Azure Data Factory

Lab : Control Flow Activity with Logic App

Pre-requisites:

- Azure Pass subscription
- Azure Data Lake Storage Gen2 storage account
- Azure Data Factory

Lab Objective:

After completing this lab, you will be able to:

- Use control flow activities
- Send custom email through logic app

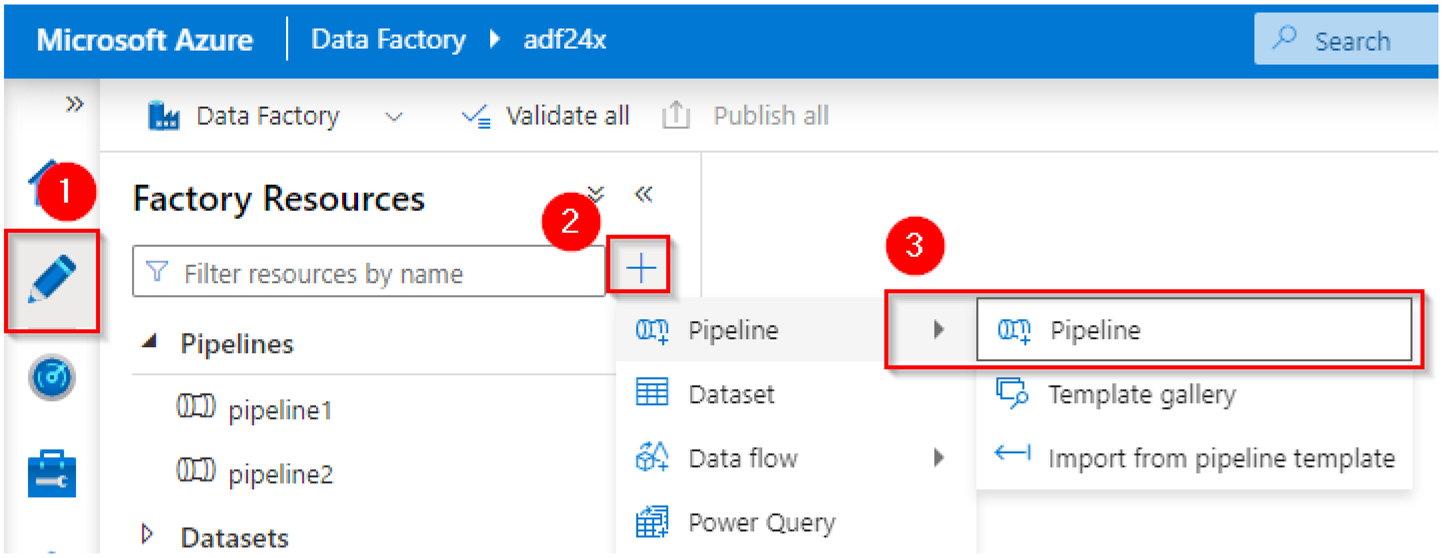
Exercise 1: Create a Pipeline to retrieve the valid list of file from Storage Account.

The task for this exercise are as follows:

1. Create a Pipeline and configure it
2. Append files to array based on modified date and size.

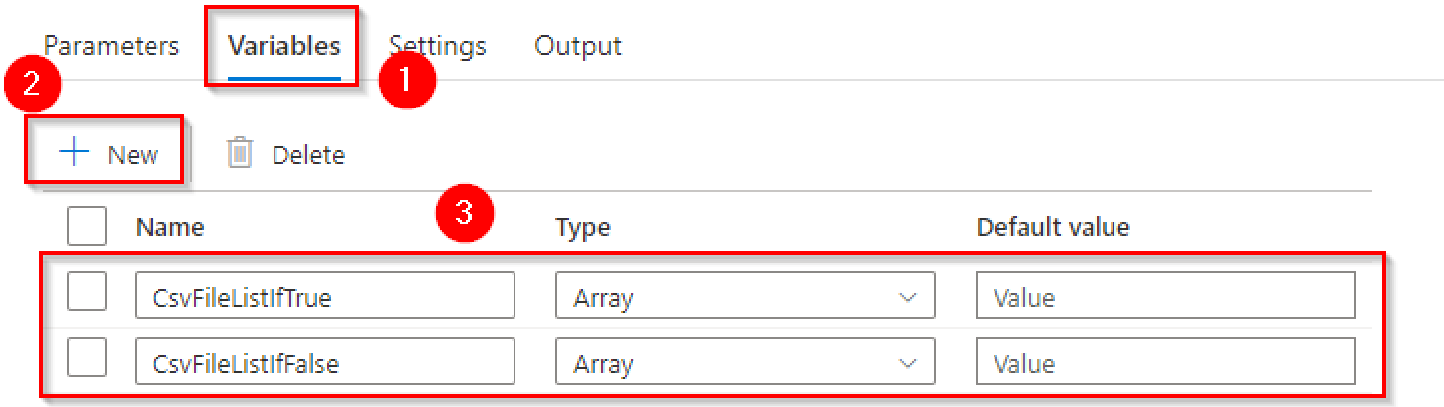
Task 1: Create a Pipeline and configure it.

1. Create a Pipeline in data factory **adfxx**.



2. Create two variables of array type in pipeline as follows:

   a. Name: CsvFileListIfTrue Type: Array
   b. Name: CsvFileListIfFalse Type: Array



	Name	Type	Default value
☐	CsvFileListIfTrue	Array	Value
☐	CsvFileListIfFalse	Array	Value

Task 2: Append files to array based on modified date and size.

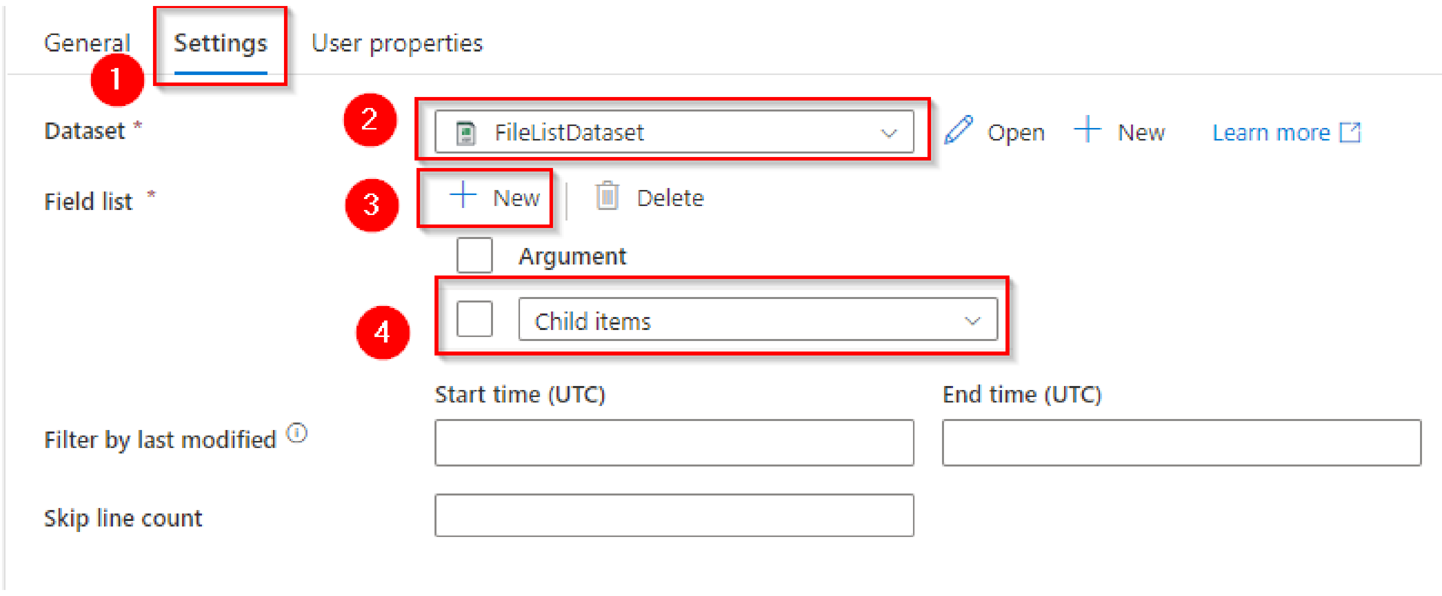
1. Add **Get Metadata** activity from **Activities** canvas under **General** Section to pipeline and configure it setting as:

   a. **Dataset:**

      - **Click + New**
      - **Type:** Azure Data Lake Storage Gen 2
      - **Format**: DelimitedText
      - **Name**: FileListDataset
      - **Linked** Service: AzureDataLakeStorage
      - **File path:** data / output /
      - **First row as header**: Unchecked
      - **Import schema**: None
      - **Click Ok**

   b. **Field list**:

      - Click + New
      - From the drop down select: **Child items**



Dataset * FileListDataset
Field list * Child items
Filter by last modified: Start time (UTC) End time (UTC)
Skip line count

##: you can do the debug run to check the out of Get Metadata activity

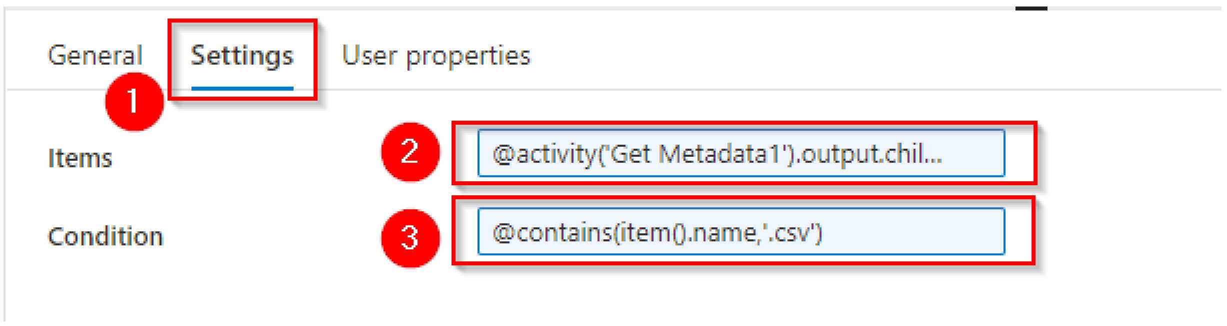
2. Add **Filter** activity to pipeline from activities canvas under **General** section and configure as follows:

   General -> Name: FilterCsvFiles

   Setting:

      Items: @activity('Get Metadata1').output.childItems

      Condition: @contains(item().name,'.csv')



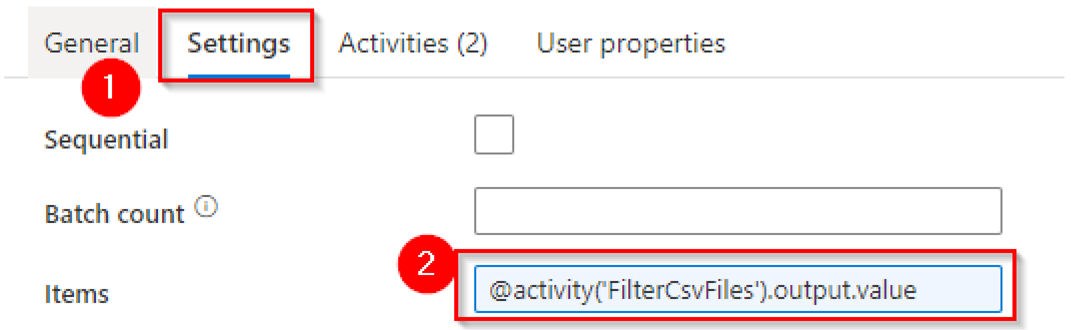
Items @activity('Get Metadata1').output.chil...

Condition @contains(item().name,'.csv')

   Connect Get Metadata activity on success to Filter activity

3. Add **ForEach** activity to pipeline from Activity canvas under **Iteration and conditionals** and configure it as:

   Setting -> items: @activity('FilterCsvFiles').output.value

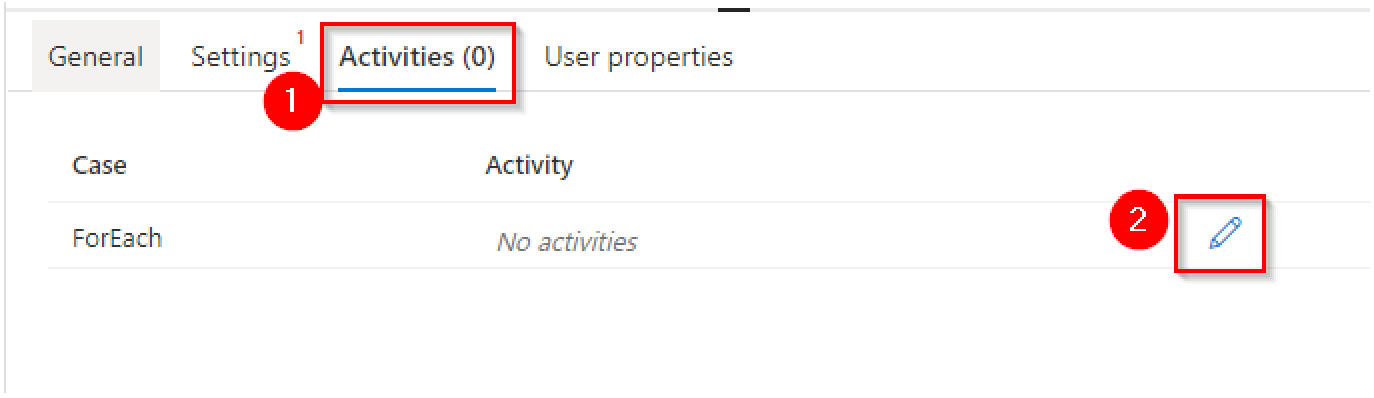


Sequential ☐

Batch count

Items @activity('FilterCsvFiles').output.value

   Activities -> Click on Pencil Icon.



Case	Activity
ForEach	*No activities*

   Connect **Filter Activity** to **ForEach activity**

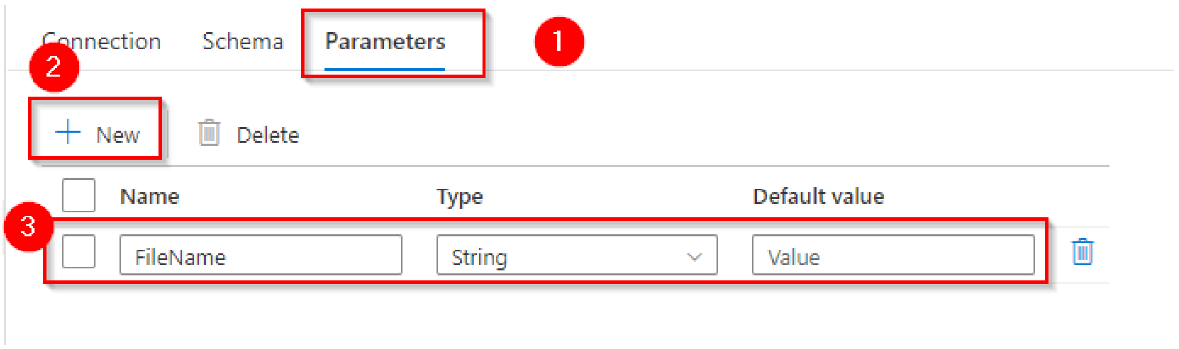
4. Inside the ForEach Activity add Get Metadata and configure it as:

   General -> Name: Get Csv File Size

   Setting:

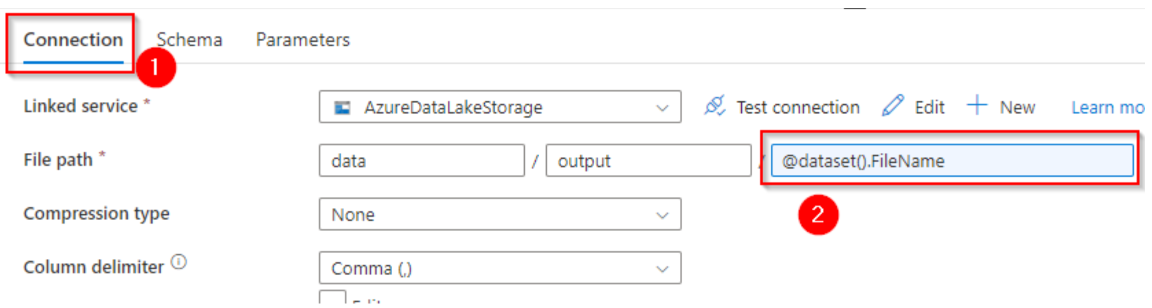
   a. **Dataset:**

      - **Click + New**
      - **Type:** Azure Data Lake Storage Gen 2
      - **Format**: DelimitedText
      - **Name**: FileMetadataInfo
      - **Linked** Service: AzureDataLakeStorage
      - **File path:** data / output /
      - **First row as header**: Unchecked
      - **Import schema**: None
      - **Advance:** Open this dataset
        - In Parameters tab create a new parameter
          Name: FileName



Name	Type	Default value
FileName	String	Value

        - In connection tab for File path give the File Name: **@dataset().FileName**



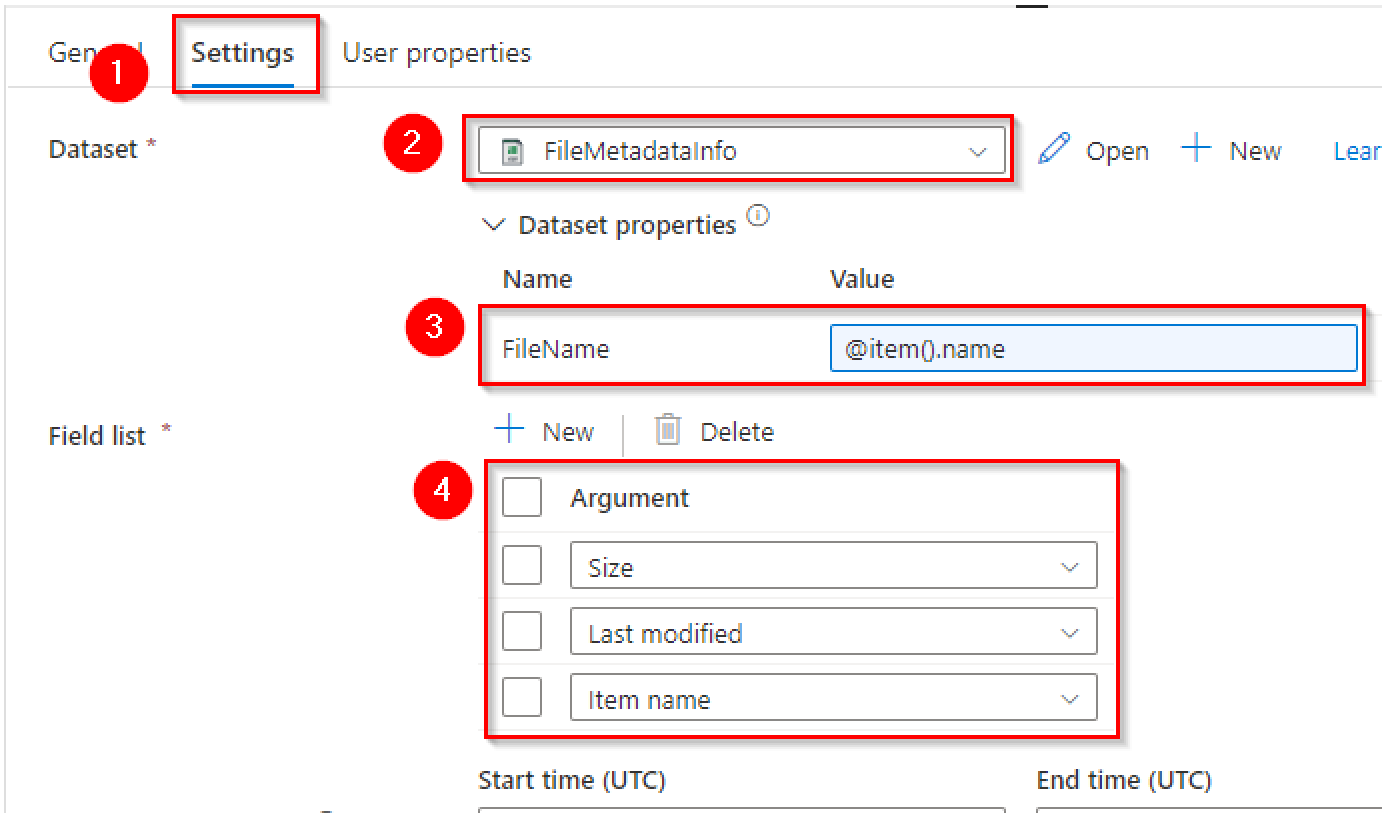
Linked service *: AzureDataLakeStorage
File path *: data / output / @dataset().FileName
Compression type: None
Column delimiter: Comma (,)

b. **Dataset Properties**

   FileName: @item().name

c. **Field list**:

   - Click + New
   - From the drop down select:
     - **Size**
     - **Last Modified**
     - **Item name**



Dataset * 2 FileMetadataInfo

Name	Value
FileName	@item().name

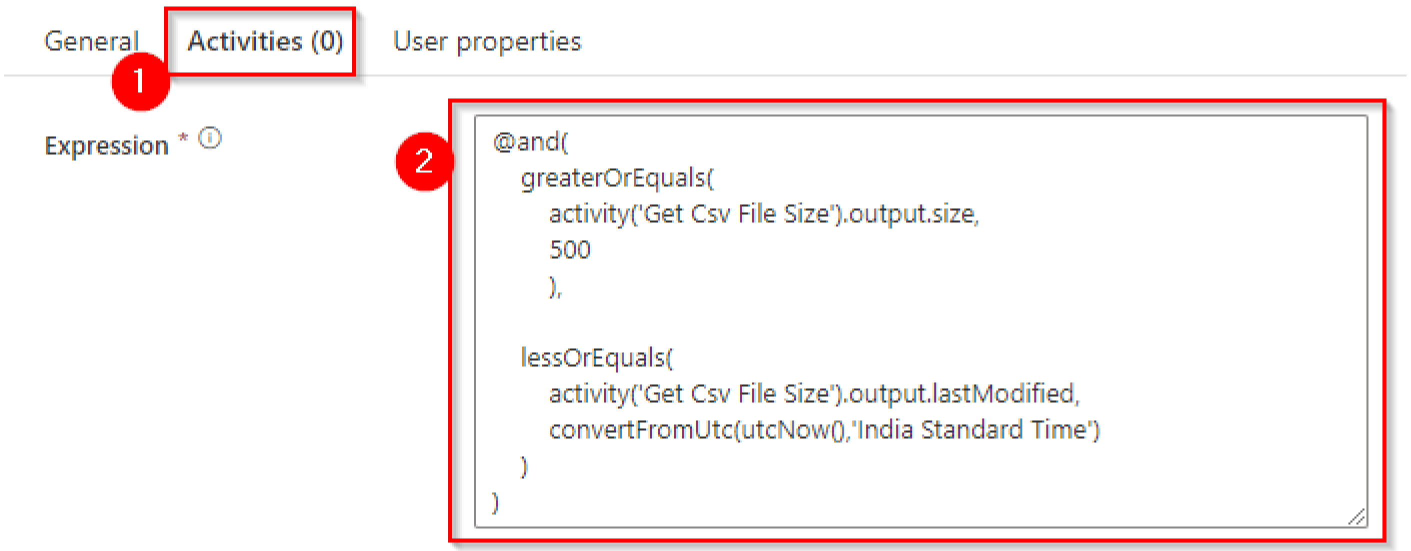
Field list *

Argument
Size
Last modified
Item name

5. Add **If condition** from **Iteration & Conditionals** section **inside** the **ForEach activity** in pipeline and enter the following expression under activities tab:

```
@and(
    greaterOrEquals(
        activity('Get Csv File Size').output.size,
        500
        ),

    lessOrEquals(
        activity('Get Csv File Size').output.lastModified,
        convertFromUtc(utcNow(),'India Standard Time')
    )
)
```

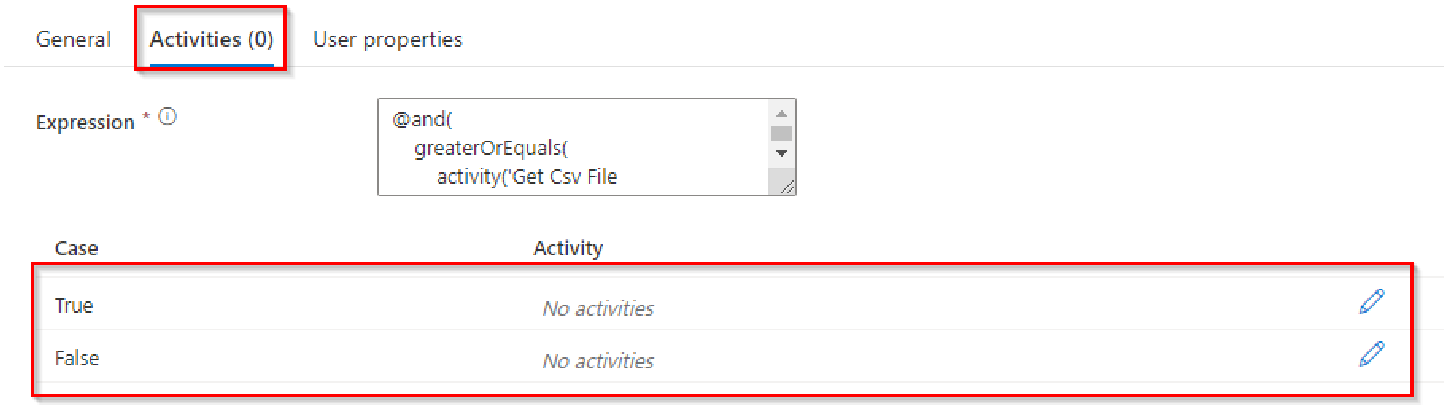


Expression *

```
@and(
    greaterOrEquals(
        activity('Get Csv File Size').output.size,
        500
        ),

    lessOrEquals(
        activity('Get Csv File Size').output.lastModified,
        convertFromUtc(utcNow(),'India Standard Time')
    )
)
```

6. Click the pencil icon **for True case**



Expression *

```
@and(
    greaterOrEquals(
        activity('Get Csv File
```

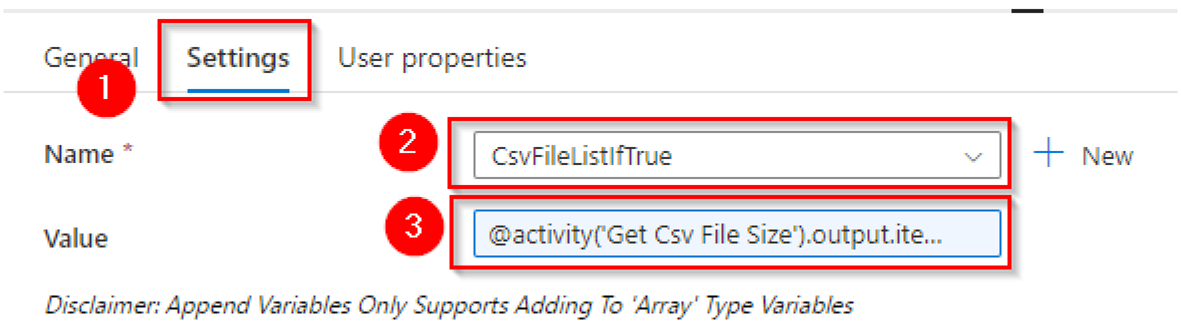
Case	Activity
True	*No activities*
False	*No activities*

Add **Append Variable** activity

Under **setting** tab of it provide the following details

Name : From Drop down list select **CsvFileListIfTrue**

Value: **@activity('Get Csv File Size').output.itemName**



Name * CsvFileListIfTrue

Value @activity('Get Csv File Size').output.ite...

Disclaimer: Append Variables Only Supports Adding To 'Array' Type Variables

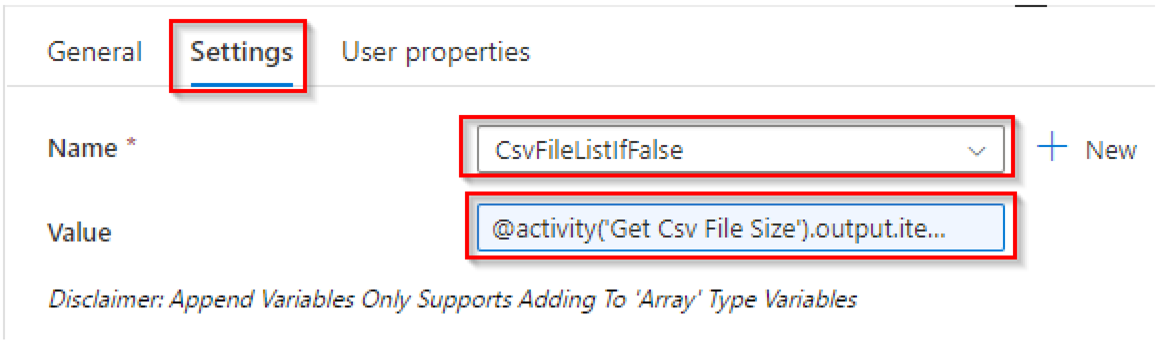
7. Go back to **ForEach** activity, Click the pencil icon **for False case** in **If Condition** Activity.

   Add **Append Variable** activity

   Under **setting** tab of it provide the following details

      Name : From Drop down list select **CsvFileListIfFalse**

      Value: **@activity('Get Csv File Size').output.itemName**

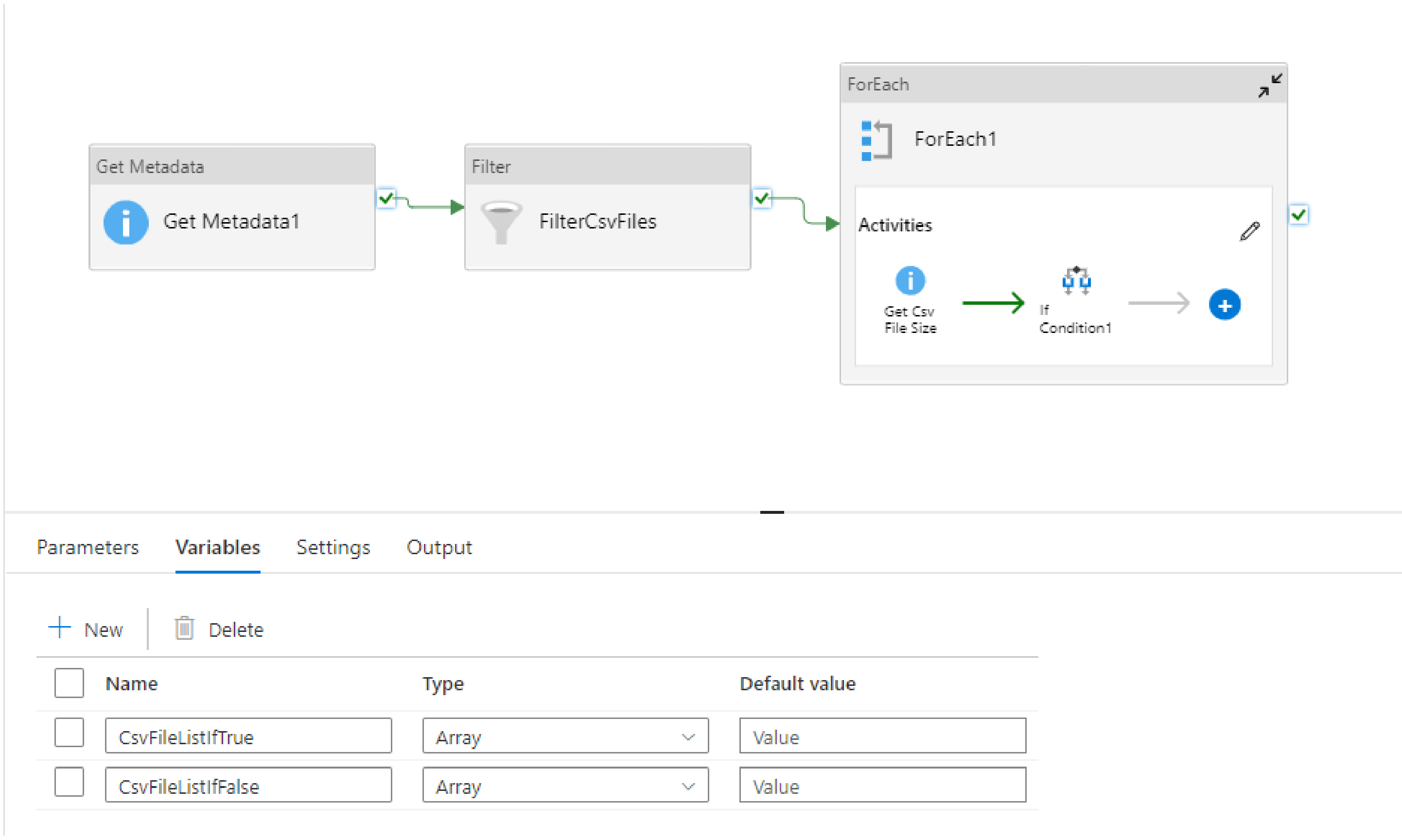


Name * : CsvFileListIfFalse

Value : @activity('Get Csv File Size').output.ite...

Disclaimer: Append Variables Only Supports Adding To 'Array' Type Variables

8. Publish the pipeline. Your pipeline look like as:



Name	Type	Default value
CsvFileListIfTrue	Array	Value
CsvFileListIfFalse	Array	Value

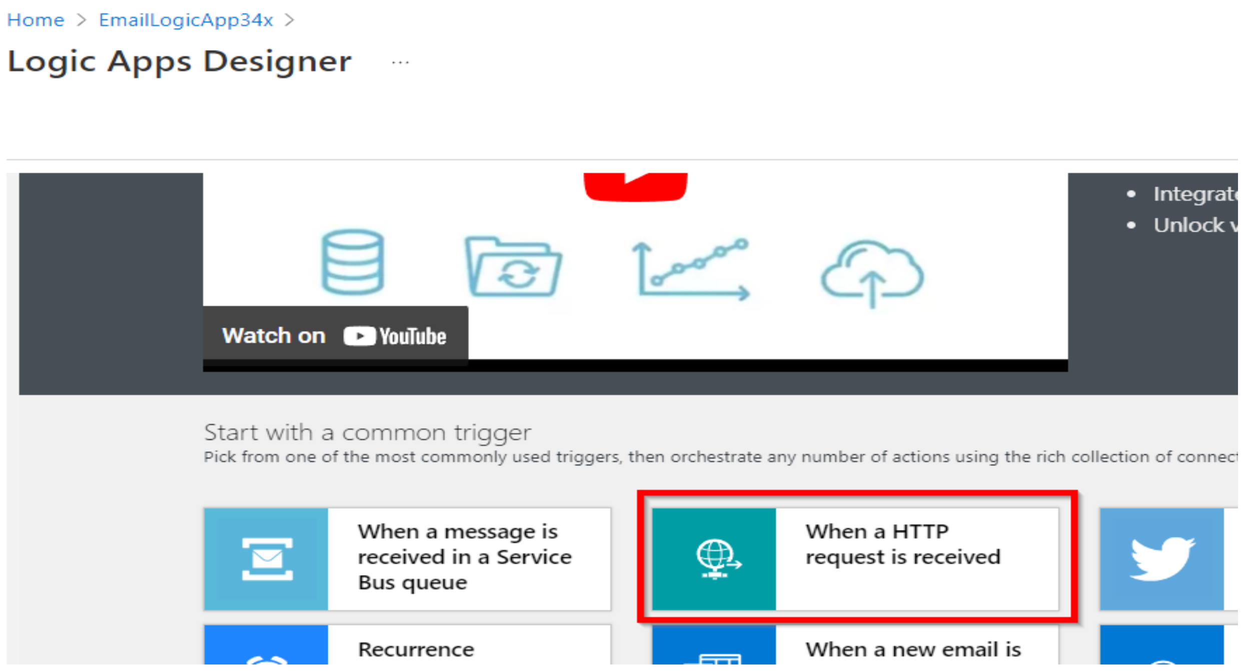
Exercise 2: Create a Logic App to send custom email form Azure Data Factory Pipeline.

The task for this exercise are as follows:

1. Create and Configure the Logic App to send the email.
2. Configure the pipeline to call the Logic App.

Task 1: Create Logic App

1. Search for **Logic Apps** in search bar at the top, click on Create and provide the following details.
   a. **Subscription** –> Azure Pass - Sponsorship
   b. **Resource group Name** -> adfxx-rg
   c. **Logic App name** –> EmailLogicApp-xx
   d. **Region** –> Same as your Resource Group
   e. **Enable log analytics** -> No
   f. **Plan Type** –> Consumption
2. Leave the other settings to Default and click **Review + Create** and then **Create**.
3. Once deployed click on **Go To Resource** and click on **When a HTTP request is received**. Or inside the logic app select **Logic App Designer** from the left-hand side menu under development Tools section.


Home > EmailLogicApp34x >
Logic Apps Designer
Watch on YouTube
Integrat
Unlock v
When a message is received in a Service Bus queue
When a HTTP request is received
Recurrence
When a new email is

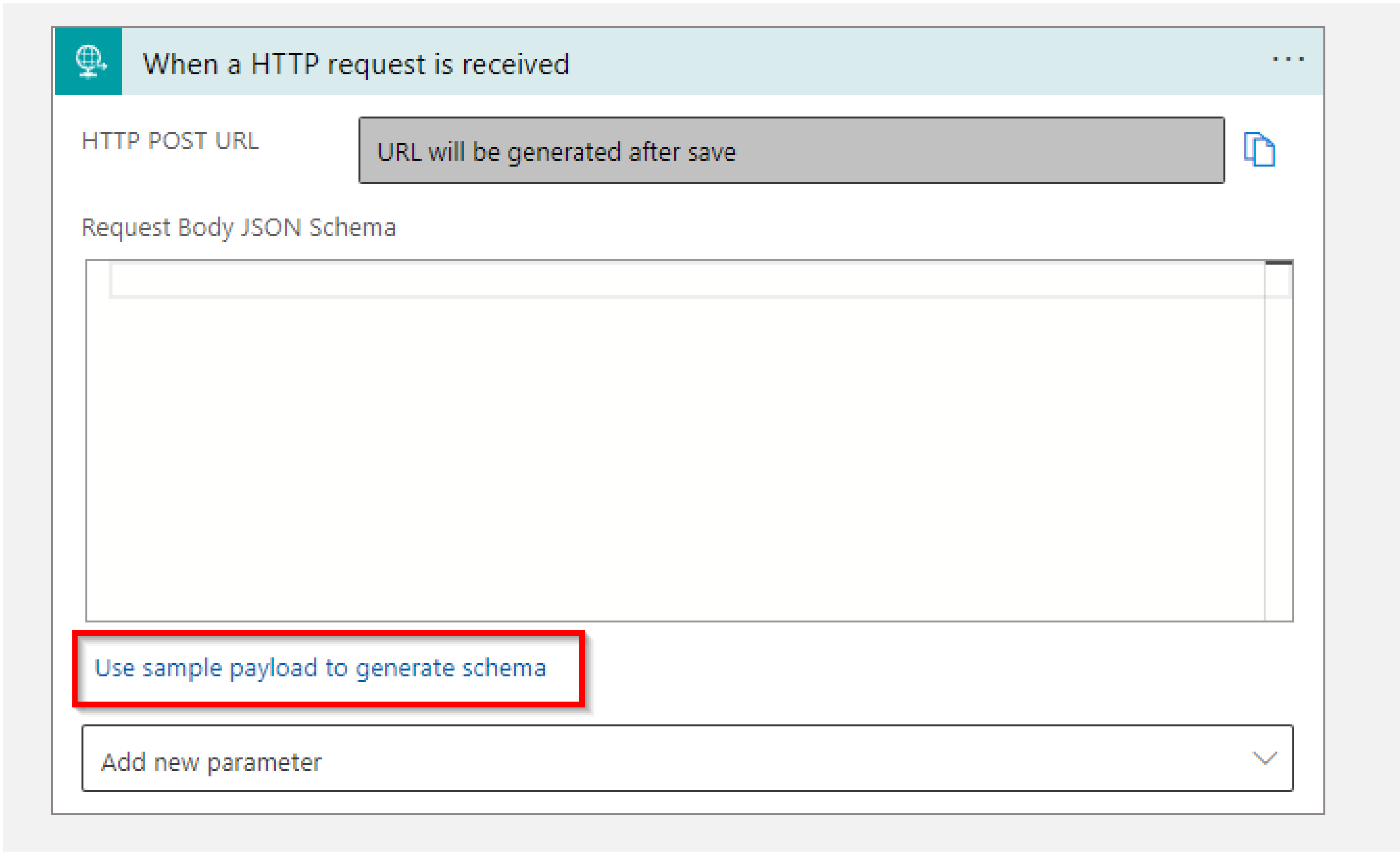
Start with a common trigger

Pick from one of the most commonly used triggers, then orchestrate any number of actions using the rich collection of connect

4. **If coming from Logic App Designer** then under Choose and Operation, Search for **Request**, Select the **Request** and then Select **When a HTTP Request is received**.

5. Click Use Sample payload to generate schema and paste the following code and click on Done.

```
{
    "InvalidFileList" : [],
    "ValidFileList" :[],
    "PipelineRunId" : [],
    "PipelineName" : "",
    "ADFName" : ""
}
```

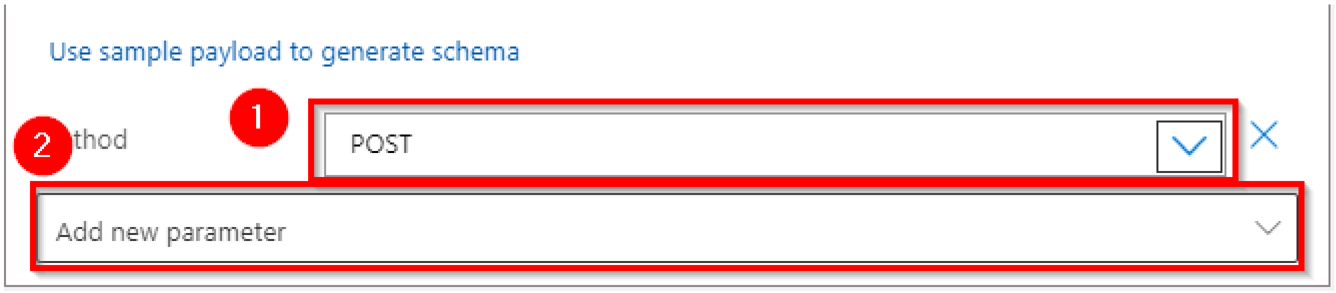


When a HTTP request is received

HTTP POST URL: URL will be generated after save

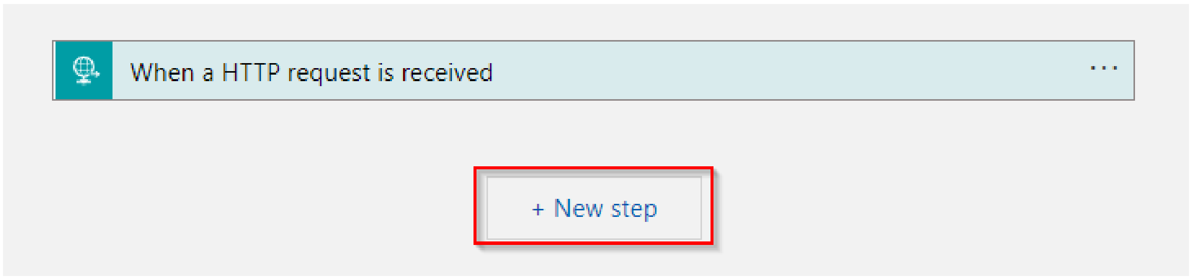
Request Body JSON Schema

6. Click on Add new parameter, For Method and Select POST.

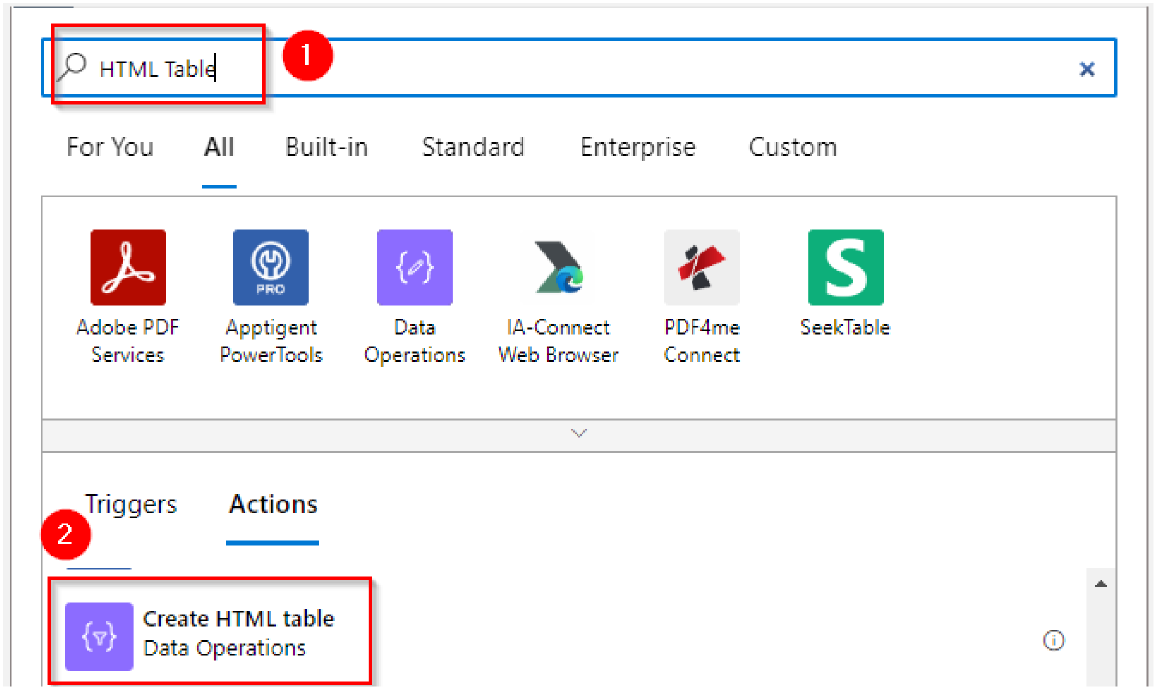


Method: POST

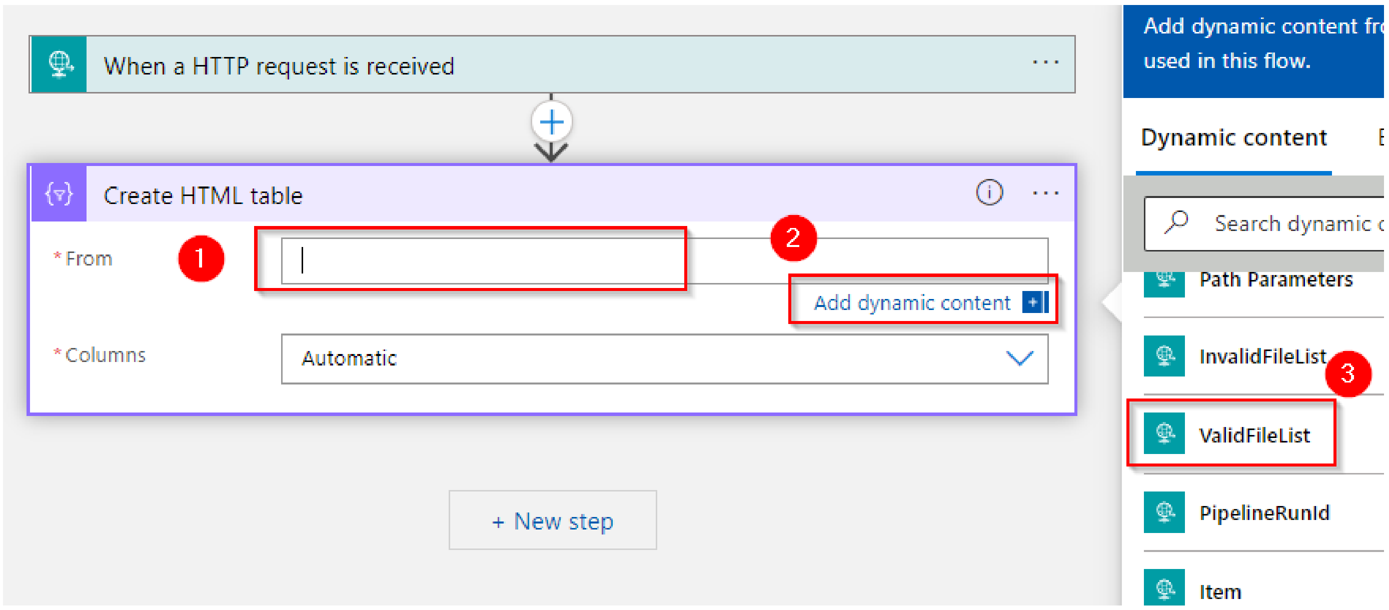
7. Click on the Plus icon to insert a new Step and Click on Add an Action.



8. Under choose an operation, search for HTML Table. Select the Create HTML Table and rename it to **Create HTML Table Valid File**

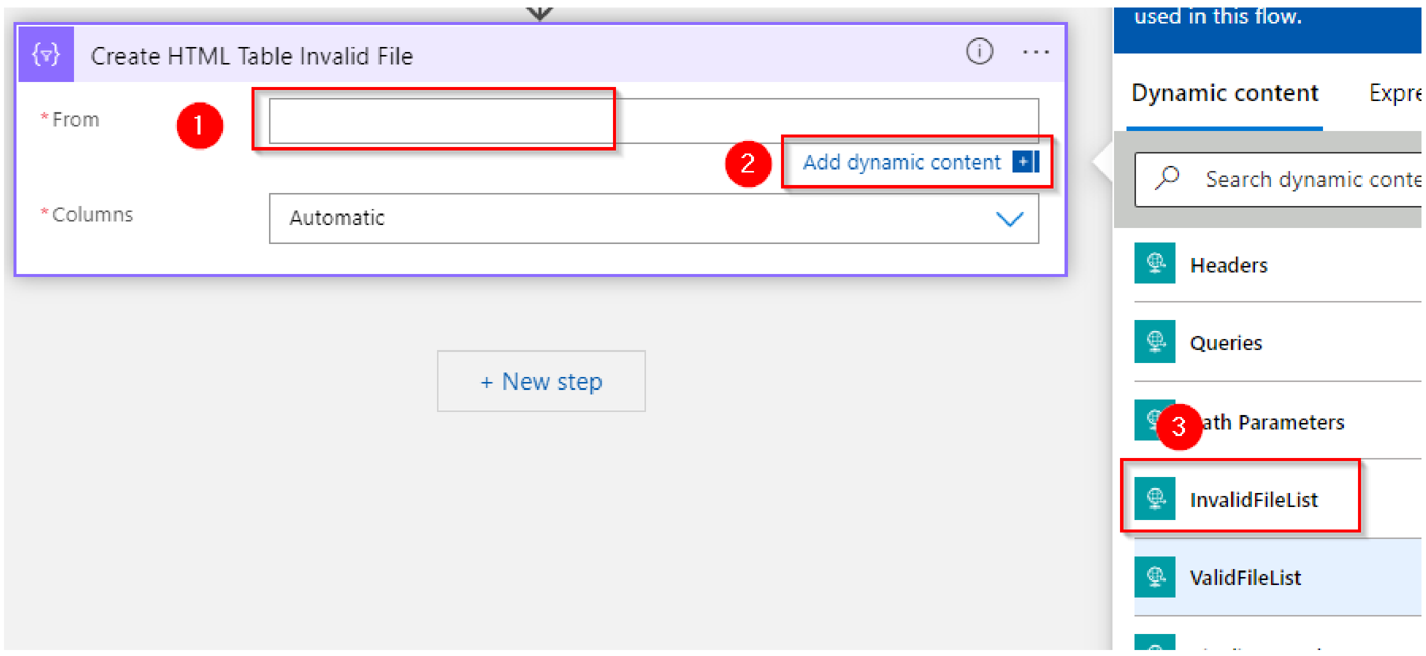


9. In Create HTML Table Valid File, do as show below.



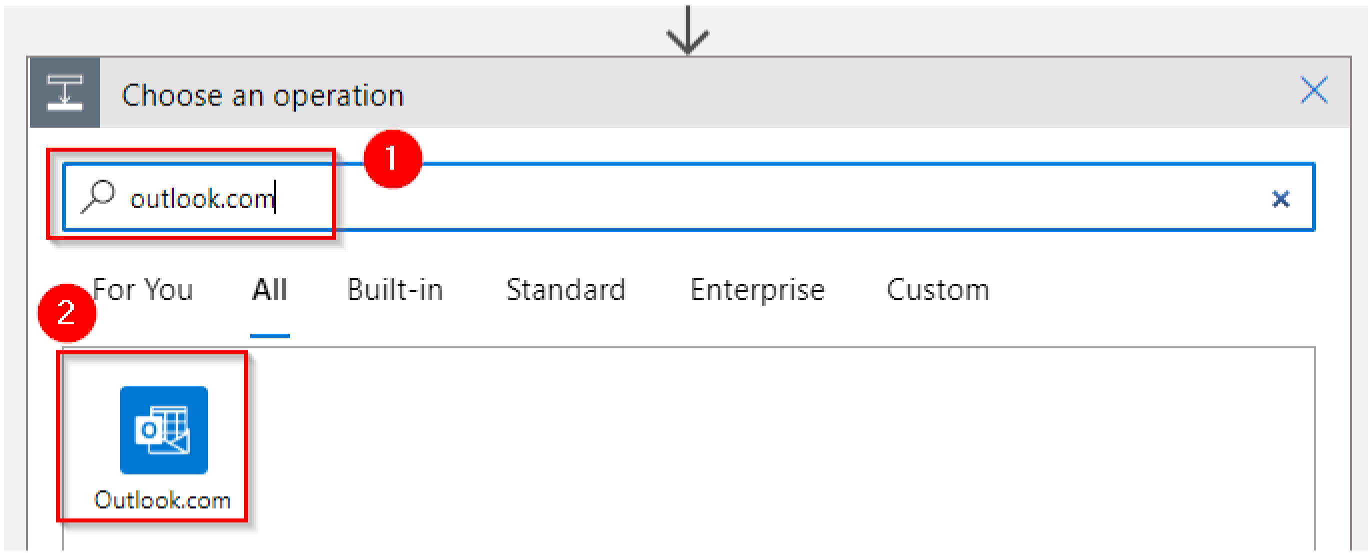
10. Click on the Plus icon to insert a new Step.

11. Under choose an operation, search for HTML Table. Select the Create HTML Table and rename it to **Create HTML Table Invalid File**

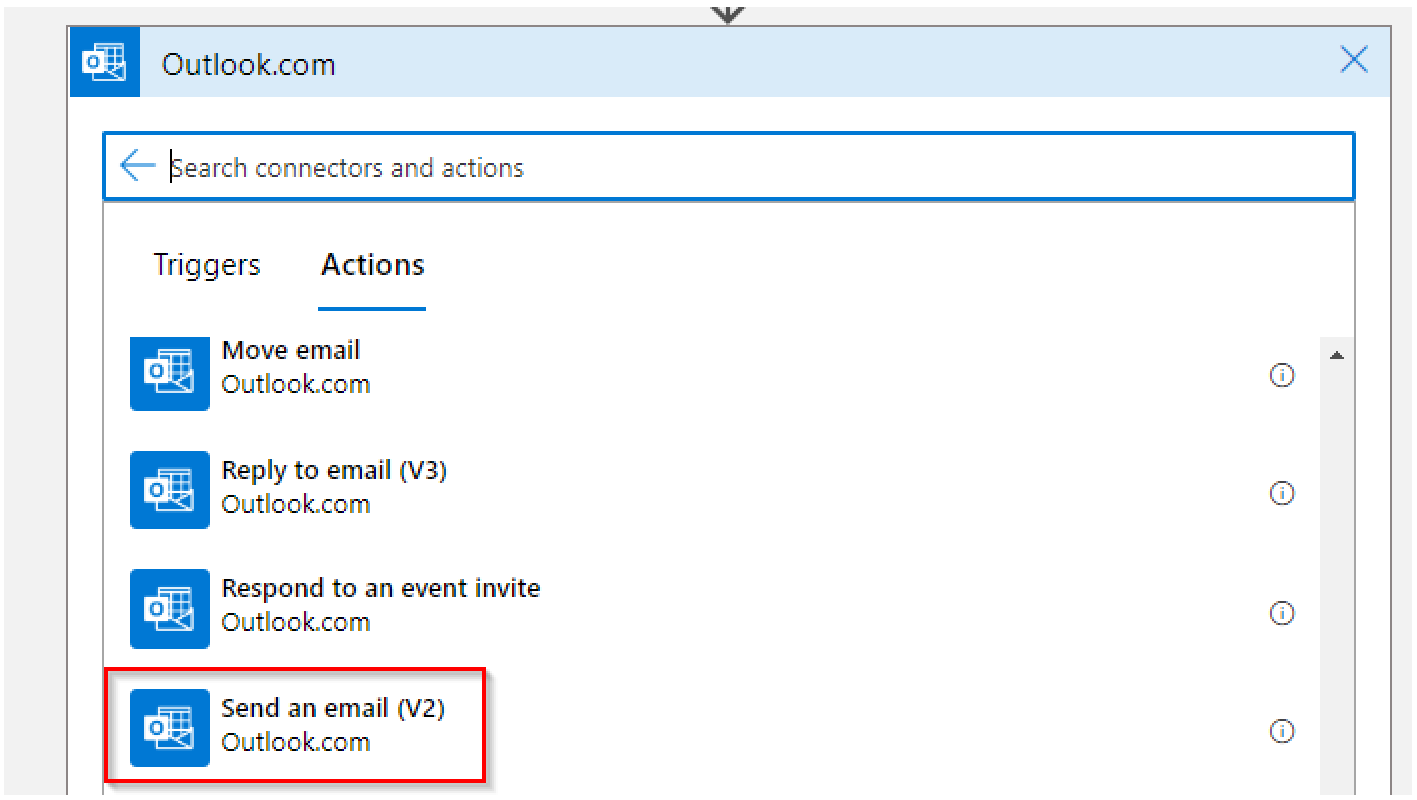


12. Click on the Plus icon to insert a new Step.

13. Under Choose an operation, Search for Outlook.com and select it.



14. Under Actions, Select **Send an email (V2)** and Sign in using an Outlook Account. (Create one if you don't have)



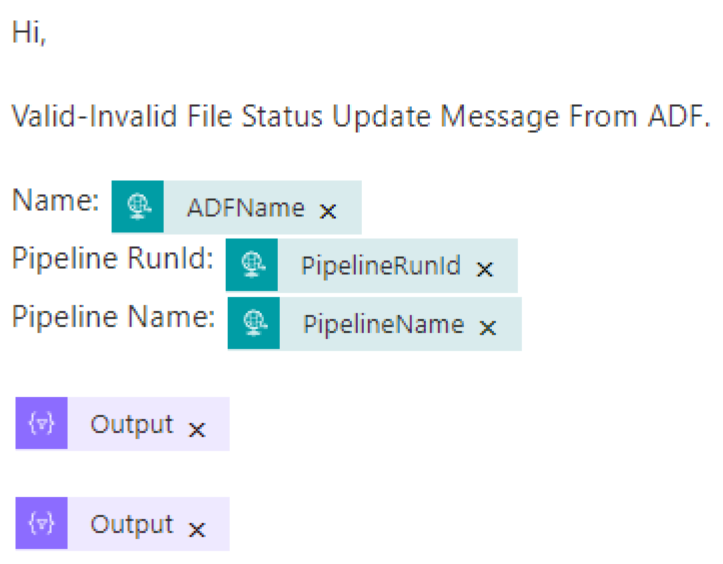
15. After Signing In, Click on Save.

16. Enter the following Details in Parameters.

To: Your Outlook Mail ID

Subject: Valid-Invalid Files – ADF Status

Body:



Hi,

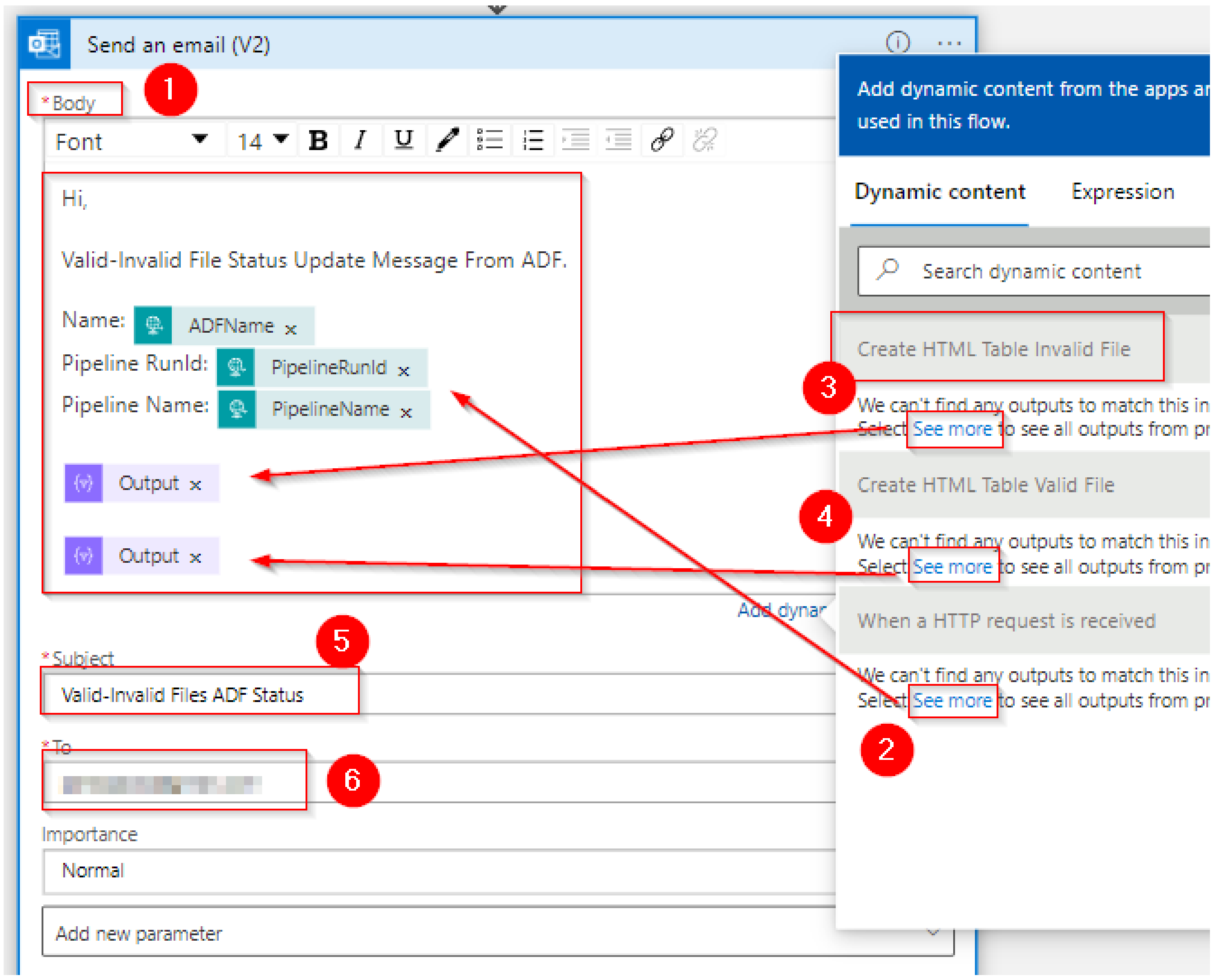
Valid-Invalid File Status Update Message From ADF.

Name: ADFName
Pipeline RunId: PipelineRunId
Pipeline Name: PipelineName

Output

Output

Type the message body, drag and drop the relevant field as show above from the dynamic content section to its proper place (see below image).


Send an email (V2)
Body
1
Font
14
Hi,
Valid-Invalid File Status Update Message From ADF.
Name: ADFName
Pipeline RunId: PipelineRunId
Pipeline Name: PipelineName
Output
Output
Add dynar
5
Subject
Valid-Invalid Files ADF Status
To
6
Importance
Normal
Add new parameter
Add dynamic content from the apps ar used in this flow.
Dynamic content
Expression
Search dynamic content
Create HTML Table Invalid File
3
We can't find any outputs to match this in Select See more to see all outputs from pr
Create HTML Table Valid File
4
We can't find any outputs to match this in Select See more to see all outputs from pr
When a HTTP request is received
We can't find any outputs to match this in Select See more to see all outputs from pr
2

17. Click on Save.

18. Go back to the first step, **When a HTTP request is received**. Open it and can copy the **HTTP Post URL** and paste it in a notepad.

Task 2: Configure the pipeline to send the email.

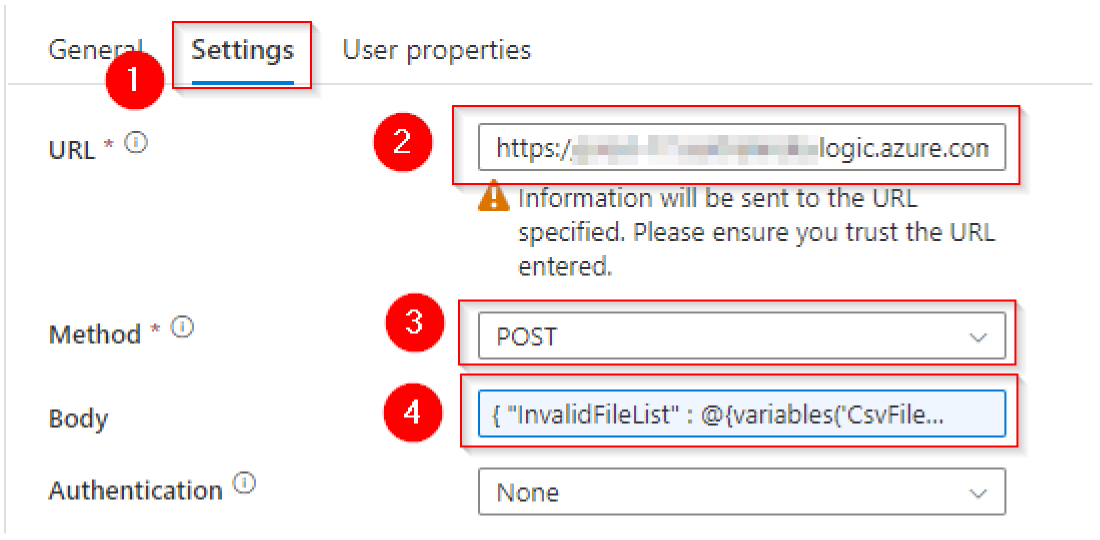
1. Go to the **Azure Data Factory** workspace and open the pipeline if not open already.
2. Add **Web Activity** to the pipeline from Activities canvas under General section. Connect **ForEach** activity to **web activity** on success.
3. Under **setting** tab of **Web Activity** provide the following details.

   URL: past the **HTTP Post URL** copied from Logic App.

   Method: **Post**

   Body:

```
{
    "InvalidFileList" : @{variables('CsvFileListIfFalse')},
    "ValidFileList" : @{variables('CsvFileListIfTrue')},
    "PipelineRunId" : "@{pipeline().RunId}",
    "PipelineName" : "@{pipeline().Pipeline}",
    "ADFName" : "@{pipeline().DataFactory}"
}
```



URL * https:/[redacted]logic.azure.con

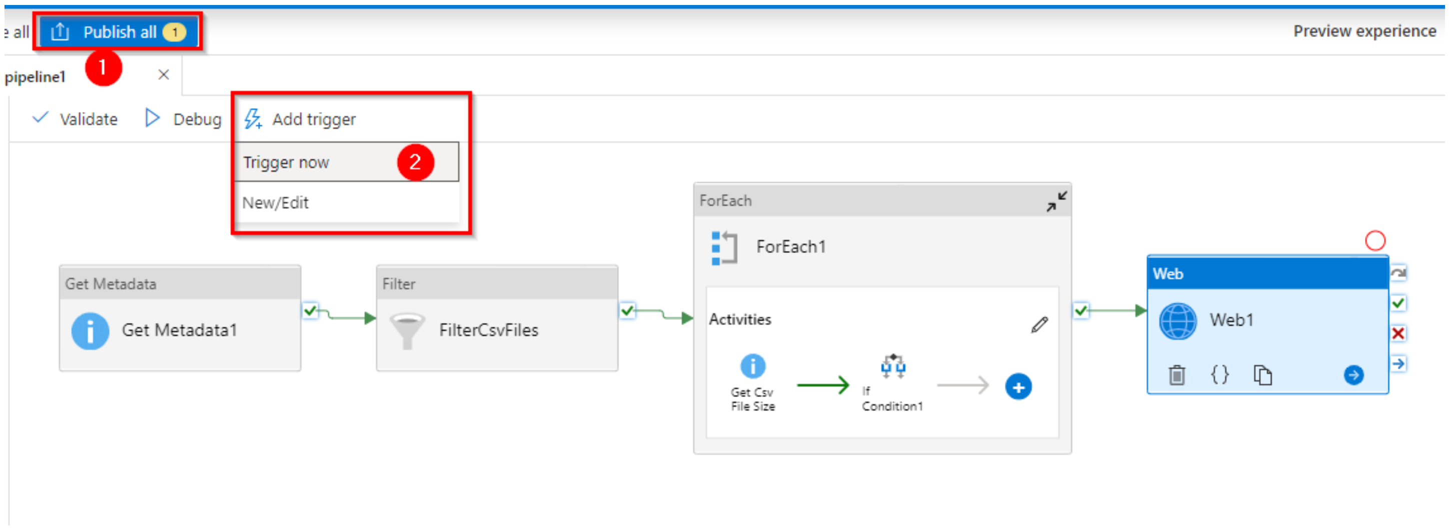
Information will be sent to the URL specified. Please ensure you trust the URL entered.

Method * POST

Body { "InvalidFileList" : @{variables('CsvFile...

Authentication None

4. Publish the pipeline and Run it.



5. When run successfully check the inbox of the provided mail.