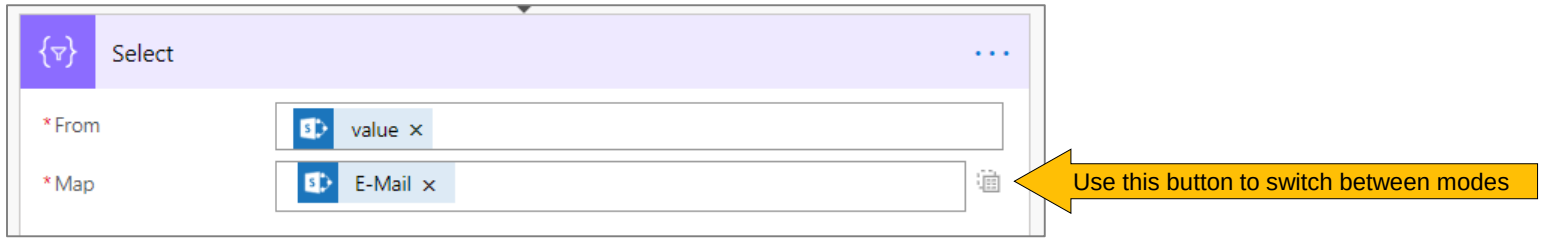


  - Example 2: use compose step to generate content for letter

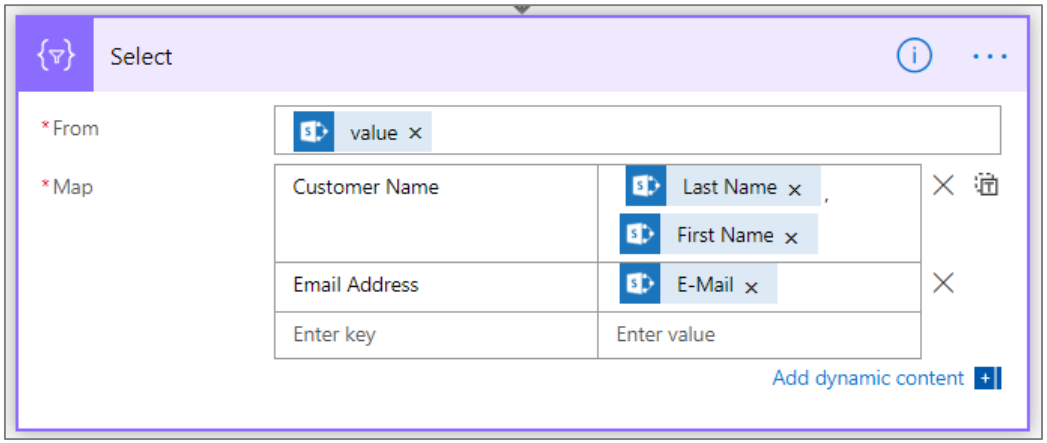


Transforming Arrays using a Select Action

- The Select action configured to **Text mode** or **Key-Value mode**
- Text mode used to create a simple array
  - Example: use Text mode to create an array of email addresses

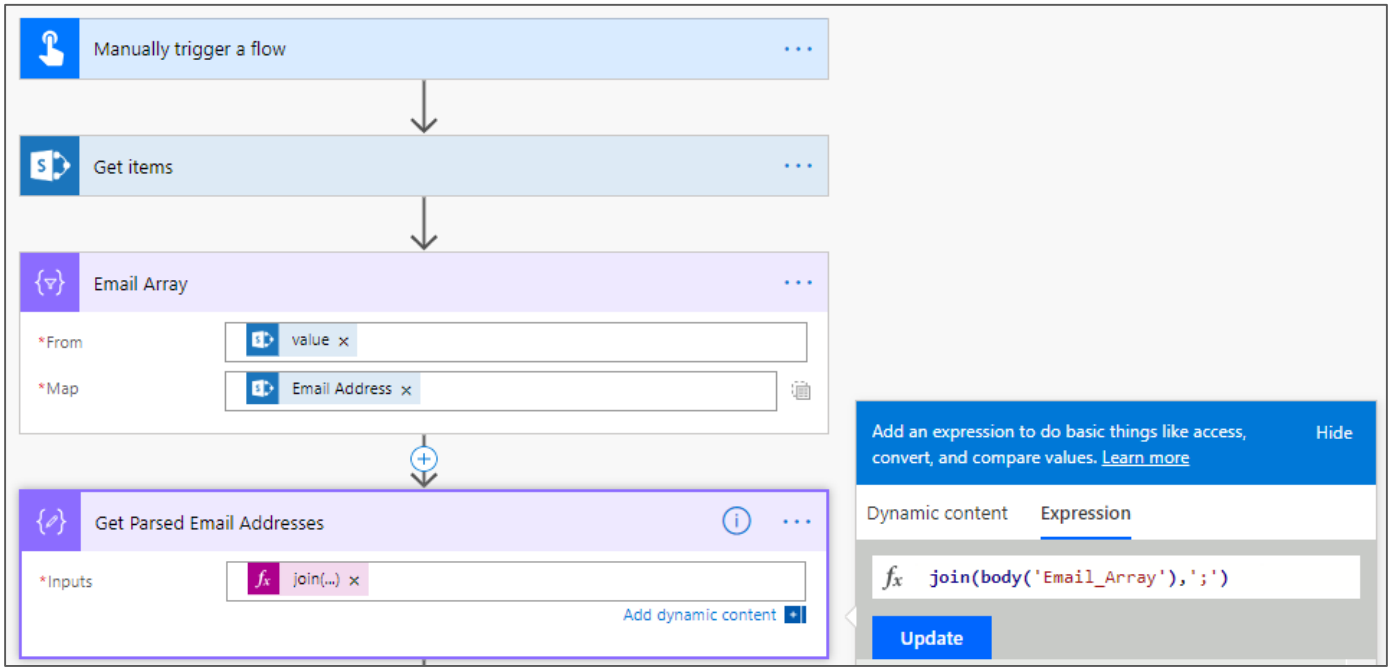


- Key-Value mode allows you to remap columns from table or list
  - Example use Key-Value mode to remap columns from Customers list



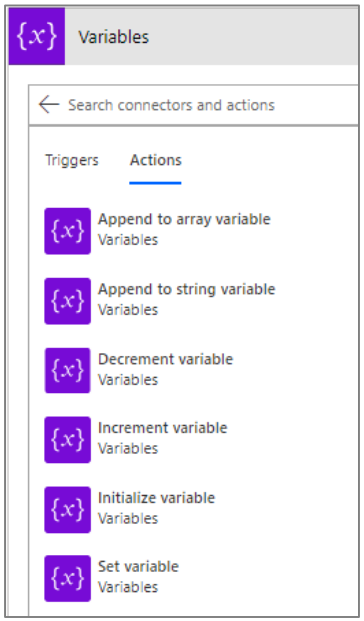
Convert Array of Email Addresses to String

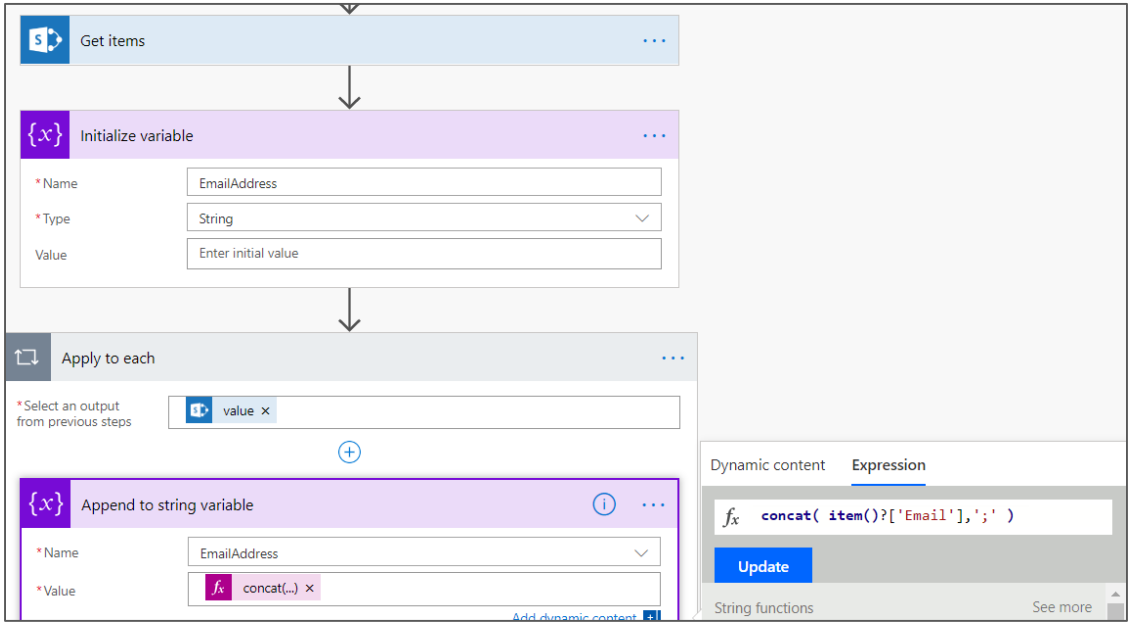
- Steps to create a semicolon-delimited list of emails
  - Use **Get Items** action to get SharePoint Customers list
  - Use **Select** action to create array of email addresses
  - Use **Compose** action to build string from array of email addresses
  - This string can be used in **To** field of **Send an Email** action



Tracking State using Variables

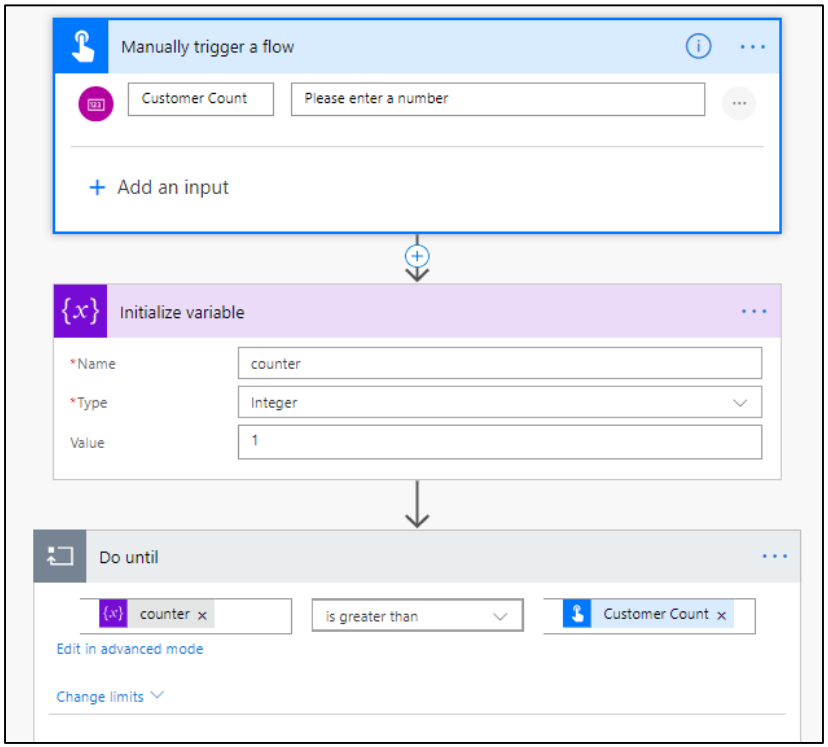
- Variables used to track state during flow lifetime
  - **Initialize Variable** action used to create variable instance
  - Other variable actions uses to update variable values
  - By default, variable value persisted for lifetime of flow
  - Variables can be initialized inside **Scope** action to reduce lifetime





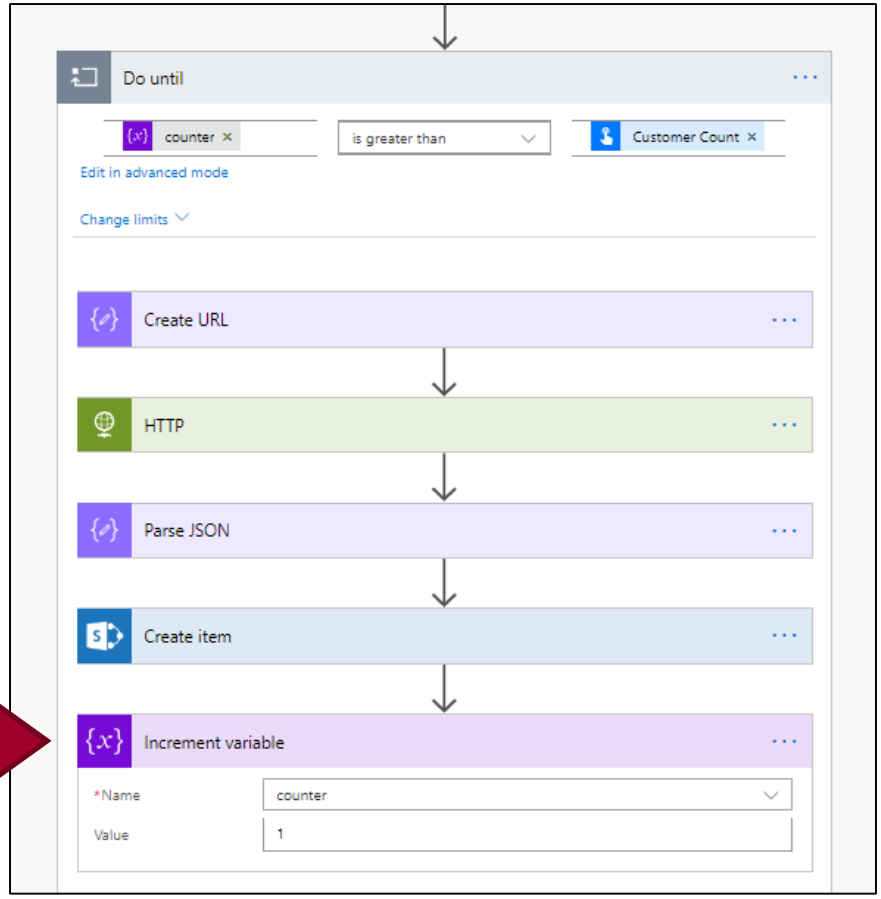
Do Until Action with Counter Variable

- Using a variable to control looping in **Do until** action
  - Initialize integer variable before **Do until** action to act as counter
  - **Do until** action condition checks variable value
  - **Do until** action body requires action to increment variable value



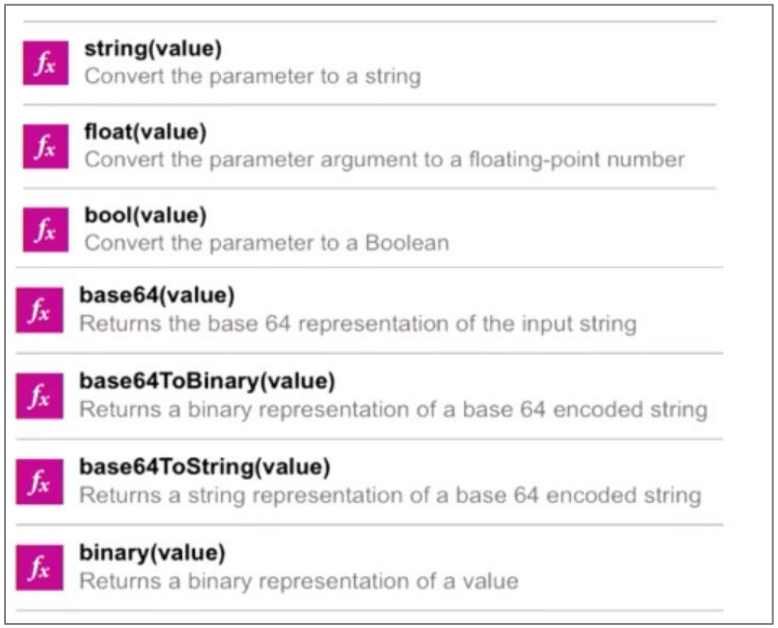
Executing Operations inside Do Until Loop

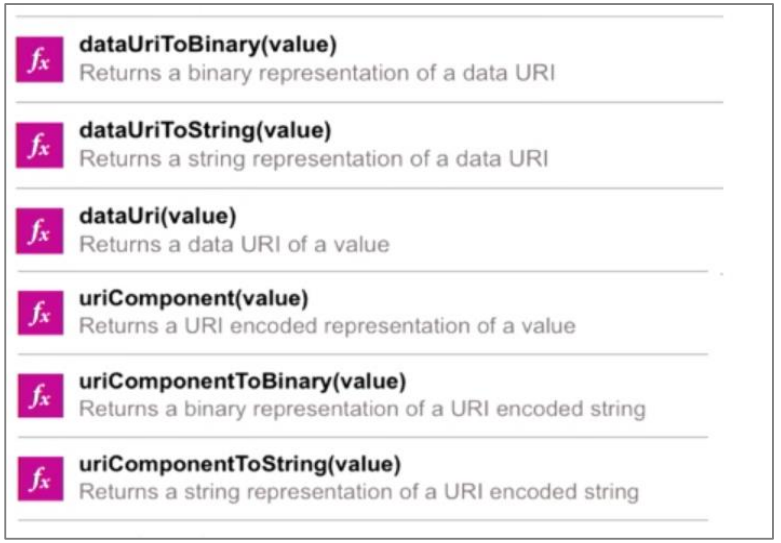
- Implementation of **Do until** action body
  - **Increment variable** action used to change variable value



Handling Type Conversion

- Some conversion is automatic
  - Sometimes conversions are performed for you
  - In other cases, you must explicitly convert between types

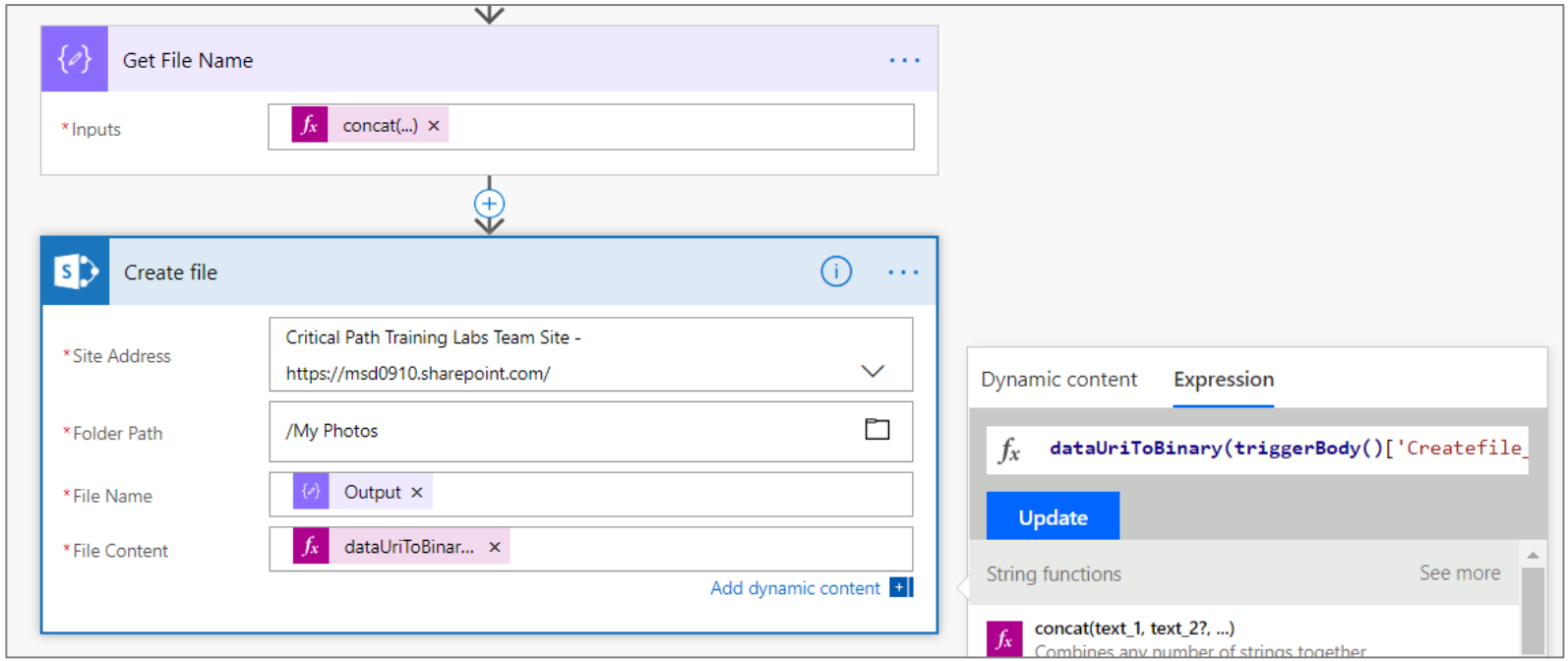




dataUriToBinary()

- PowerApps photos require conversion
  - Allows you to upload phots to SharePoint
  - Accomplished using **dataUriToBinary()** function

```
dataUriToBinary(triggerBody()['Createfile_FileContent'])
```

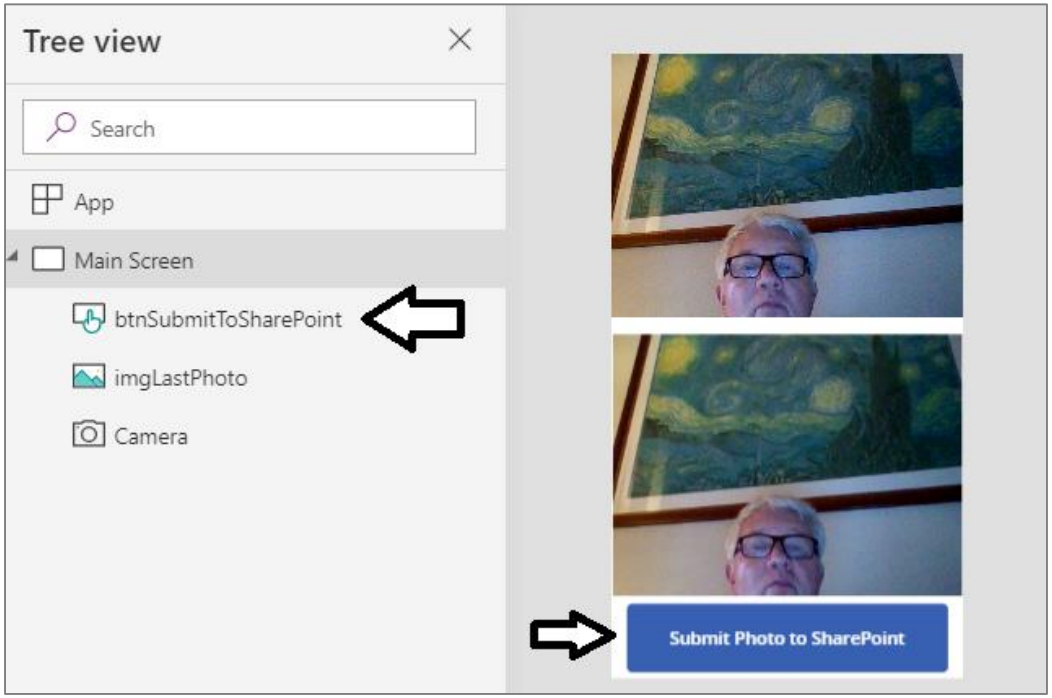


Agenda

- ✓ Converting and Reshaping Data
- ➢ Uploading Photos to SharePoint
- Automating Approval Processes
- Integrating Flow with Microsoft Forms
- Handling Runtime Errors
- Understanding Parallel Execution

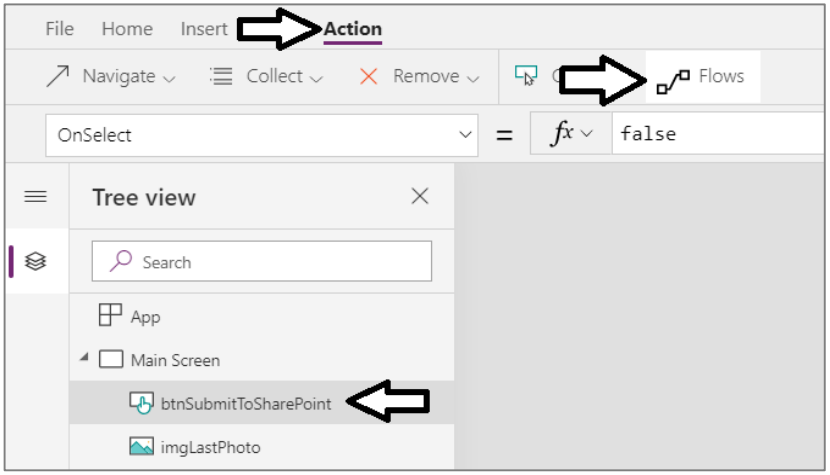
Creating a Canvas App to Upload Photos

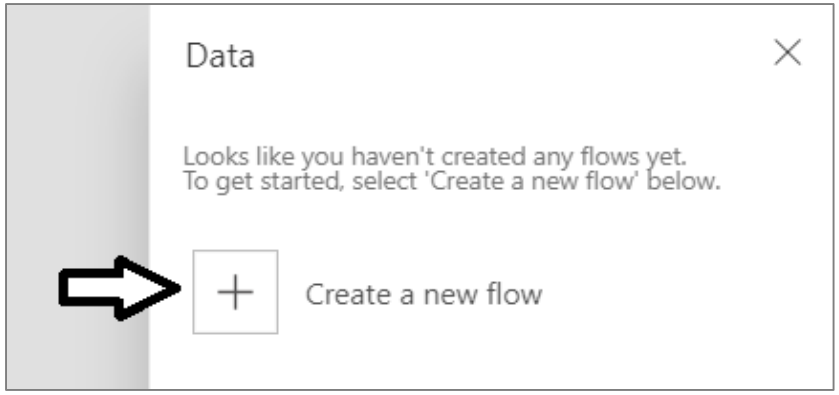
- Begin by creating a blank canvas app
  - Add a Camera control
  - Add an Image control to display camera's last photo
  - Add a button control to trigger a flow to upload photo



Creating a Flow with a PowerApps Trigger

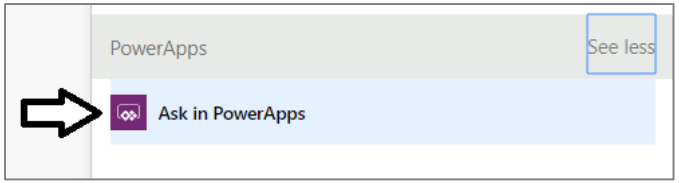
- Create a new flow from the Canvas App editor
  - Select the button which should trigger the flow
  - Click Flows button in Action tab to display the Data pane
  - Click the **Create a new flow** button
  - The new flow automatically created with PowerApps button trigger



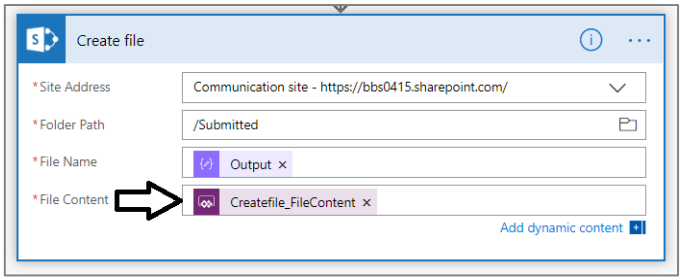


Adding Parameters with Ask In PowerApps

- PowerApps trigger provides Ask in PowerApps parameter
  - Clicking **Ask in PowerApps** automatically creates new parameter

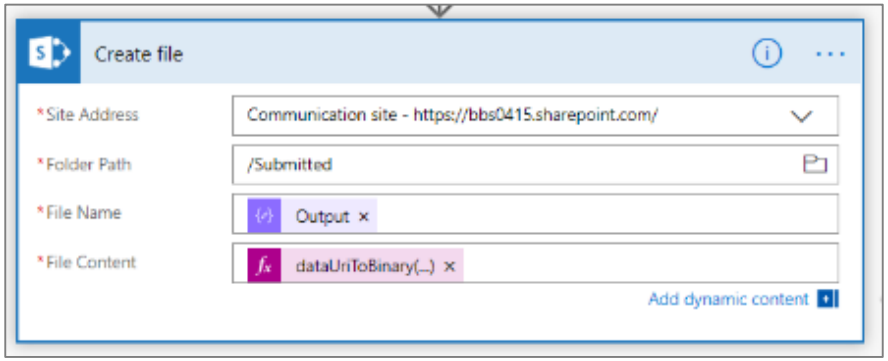


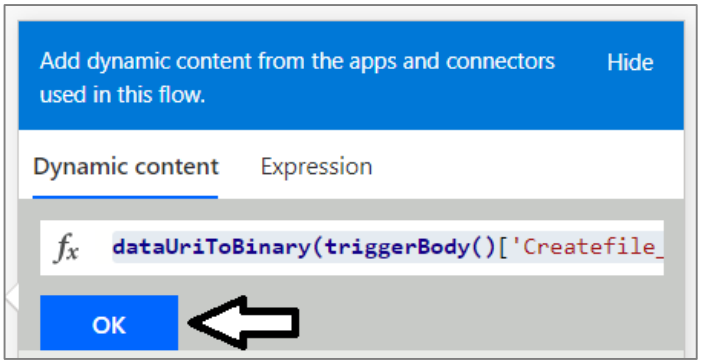
  - New parameters used to pass data from canvas app to flow

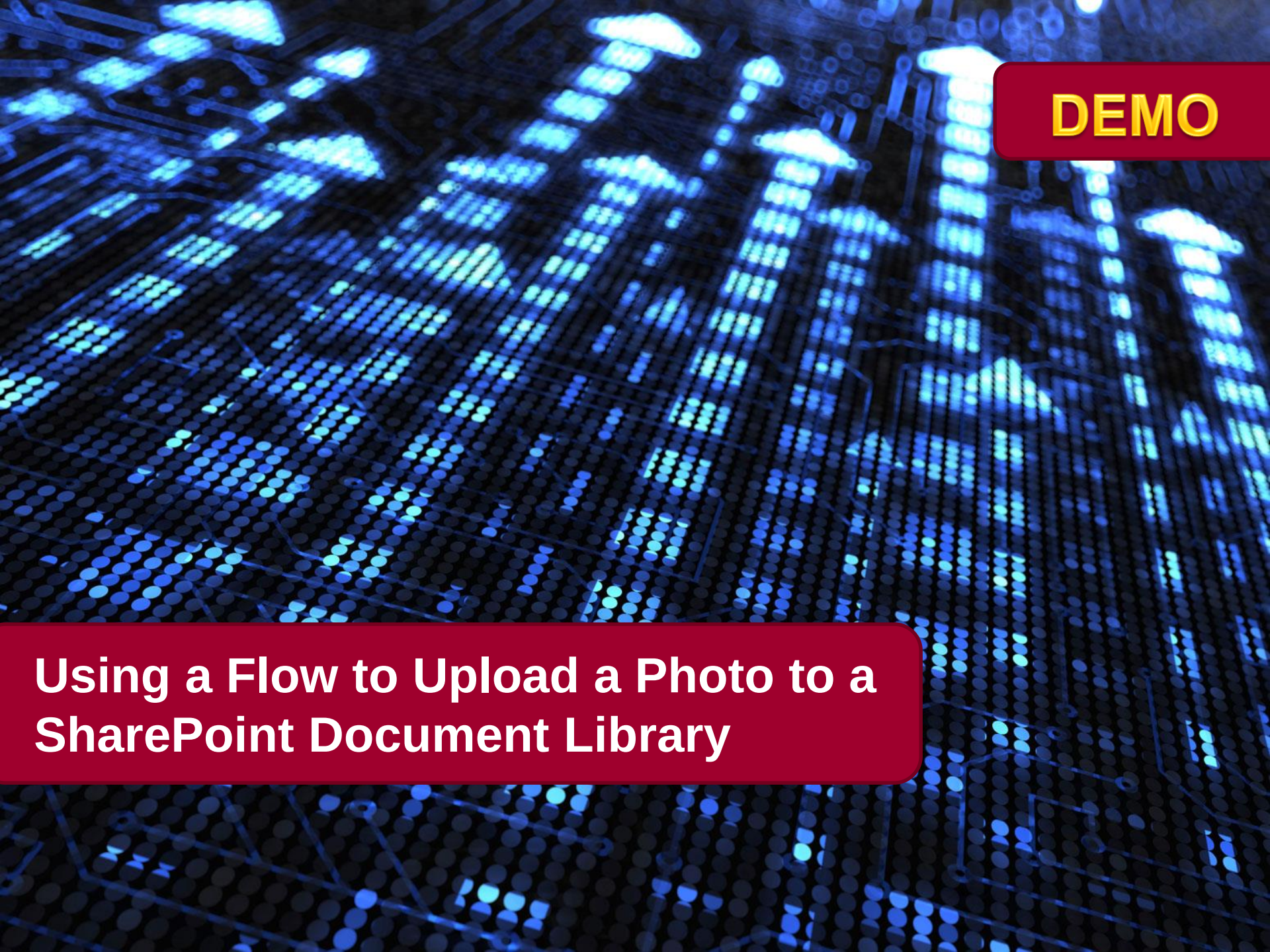


Converting Photos to Binary Format

- Camera control image not compatible with SharePoint
  - Camera control photo image based on Data Uri format
  - SharePoint document library expects files in binary format
  - You must convert photos using **dataUriToBinary** function





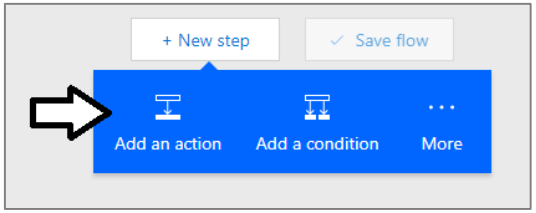

DEMO
Using a Flow to Upload a Photo to a SharePoint Document Library

Agenda

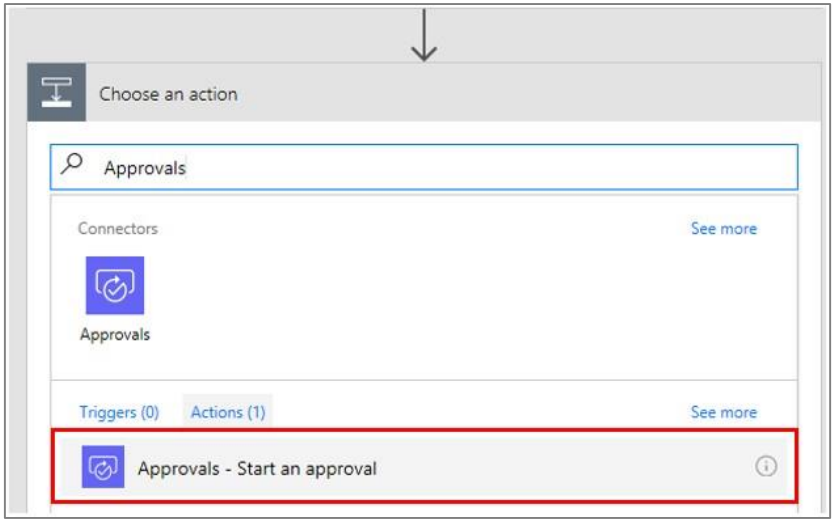
- ✓ Converting and Reshaping Data
- ✓ Uploading Photos to SharePoint
- ➢ Automating Approval Processes
- Integrating Flow with Microsoft Forms
- Handling Runtime Errors
- Understanding Parallel Execution

Add the Start an Approval Action

- An **Approval** process is added as an action

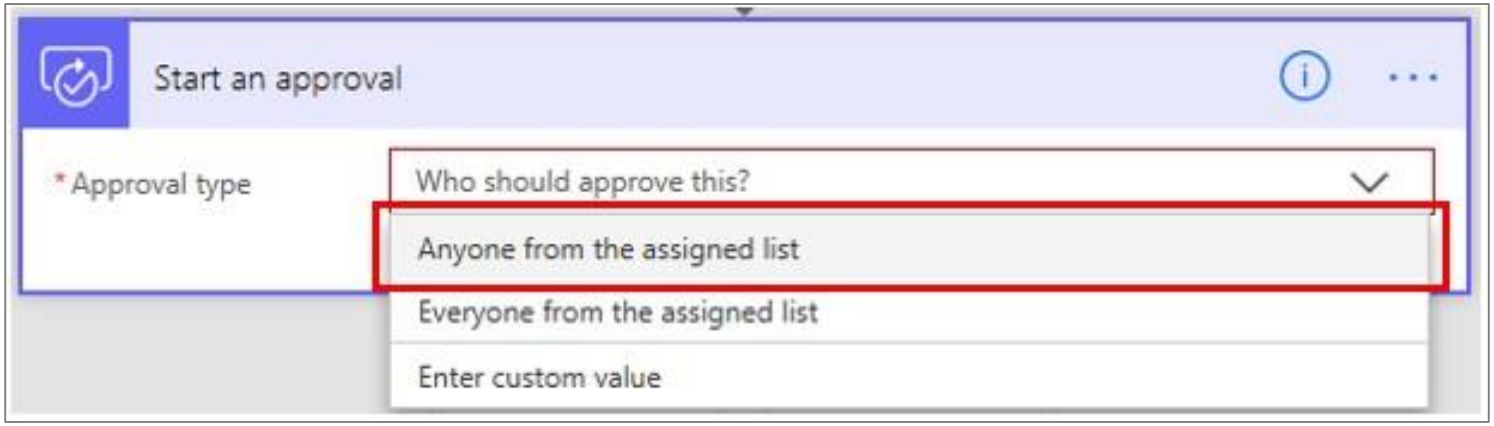


- Select the Approvals action named **Start an approval**



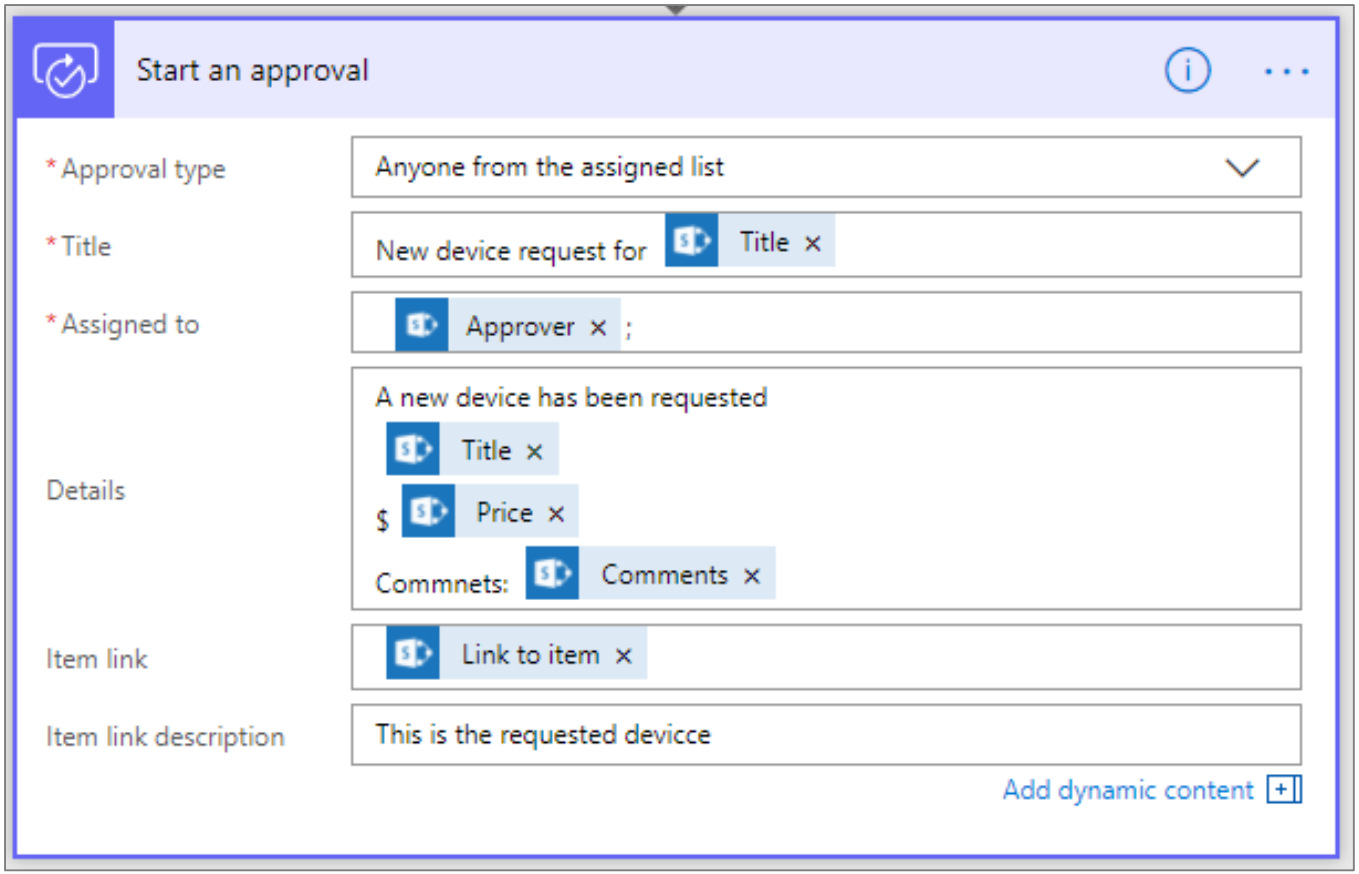
Approval Types

- There are two types of approvals
  - Determine behavior when there are two or more approvers
  - "Anyone" allows single approver to complete approval process
  - "Everyone" requires all approver to approve the request



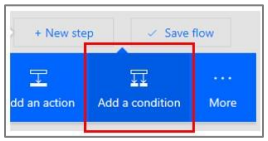
Building Out The Start an Approval Action

- You provide data which is sent to approver

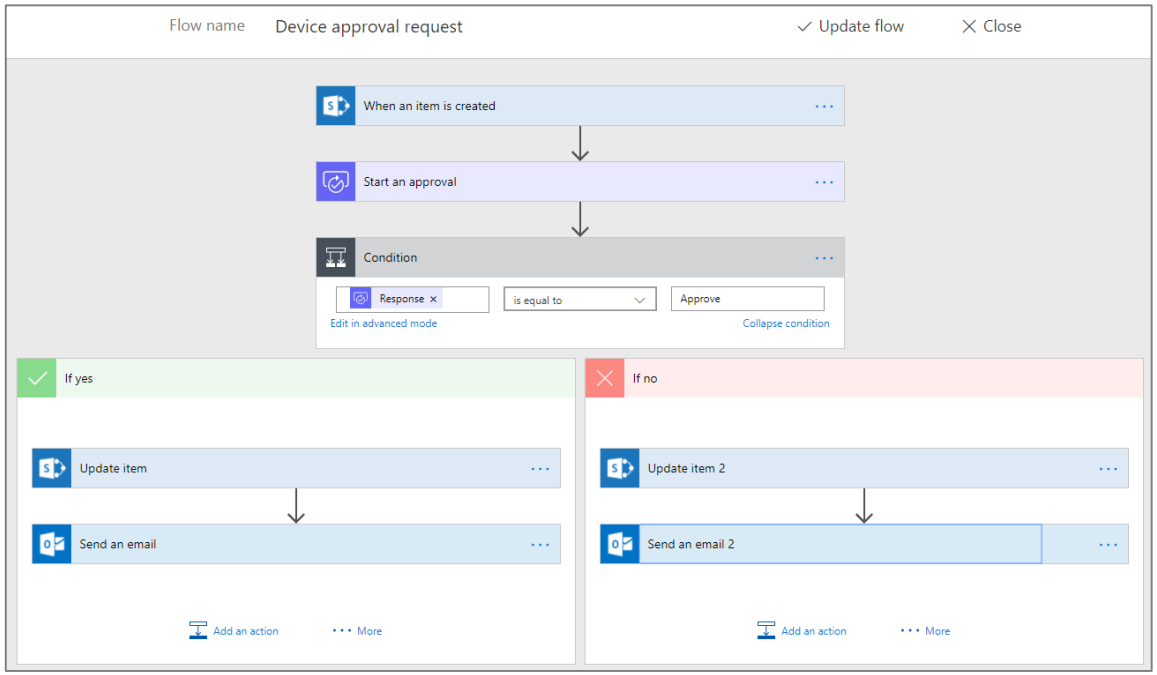


Responding to the Approval Response

- Start an Approval action followed by a condition
  - Allows flow to determine if approval was accepted or rejected

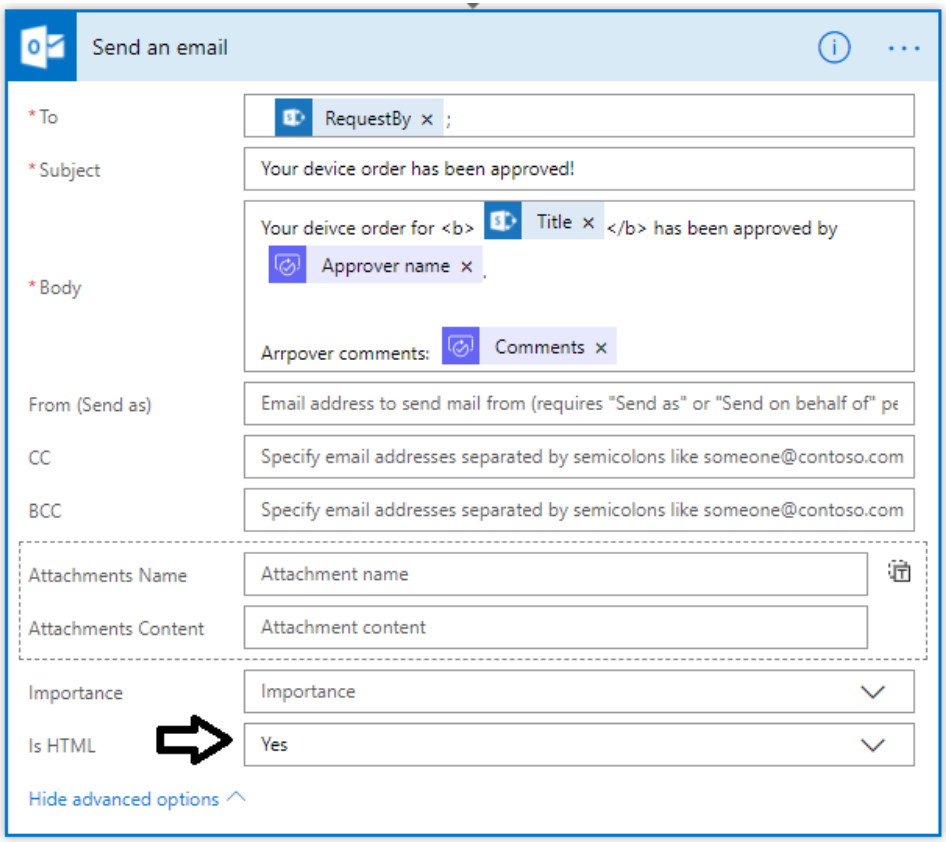


  - Condition has If yes branch and If no branch for both outcomes



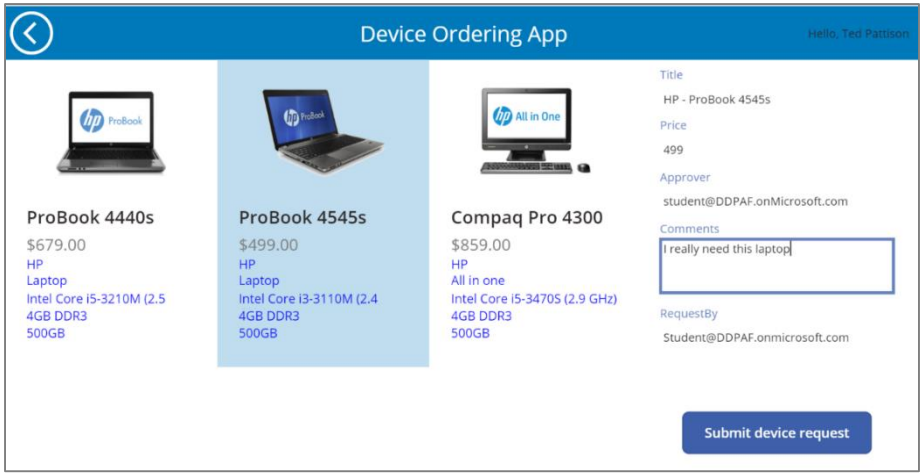
Implementing If yes Branch using Email

- If request is approved, send notification email to requestor

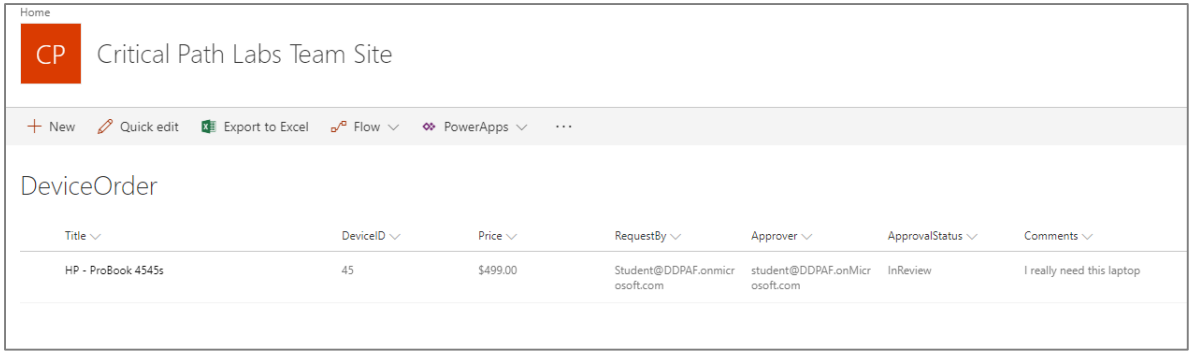


Testing the Approval Flow

- Start by creating a new device request

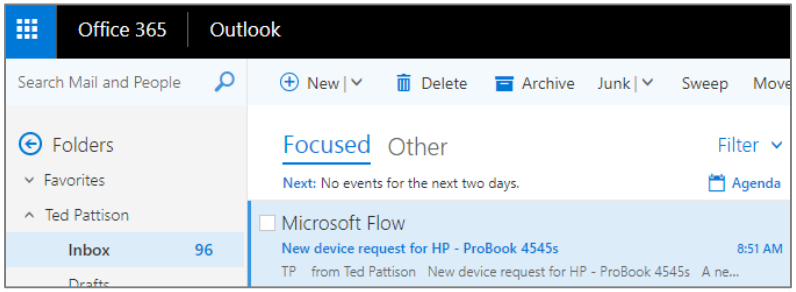


- Adding item to SharePoint list triggers approval process to start

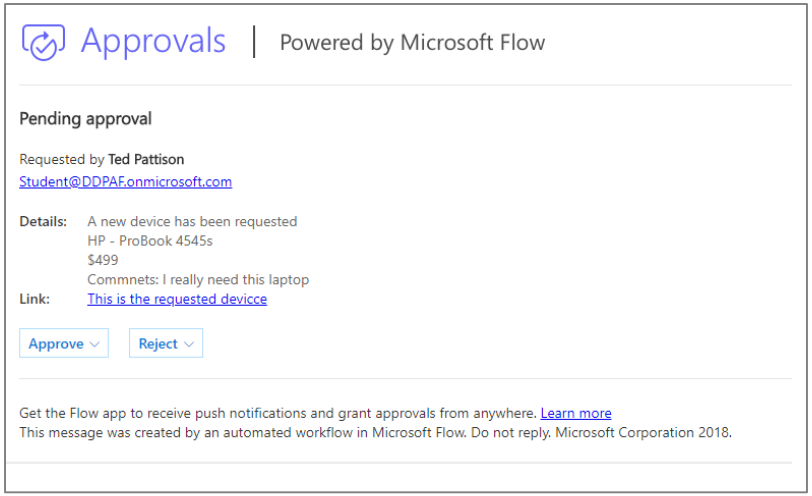


Sending Email Notification to an Approver

- The flow sends notification email to the approver
  - Flow execution currently paused inside **Start an Approval** action



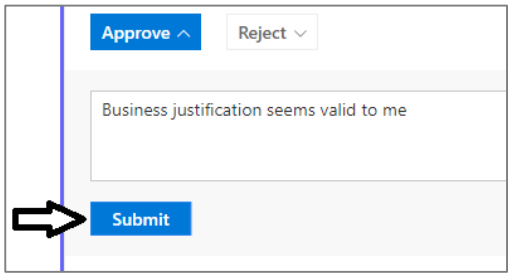
  - Email allows approver to approver or reject approval request



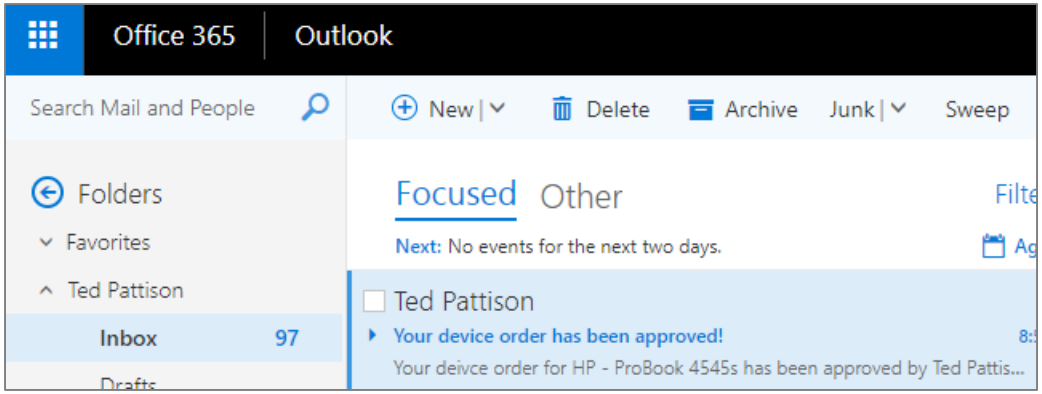
Approving an Approval Request

- Notification email provides button to approve or reject request

- Approver can enter comment and submit approval (or rejection)

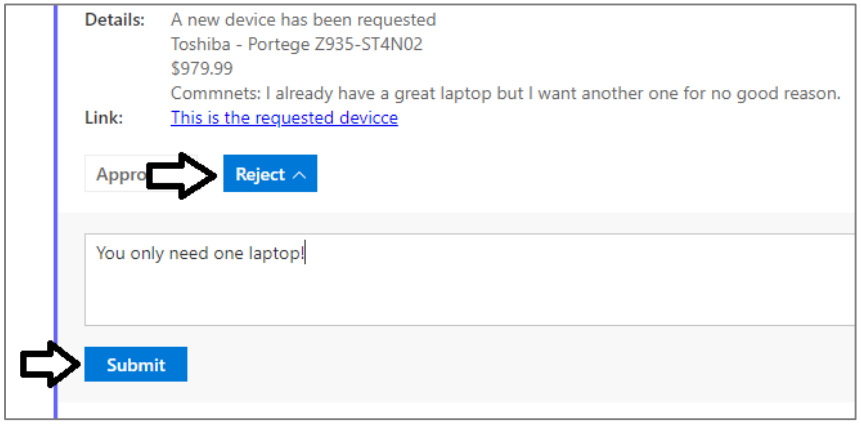


- Approval or rejection unblocks flow which continue down appropriate branch
  - Approval response determines whether to send approval email or rejection email

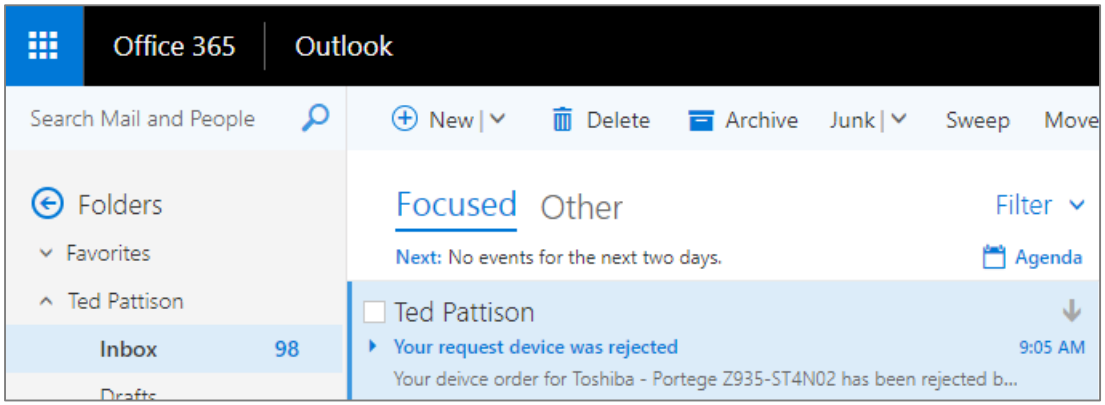


Rejecting an Approval Request

- In the case of a rejected request…

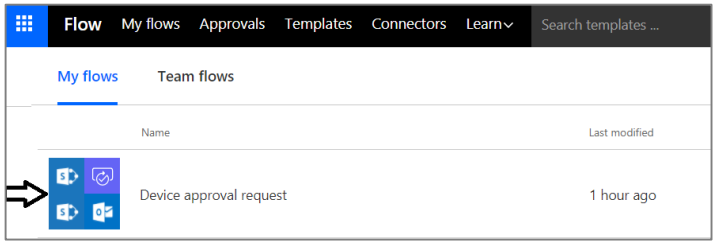


- Approval flow sends a notification about rejection

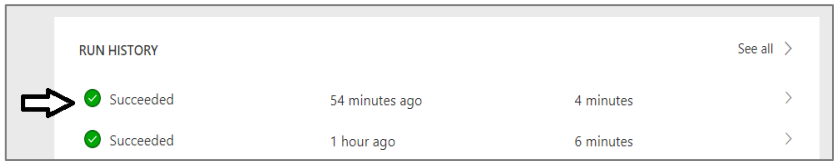


Run History

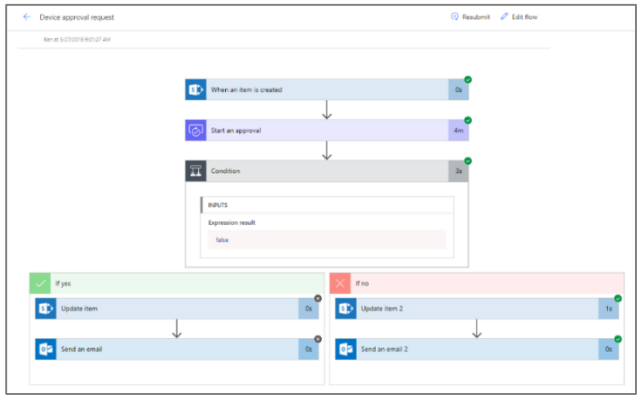
- Flow tracks run history of flow that have started
  - Click on a flow to see its RUN HISTORY list



  - RUN HISTORY list has entry for each flow that has started

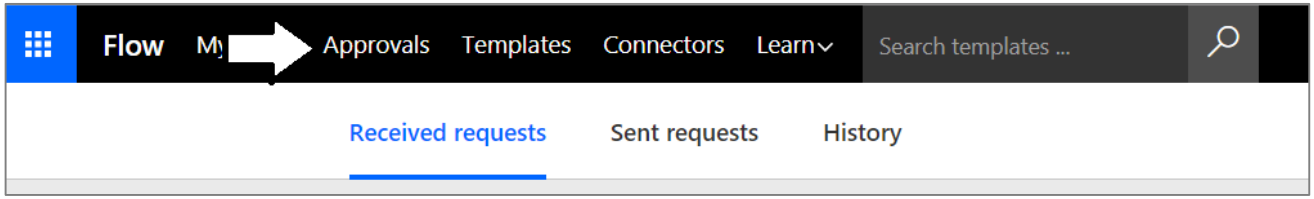


  - Drilling into flow run history shows execution path and data

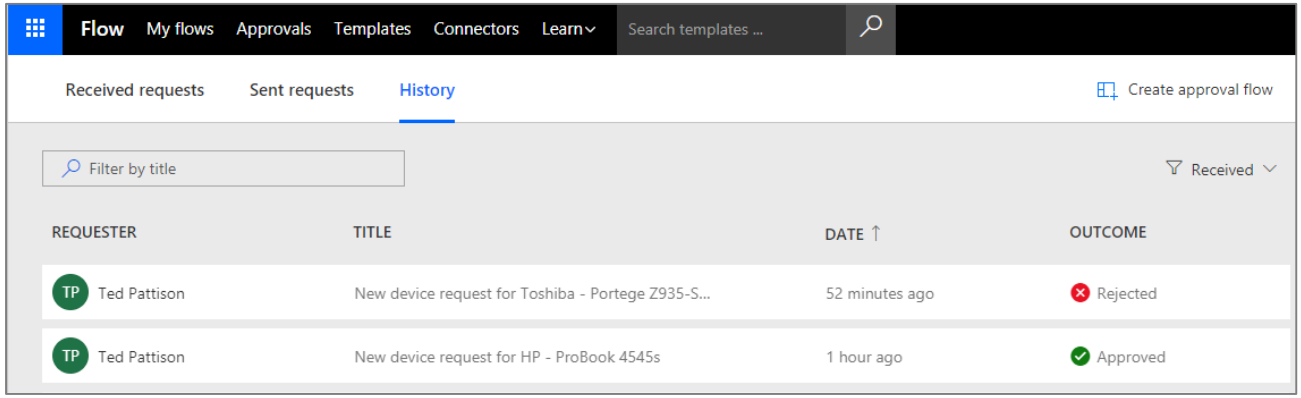


Approvals Center

- Microsoft Flow provides Approvals Center
  - Provides alternative to email for approve/reject processing
  - Accessible through browser

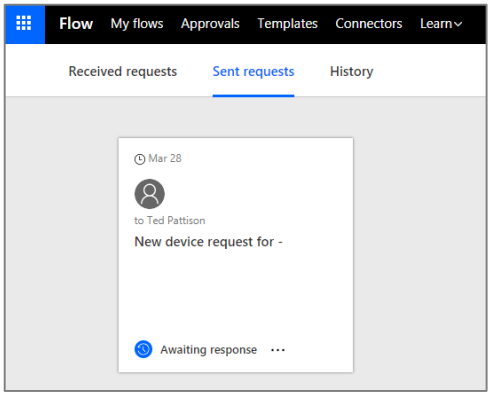


  - Provides monitoring of completed approvals and pending approvals

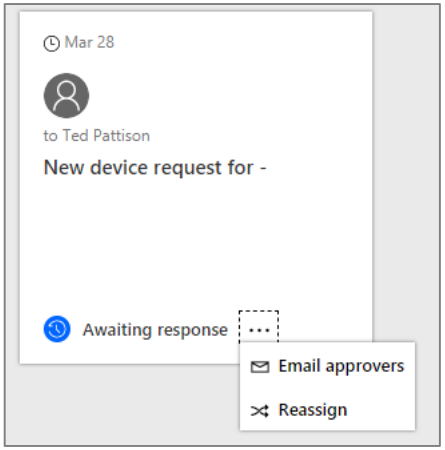


Examining Sent Requests

- Requester can view requests he/she has submitted

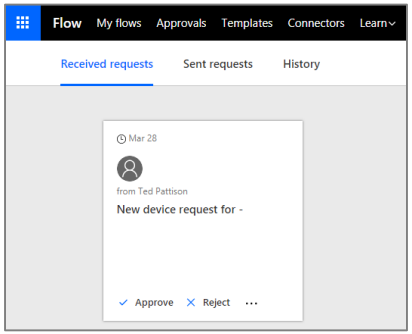


- Requester can email approver(s) or resign to different approver

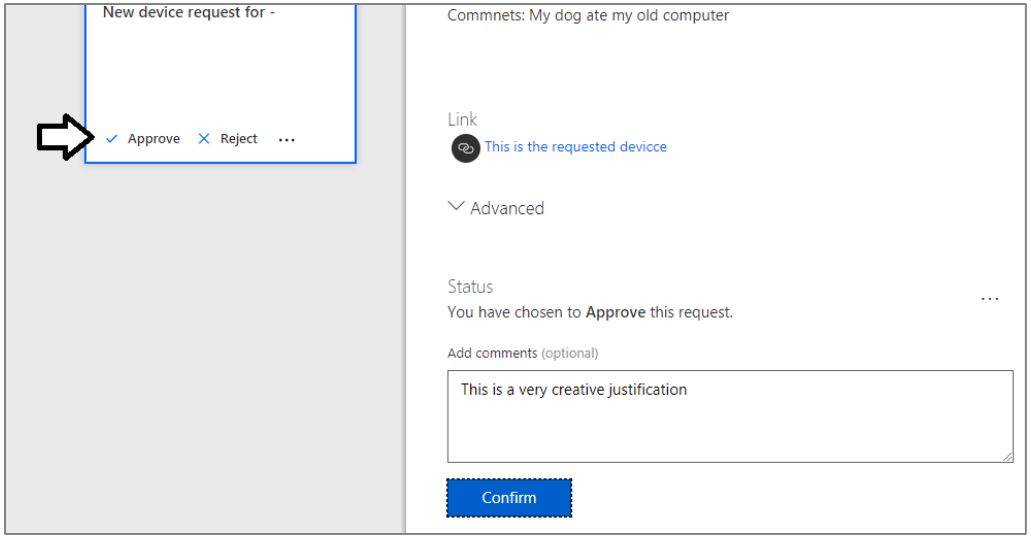


Examining Received Requests

- Approvers can see list of all their approval requests

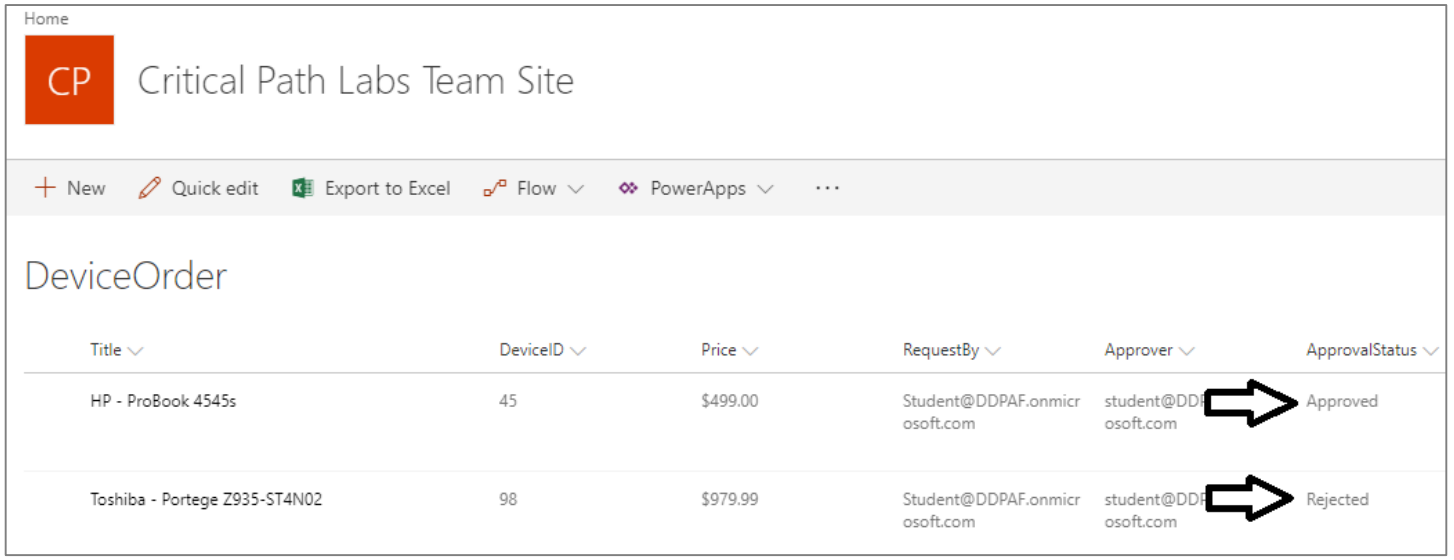


- Approver can approve or reject approval request



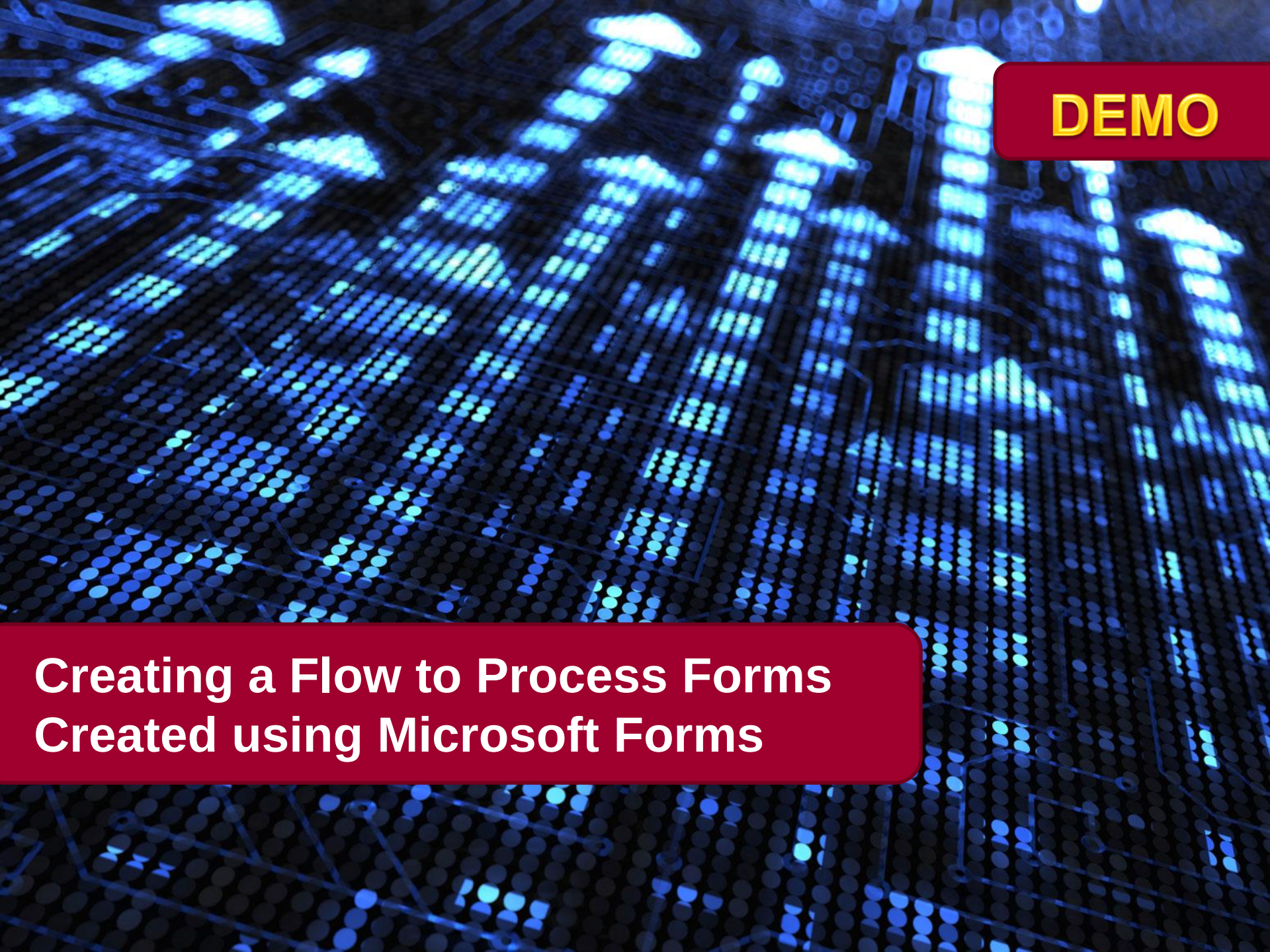
Updating a SharePoint List Item

- Approval flow can update SharePoint list items
  - Used to show SharePoint users the outcome of approval process



Agenda

- ✓ Converting and Reshaping Data
- ✓ Uploading Photos to SharePoint
- ✓ Automating Approval Processes
- ➢ Integrating Flow with Microsoft Forms
- Handling Runtime Errors
- Understanding Parallel Execution


DEMO
Creating a Flow to Process Forms
Created using Microsoft Forms

Agenda

- ✓ Converting and Reshaping Data
- ✓ Uploading Photos to SharePoint
- ✓ Automating Approval Processes
- ✓ Integrating Flow with Microsoft Forms
- ➢ Handling Runtime Errors
- • Understanding Parallel Execution

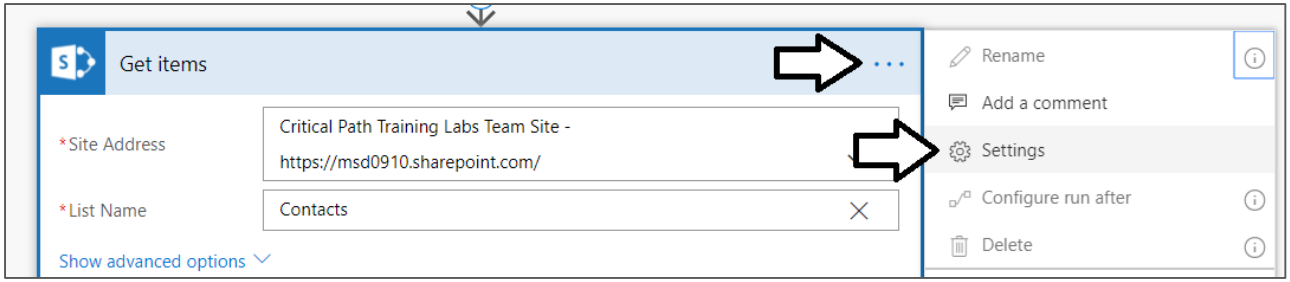
Normal action execution

- Standard behavior of a flow
  - Action steps execute in sequential order
  - Flow terminates if error occurs (failure or timeout)

- After flow runs, every action left in 1 of 4 possible states

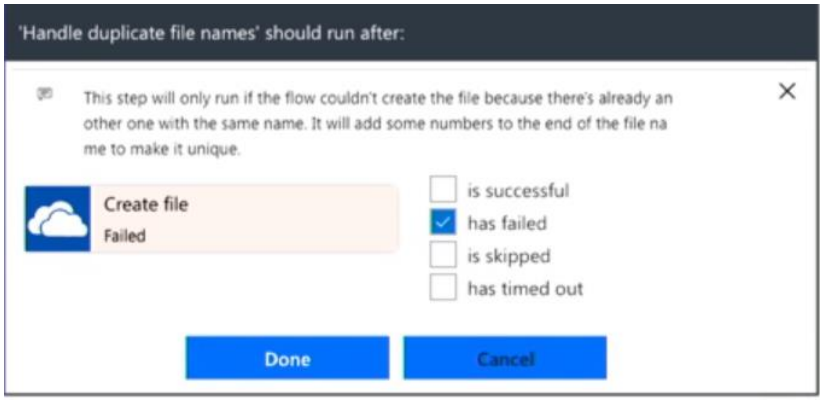
Action Settings

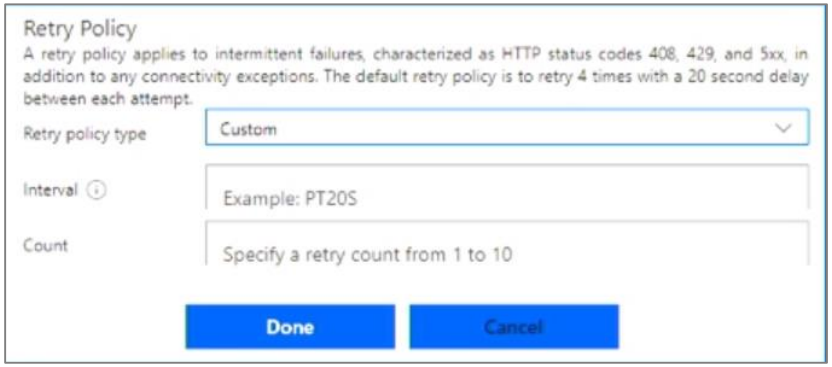
- Settings let you configure
  - Async Actions
  - Timeouts
  - Retry Policy
  - Sequential Behavior
  - And more!



Error Handling

- Select the **Run after** option from action menu
  - Choose which error conditions, the arrow will turn dotted red
  - Use parallels for errors that are not at end of flow
  - Retry policy by default handles transient failures
  - Recommended to select exponential as they last a long time



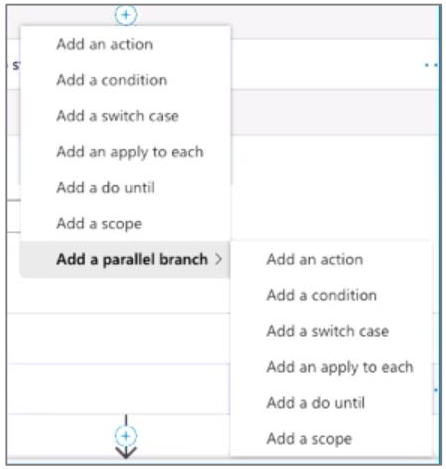


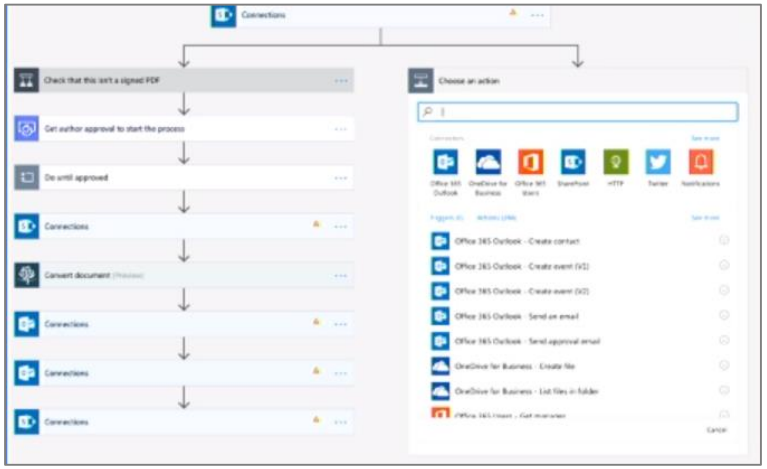
Agenda

- ✓ Converting and Reshaping Data
- ✓ Uploading Photos to SharePoint
- ✓ Automating Approval Processes
- ✓ Integrating Flow with Microsoft Forms
- ✓ Handling Runtime Errors
- ➢ Understanding Parallel Execution

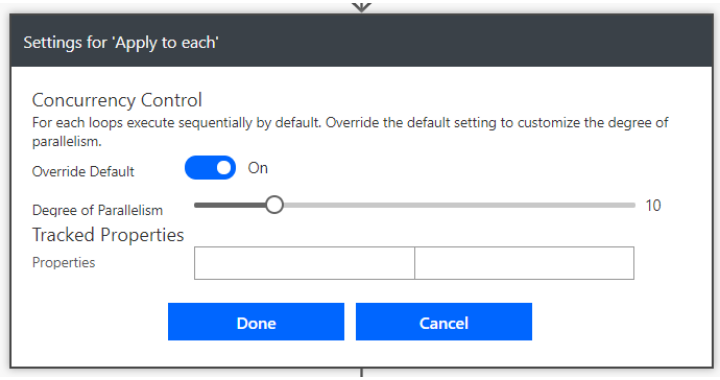
Parallel Execution

- Add parallel branch from above using ⊕





- Apply to each is sequential by default
  - Adding parallel execute to Apply to each



Summary

- ✓ Converting and Reshaping Data
- ✓ Uploading Photos to SharePoint
- ✓ Automating Approval Processes
- ✓ Integrating Flow with Microsoft Forms
- ✓ Handling Runtime Errors
- ✓ Understanding Parallel Execution