

Lab 4. Flow control, variables, expressions

Author: Serge Luca aka "Doctor Flow"

Updated: 6/14/2020, based on the feedback received from Rishona Elijah.

Updated: 9/15/2020 : more complex expression in the Select action. And modification of the generated html code

Learning objectives: Flow control, expressions, variables, using Date/Time

Duration: 50 minutes.

Scenario: We have a list of offices in an Excel sheet. Create a Flow that will send a report describing this list of offices, including the biggest office.

Task 4.1: Create an Excel workbook and a scheduled flow

1. Create an Excel workbook to use in this lab.
 - a. In your One Drive (for Business), create an Excel workbook named **Offices.xlsx**.
 - b. Add two columns similar to the screenshot below, with the cities and capacities data, and then format the data as a table with headers:

Contoso Offices	
city	Capacity
London	100
Brussels	250
Seattle	80
Vancouver	200
Toronto	400
Antwerpen	15
Warsaw	300
Paris	54
Berlin	70
Amsterdam	60
Montreal	78

Note: This document contains the list of offices of Contoso Corp. Each office has a limited number of seats.

Every month a report describing the list of offices and the total number of seats is sent to the management (in this case the management is...yourself). The e-mail should look like this:

Office Capacity Report



Today, 3:10 PM

This message was sent with low importance.

The biggest office is : Toronto

Its capacity is : 400

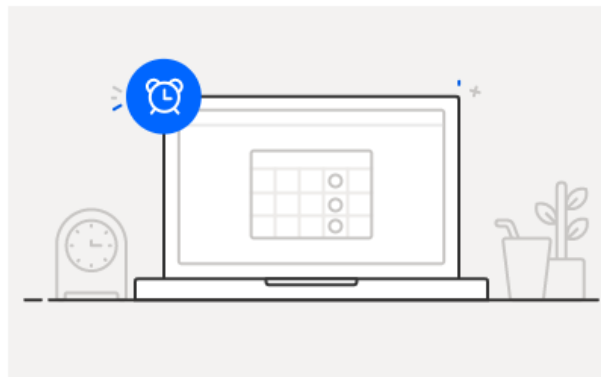
The total capacity is :

List of offices

City	Capacity
London	100
Brussels	250
Seattle	80
Vancouver	200
Toronto	400
Antwerpen	15
Warsaw	300
Paris	54
Berlin	70
Amsterdam	60
Montreal	78

2. Create a flow to generate this email report.
 - a. Navigate to the [Microsoft Flow portal](#).
 - b. **New > Scheduled –from blank**
 - c. Use the following screenshot to name the flow, and set the flow frequency.

Build a scheduled flow



Flow name

Office Capacity

Run this flow *

Starting 5/13/19 at 10:00 AM

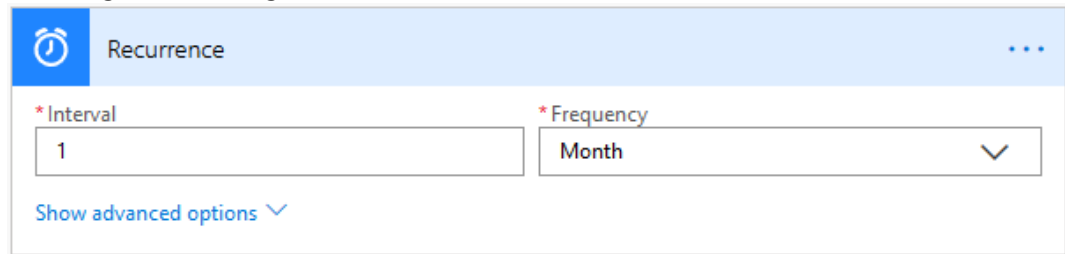
Repeat every 1 Month

This flow will run:

Every month

- d. Click **Create**.

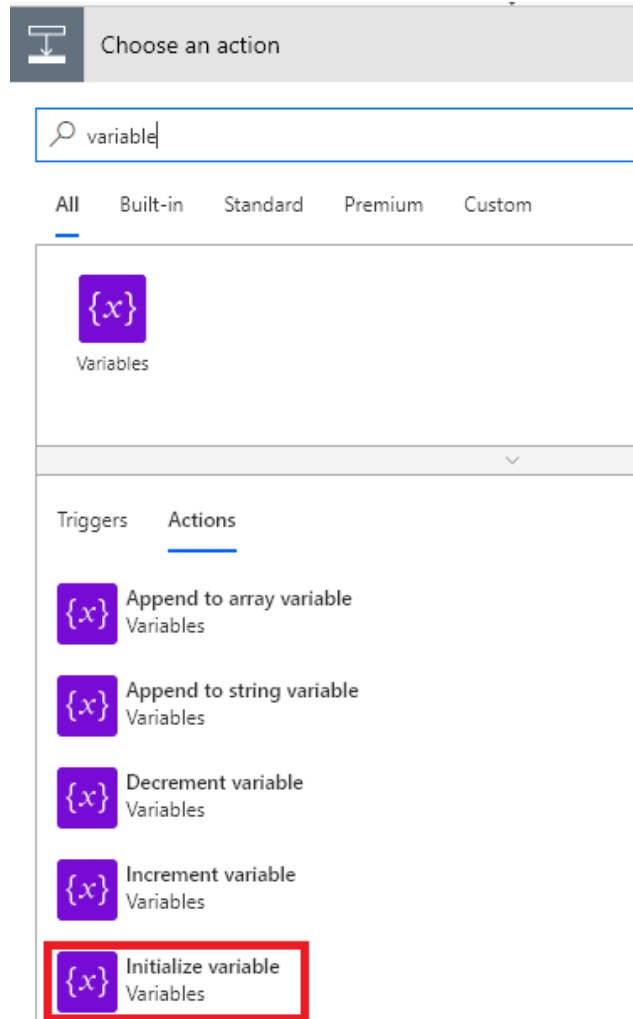
The following flow will be generated:



The image shows a 'Recurrence' configuration panel. It has a blue header with a clock icon and the word 'Recurrence'. Below the header, there are two input fields: '* Interval' with the value '1' and '* Frequency' with a dropdown menu showing 'Month'. A 'Show advanced options' link with a downward arrow is located below these fields.

3. The first challenge will be to define the **Total Capacity**. Use the following steps to create a variable that will contain that value.

- a. Add an action below the trigger and select **Initialize variable**:



The image shows the 'Choose an action' interface. At the top is a search bar with the text 'variable'. Below the search bar are tabs for 'All', 'Built-in', 'Standard', 'Premium', and 'Custom'. The 'All' tab is selected. Below the tabs is a list of actions, each with a purple icon containing '{x}' and a text label. The actions are: 'Variables', 'Append to array variable', 'Append to string variable', 'Decrement variable', 'Increment variable', and 'Initialize variable'. The 'Initialize variable' action is highlighted with a red rectangular box.

- b. Rename this action to **Initialize variable Total Capacity**, set the variable name **Total Capacity**, and select **Integer** as the type with an initial **Value** of **0**:

{x}

Initialize variable Total Capacity

i

...

* Name	Total Capacity
* Type	Integer
Value	0

Add dynamic content

+

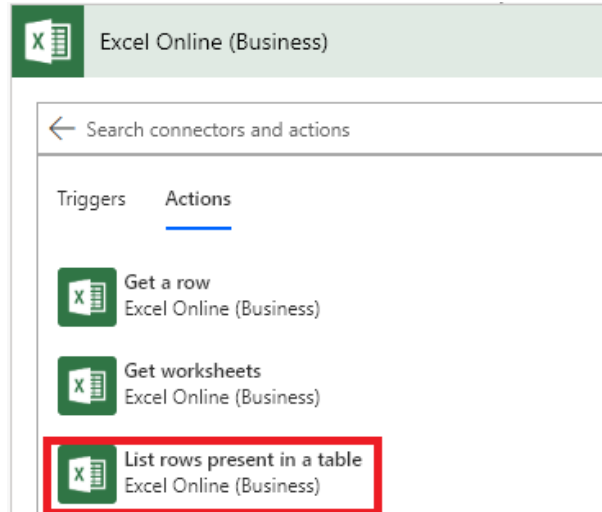
4. Name the flow **Office Capacity** and **Save**.

Task 4.2: Extend the flow to loop through all offices

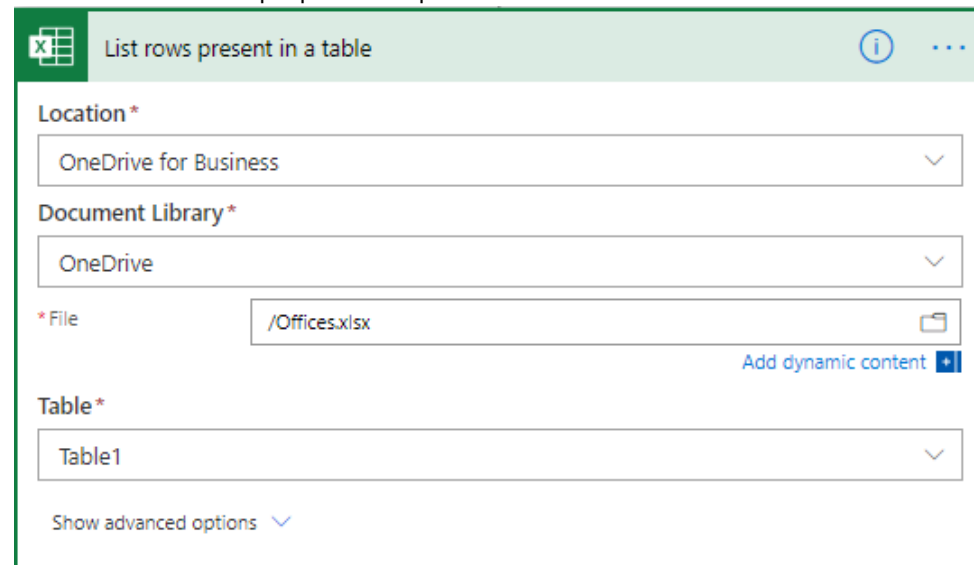
In this task you will update the flow; make it loop through all offices, retrieve their capacity, and increment the Global Capacity variable to calculate the total capacity.

1. To retrieve the list of offices.

- a. **Add an action > List rows present in a table:**

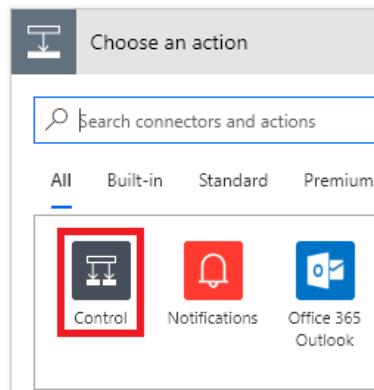


- b. Set the Excel action's properties as per the next screenshot:

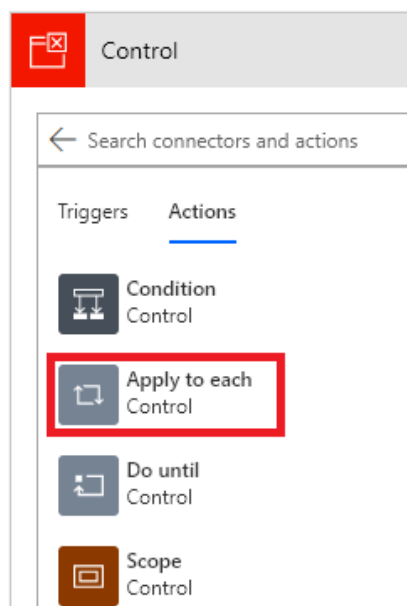


2. Loop through the cities

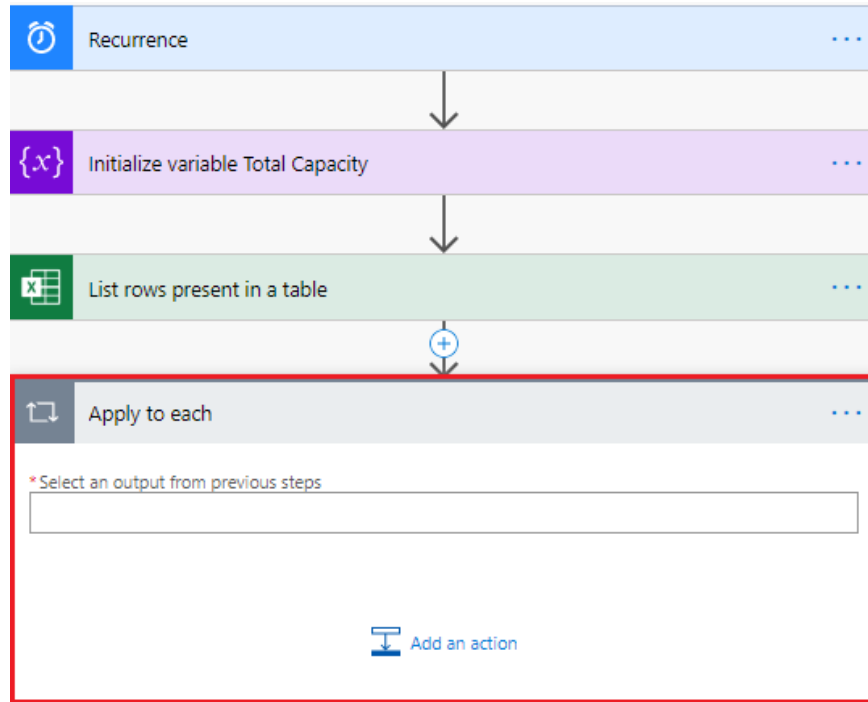
- a. **Add an action > Control**



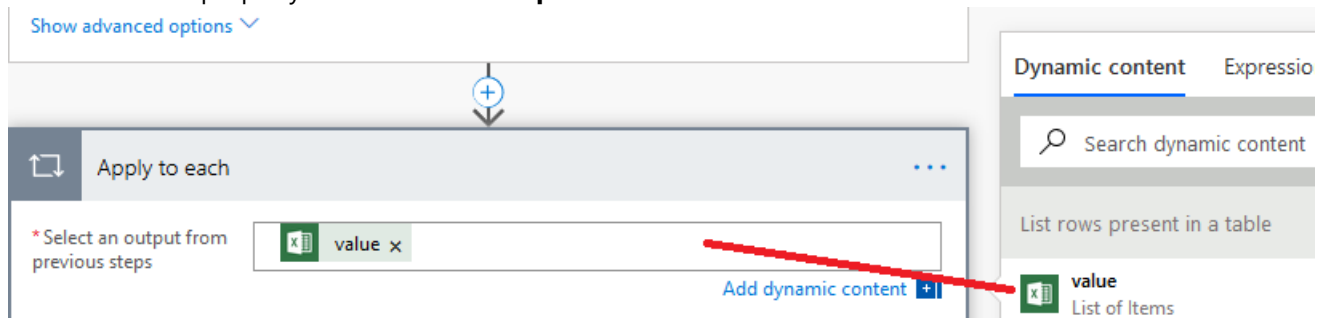
b. Click **Apply to each:**



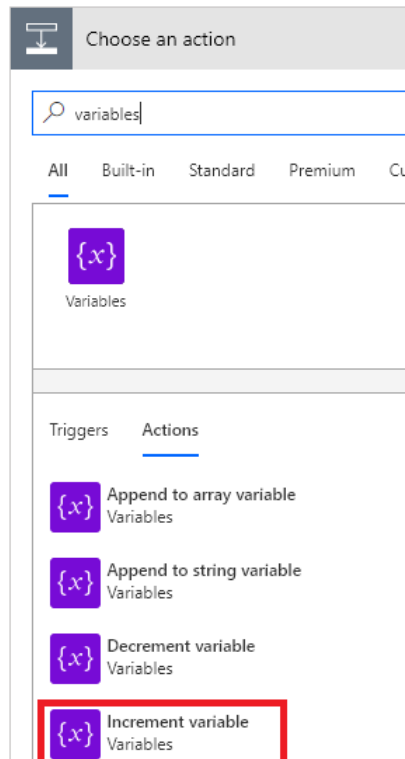
Your Flow should now look like this:



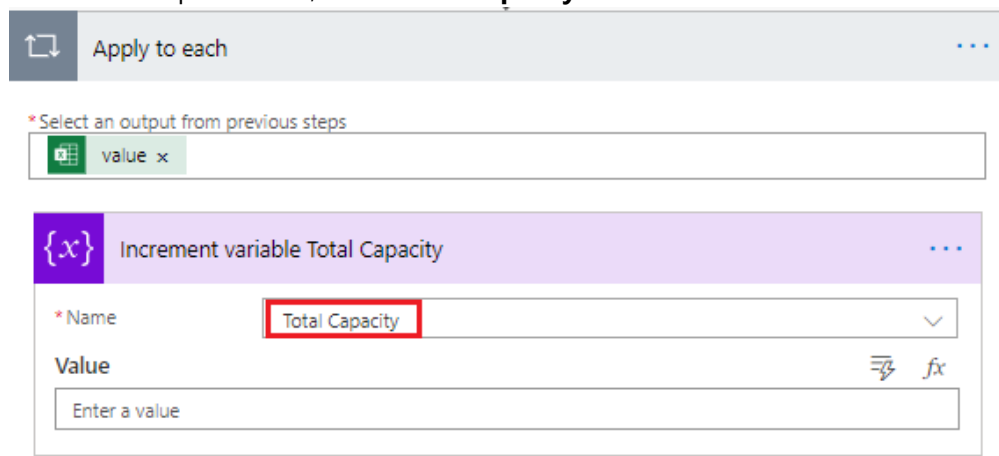
3. Configure the **Apply to each** action (it expects a list of values), using the Dynamic content tab to select the **value** property from the **List rows present in a table** action.



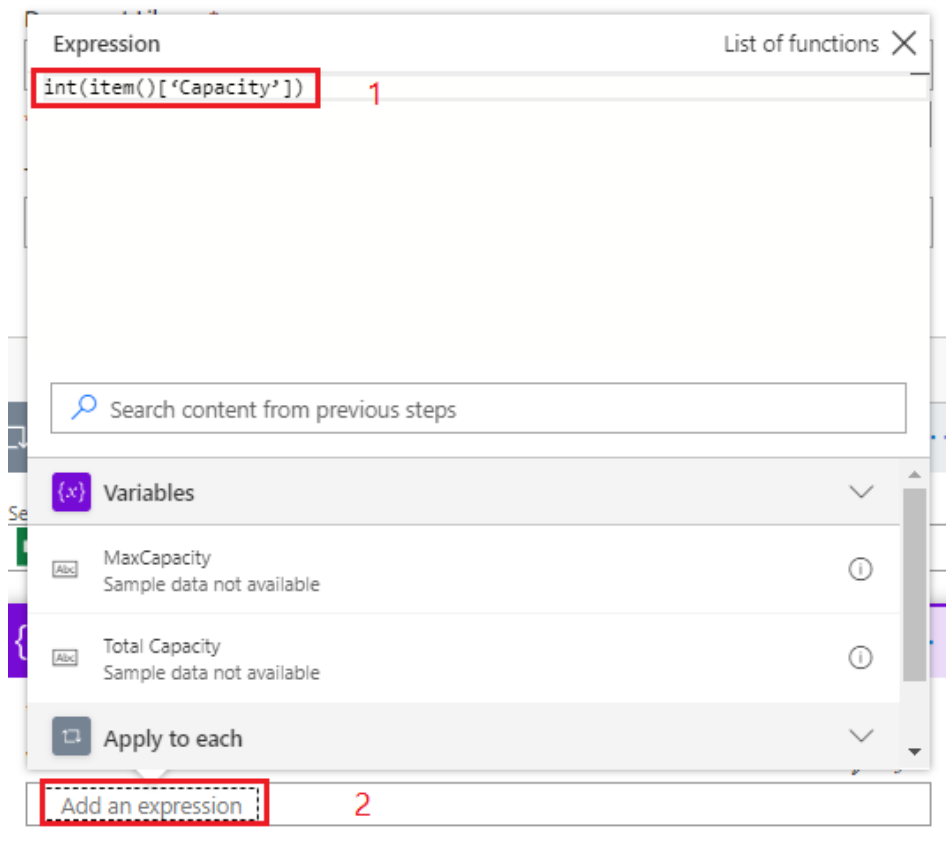
4. Calculate the current office capacity using a variable and an expression.
 - a. In the **Apply to each** action, click **Add an action** > **Increment variable**:



- b. In the **Name** drop-down list, select **Total Capacity**.



- c. Click inside the **Value** text box, and then using the **Expression** tab in the **fx** textbox, type `int(item()['Capacity'])` and click **Add an expression** :



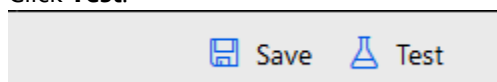
Note:

The **item()** expression retrieves the current record information in the current loop and **['Capacity']** provides the field name to retrieve. **Item()['Capacity']** returns a string. To transform a string to an integer (because we need to increment it), we use the **int()** function.

There are many other expressions available in flow, and we encourage you to read the flow documentation related to expressions after doing the labs. You can start from the following web page, <https://Flow.microsoft.com/en-us/blog/use-expressions-in-actions/>.

5. To test the flow, without waiting one month before the flow starts, use the **Test** button to manually start the flow on demand (in test mode). This is convenient for testing and debugging purposes.

- a. Click **Test**.



- b. Select **I'll perform the trigger action:**

Test Flow

- ☒ I'll perform the trigger action
- ☐ Using data from previous runs
Choose data from a list of previous runs:

Test Cancel

c. Click on **Run Flow**:

Run flow




Office Capacity
Owners: serge Luca


See details


This flow doesn't need additional input to run


Run flow Cancel


d. Wait until you get the message: **Your Flow ran successfully.**


 Your flow ran successfully.




 Recurrence

 Initialize variable Total Capacity

 List rows present in a table

 Apply to each

< Previous < Previous failed Show 1 of 11 Next failed > Ne

 Increment variable

e. To check the **Total Capacity** value, you can examine the value of **Total Capacity** for each step. For example, in our case, we will check its value once it has completed the loop 11 times: so, type **11** In the **Show** textbox:

↺

Apply to each

4s

↩ Previous

↩ Previous failed

Show

11

of 11

Next failed

Next

{x}

Increment variable

0s

- f. Click **Increment variable** to display a value of **1607** (if you use the values in the Excel workbook as defined at the beginning of the lab).

↺

Apply to each

4s

↩ Previous

↩ Previous failed

Show

11

of 11

Next failed

Next

{x}

Increment variable

0s

INPUTS

Name

Total Capacity

Increment By

78

OUTPUTS

Name

Total Capacity

Value

1607

6. Define 2 new variables

- a. Below the variable, **Total capacity** and before the loop, add two new variables named:
- **Bigger Office** (type string)
 - **MaxCapacity** (type integer)

The screenshot shows a flowchart with three initialization steps:

- Initialize variable Total Capacity** (purple box)
- Initialize variable Bigger office** (purple box) - A red arrow points to this box.
- Initialize variable MaxCapacity** (purple box) - A red arrow points to this box.

Each box has a configuration panel below it:

- Bigger office:** Name: Bigger office, Type: String, Value: Enter initial value.
- MaxCapacity:** Name: MaxCapacity, Type: Integer, Value: Enter initial value.

Arrows indicate the flow from the first box to the second, and then to the third.

7. **Add a Condition** (from the Control connector) in the **Apply to each** action:

8. The goal is to compare 2 numbers and to select the larger one. In order to do so, we need to transform our capacity values into integers. In the left side of the condition, click **Choose a value** and click on **Expression**. As we already did it before, type **int(item()['Capacity'])** as illustrated below:

The screenshot shows the configuration panel for a **Condition** action. The **Expression** tab is selected, and the expression `int(item()['Capacity'])` is entered. A red arrow points to the `int` function. Below the expression, there is a list of functions, including `concat(text_1, text_2?, ...)`.

9. Click **Ok**.

10. Select the comparison operator **is greater than**:

{x} Increment variable

Condition

f_x int(...) × is greater than Choose a value

+ Add

11. In the **Choose a value** textbox, we will include an expression much as we did before by using the **int()** expression, but with a small variation. Click choose a value, click on **Expression** and type **int()**

Condition

f_x int(...) × is greater than Choose a value

Edit in advanced mode Add dynamic content Collapse condition

Dynamic content Expression

f_x int() 2

OK

12. Move the cursor within the **int()** parentheses.

13. Click **Dynamic content**, select the **MaxCapacity** variable: the editor will automatically generate the expression. Click **Ok** and **Save** the Flow.

{x} Increment variable

Condition

f_x int(...) × is greater than Choose a value

Edit in advanced mode Add dynamic content Collapse condition

Dynamic content Expression

f_x int(variables('MaxCapacity'))

OK

Variables

{x} MaxCapacity

14. Now, in the left **If yes** branch, add a new action **Variables – Set variable** for our **MaxCapacity** variable.

15. Rename the action **Set variable MaxCapacity**:

✓

If yes

{x}

Set variable MaxCapacity

...

* Name

MaxCapacity

▼

Value*

Enter variable value

⚡ fx

16. and in the **Expression** panel type of this variable **int(item()['Capacity'])** as illustrated in the next picture.

↻ Apply to each

* Select a

Expression

List of functions X

int(item()['Capacity'])

1

Search content from previous steps

{x} Variables

▼

MaxCapacity

Sample data not available

ⓘ

Total Capacity

Sample data not available

ⓘ

Apply to each

▼

Current item

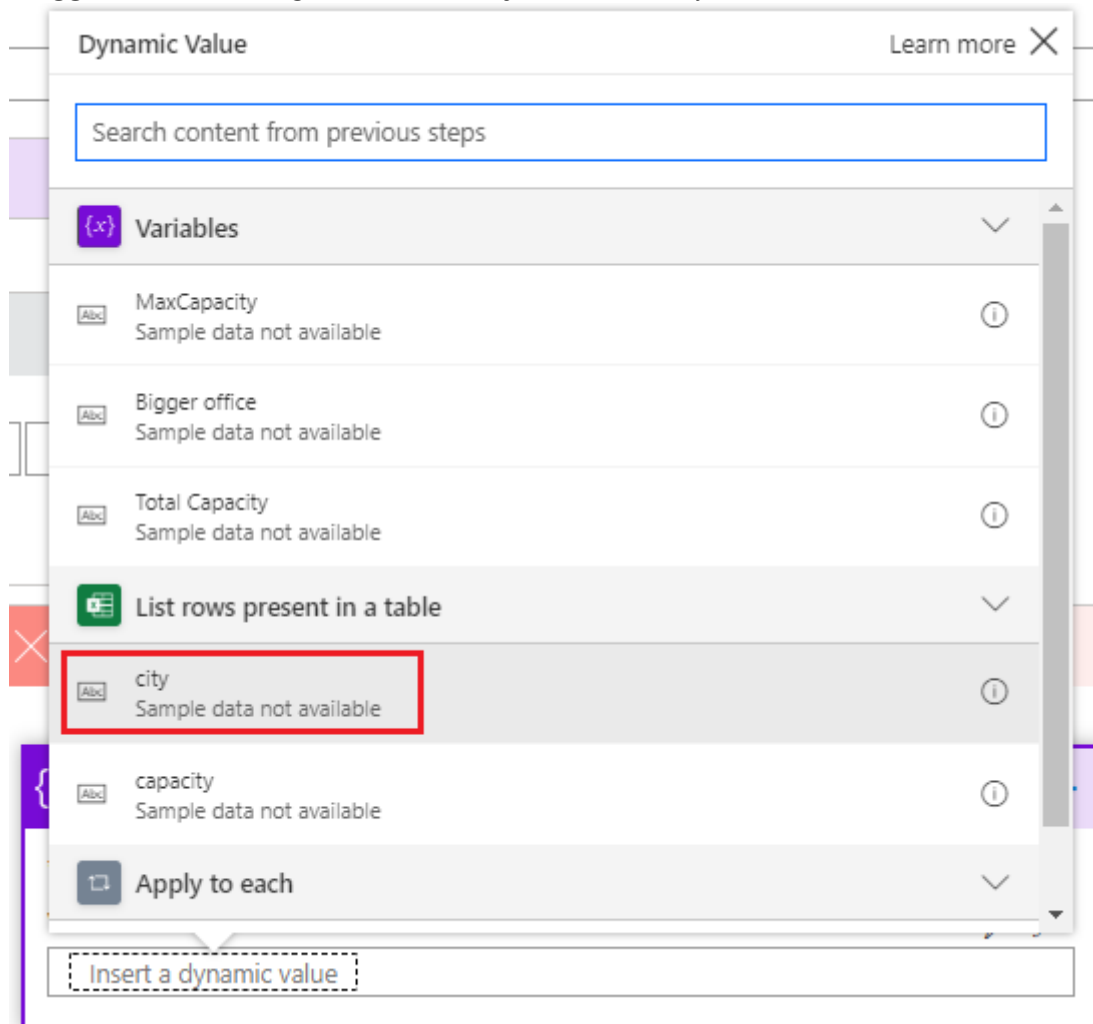
Sample data not available

ⓘ

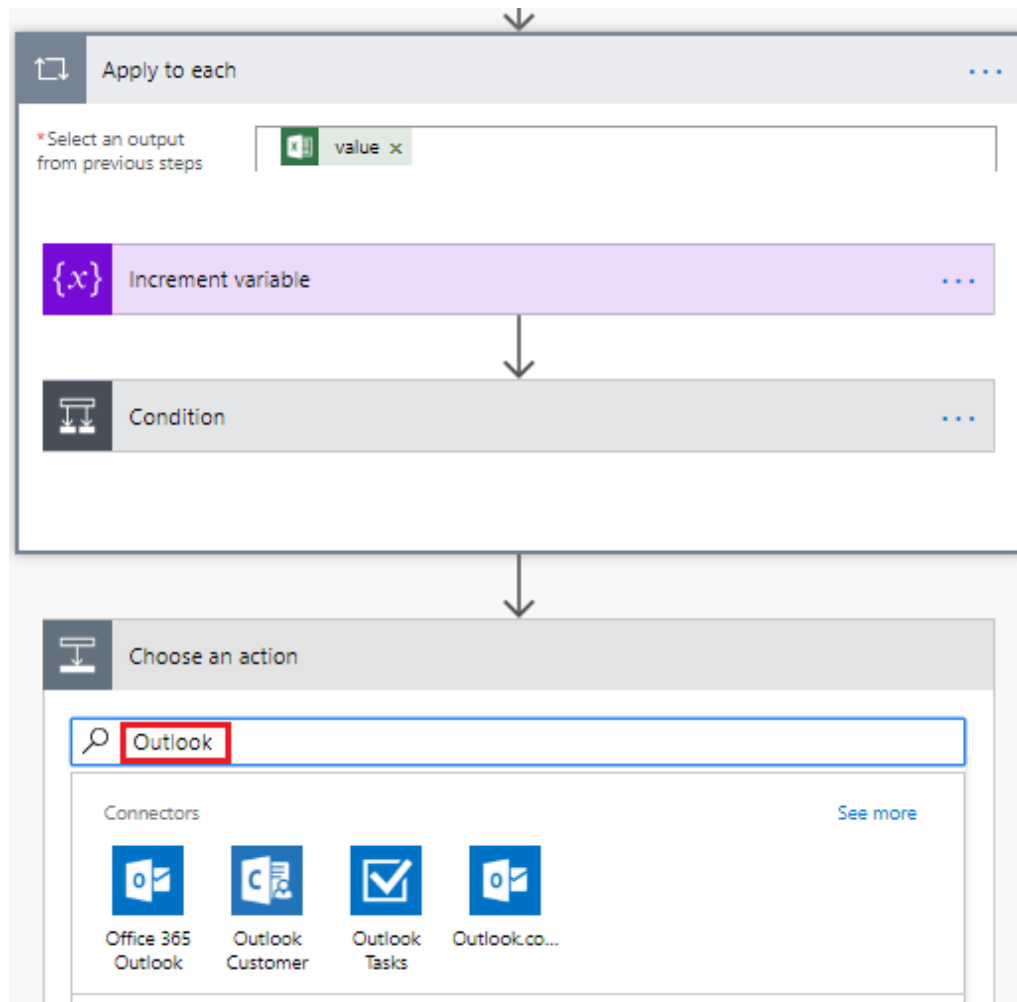
Add an expression

2

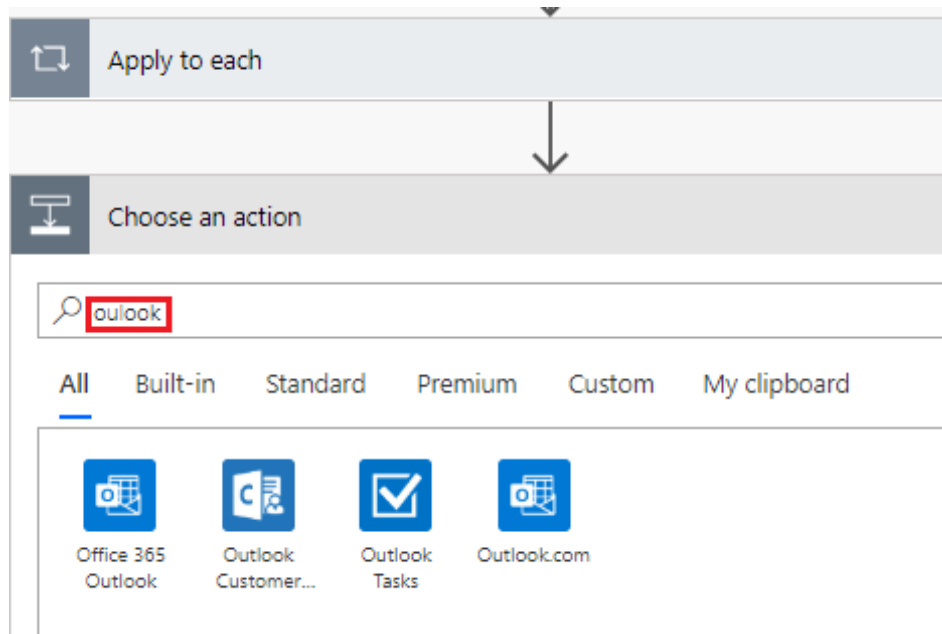
17. In the same left branch of the condition, add another **set variable action** and select the variable **Bigger office** and assign it a value of **city**. Click on the Dynamic value button to retrieve city:



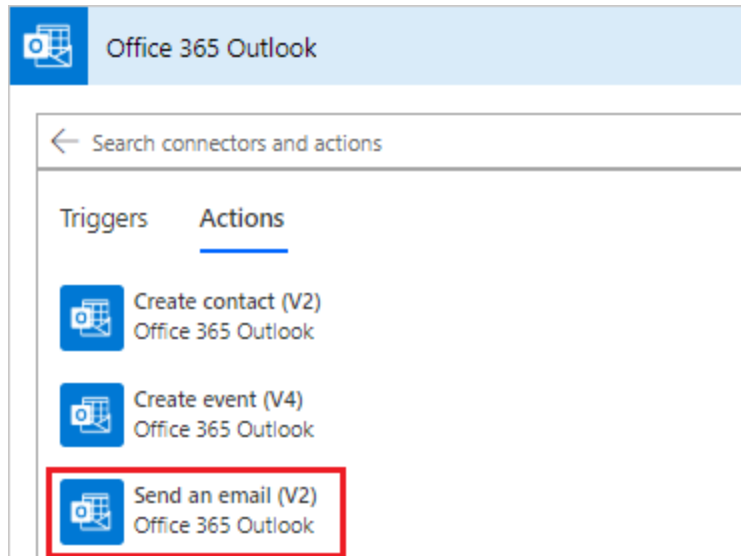
18. Save and test the Flow to figure out which city has the bigger capacity (Toronto in our case). You can debug the Flow or add a notification (or send an e-mail to yourself).
19. Next, let's send an e-mail by adding an **Outlook 365 Outlook - Send an email (v2)** action **after the Apply to each:**
- Find the action by typing Outlook:



b. In the Connectors list click **Office 365 Outlook**:



- c. Select the action **Office 365 Outlook – Send an email**:



- d. Fill-in the Send an email action with the following values
- In the **To field** provide your e-mail address
 - In the **Subject**, type "**Office Capacity Report.**"
 - In the **Body** type the following text:

A screenshot of the 'Send an email (V2)' configuration form. The 'To*' field contains a placeholder 'SL'. The 'Subject*' field contains the text 'Office Capacity Report'. The 'Body*' field contains the text 'The biggest office is:', 'Its capacity is:', and 'The total capacity is:'. Below the body field is a 'Show advanced options' link with a downward arrow. The form has a blue header bar with the Office 365 Outlook icon and the title 'Send an email (V2)'. There are also information and menu icons in the top right corner.

- We will now add the variable's value directly in the **Body**
- Move the cursor just after the colon of *The biggest office is:*

Send an email (V2)

To*

SL

Subject*

Office Capacity Report

Body*

Font 12 **B** *I* U

The biggest office is:
 Its capacity is:
 The total capacity is:

Show advanced options

vi. Click Add a Dynamic value :

Send an email (V2)

To*

SL

Subject*

Office Capacity Report

Body*

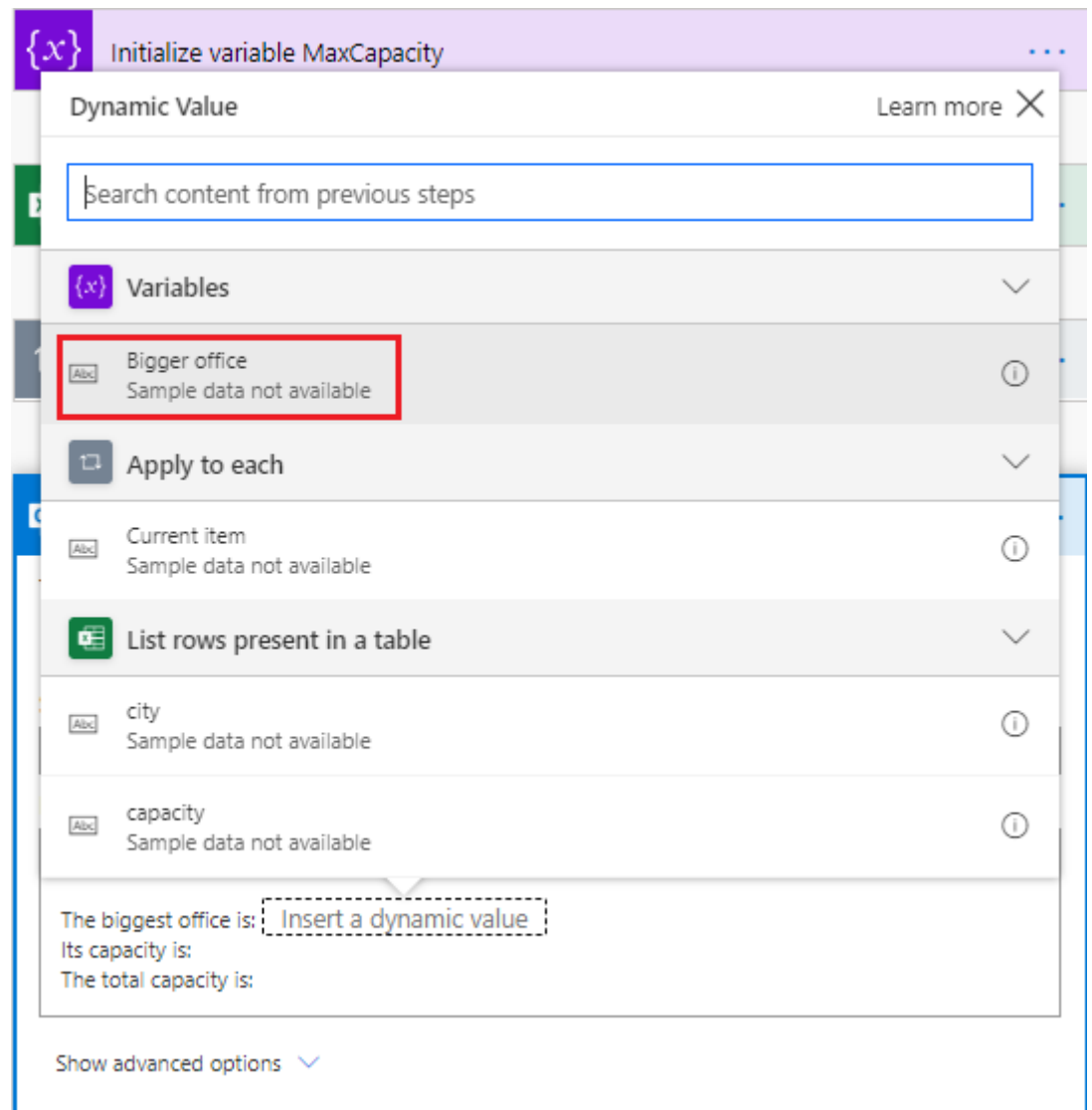
Font 12 **B** *I* U

The biggest office is:
 Its capacity is:
 The total capacity is:


Show advanced options

Add a dynamic value (Ctrl+Space)


vii. In the Variables section select "Bigger office":



viii. The variable name **Bigger office** should now be visible in the body:









 Send an email (V2) (i) ...

To*



Subject*

Body* ⌵ fx

Font ▼ 12 **B** *I* U        


The biggest office is: {x} Bigger office X

Its capacity is:


The total capacity is:

Show advanced options ▼

- ix. Proceed the same way with the other variables **MaxCapacity** and **Total Capacity**.
- x. Eventually, the e-mail body should look like this:

 Send an email (V2) (i)

To*

 serge Luca X

Subject*

Body* ⌵ fx

The biggest office is: {x} Bigger office X

Its capacity is: {x} MaxCapacity X

The total capacity is: {x} Total Capacity X

Show advanced options ▼

20. Save your Flow and test it.

21. Check your e-mail; you should receive something like this:

Office Capacity Report

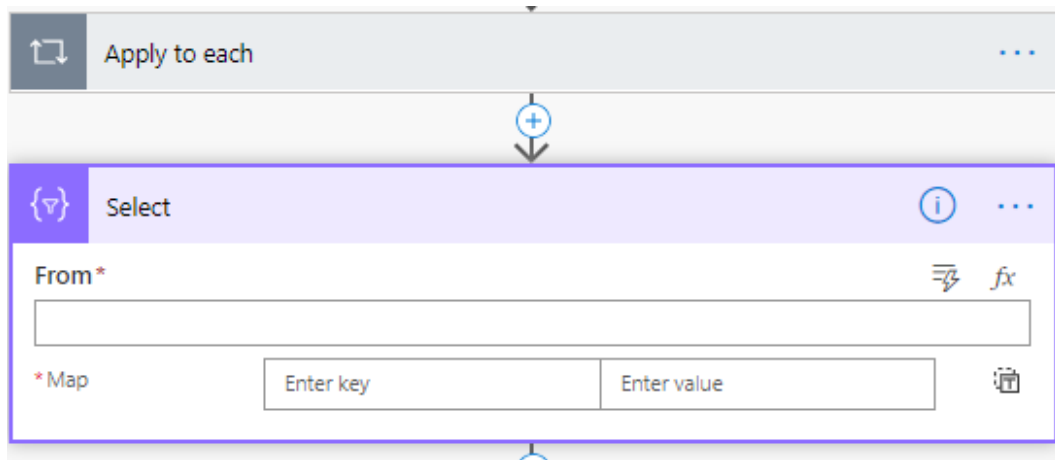


Inbox; Sent Items

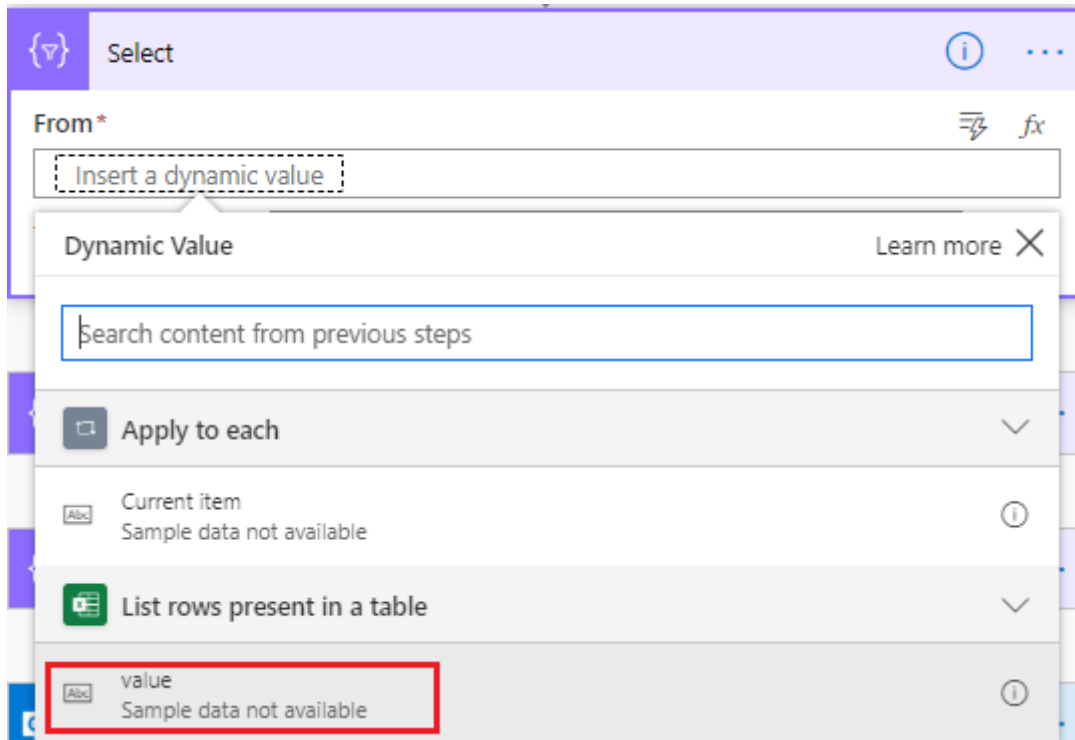
This message was sent with low importance.

The biggest office is : Toronto
Its capacity is : 400
The total capacity is :1607

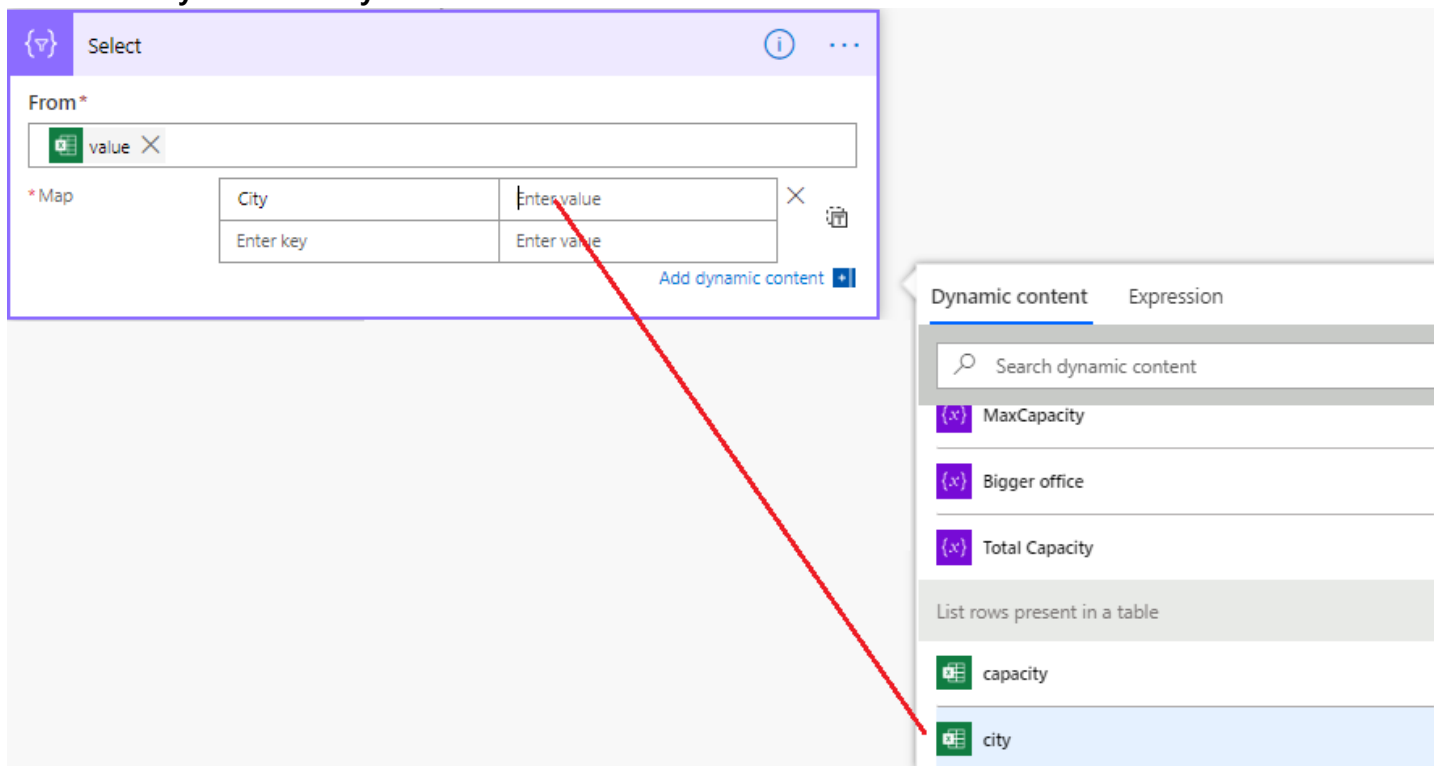
22. In the next steps, we will display the list of offices, so we will have to define a list formatting logic and create an HTML table based on this logic.
23. Let's define the list formatting logic. Before the **Send an email action**, add a **Data Operations – Select** action:



24. Move the cursor in the From field and select the dynamic value associated with **the List rows present in the table** action:



25. In the map field, add the following values: the key field should be **City**, and the value should be the **city** value in the **dynamic value**:



26. Add another map field for **Capacity**:

Select

From*

value X

***Map**

City	city X	X
Capacity	capacity X	X
Enter key	Enter value	

Add dynamic content +

27. Just **after the Select action**, add a **Data Operations - Create HTML table** action:

Select

***From**

value X

***Map**

City	city X	X
Capacity	Capacity X	X
Enter key	Enter value	

+

Create HTML table

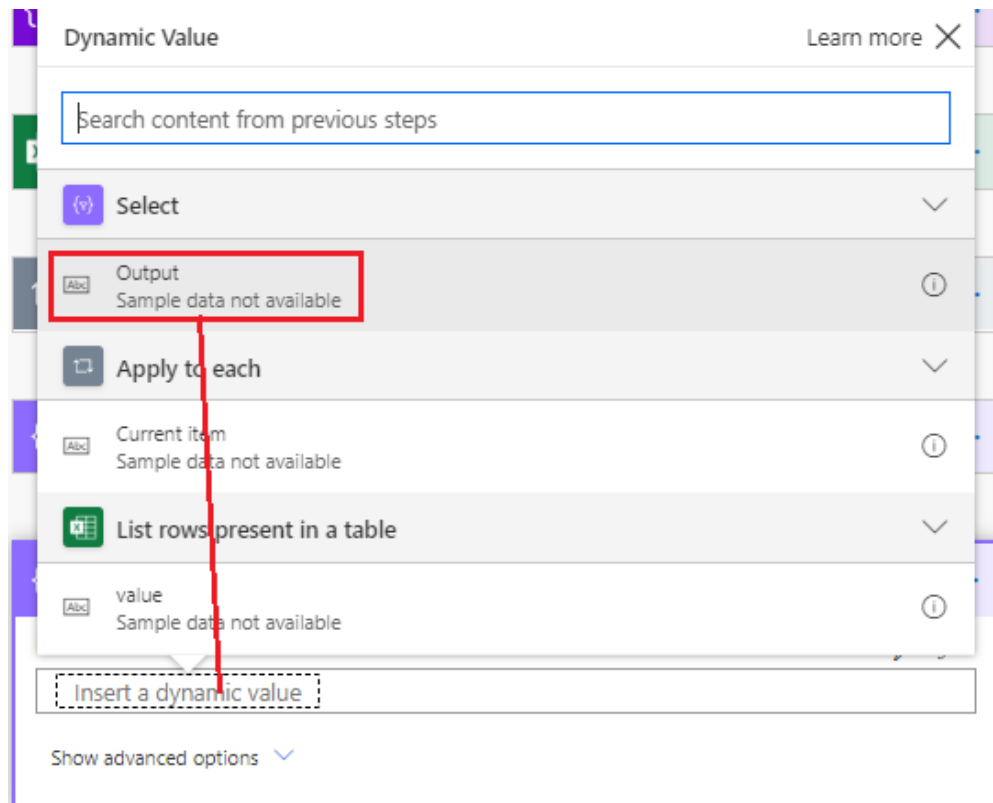
***From**

|




Add dynamic content +

Show advanced options v



28. Move the cursor to the **From** field to show the **dynamic value** panel and click the **Output** value of the **Select** action:



29. Go back to the **Send an email** action and update the **Body** text box to include the Create HTML Output value:

 Send an email (V2)  

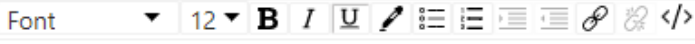
To*



 serge Luca 



Subject*



Office Capacity Report

Body*






The biggest office is:  Bigger office 

Its capacity is:  MaxCapacity 

The total capacity is:  Total Capacity 

The list of offices is:

 Output 

Show advanced options 

30. Test your Flow and check your e-mail:

Office Capacity Report



Today, 3:10 PM

This message was sent with low importance.

The biggest office is : Toronto

Its capacity is : 400

The total capacity is :

List of offices

City	Capacity
London	100
Brussels	250
Seattle	80
Vancouver	200
Toronto	400
Antwerpen	15
Warsaw	300
Paris	54
Berlin	70
Amsterdam	60
Montreal	78

Optional exercise if time permits use an object instead of dedicated variables.

You can remove any reference and definition of **MaxCapacity** and **bigger Office**.

Create a variable **Max Office** as a variable of type "Object" containing the following JSON data:

Recurrence ...

↓

{x} Initialize variable Max Office ...

Name*

Max Office

Type*

Object

Value

```
{
  "Bigger Office": "",
  "Max Capacity": 0
}
```

31. In the condition (In the existing **Apply to each**), use this **Max Office** variable content (do not forget to use the **int** function as well:

Condition ...

int(...) × is greater than **int(...)** ×

[Add dynamic content](#) +

+ Add ▾

Dynamic content **Expression**

int(variables('Max Office')['Max Capacity'])

Update

If no

32. In the **if yes** branch condition, use a **Set variable** that will update **Max Office**:

Condition

fx int(...) \times is greater than fx variables... \times

+ Add \vee

✓ If yes

{x} Set variable Max Office

* Name Max Office

Value *

```
{
  "Bigger Office": "",
  "Max Capacity": 0
}
```

✗ If no

Ac

33. Now let set the Bigger Office value of the object (within the double quotes): move the cursor between the quotes:

✓ If yes

{x} Set variable Max Office

* Name Max Office

Value *

```
{
  "Bigger Office": †,
  "Max Capacity": 0
}
```

34. Click **Add An expression**:

✓ If yes

{x} Set variable Max Office

* Name Max Office

Value*

```
{
  "Bigger Office": †,
  "Max Capacity": 0
}
```

Add an expression (Ctrl+Shift+Space)

35. Type the following expression:

✓ If yes

{x} Set variable Max Office

* Name Max Office

Value*

```
{
  "Bigger Office": Add an expression,
  "Max Capacity": 0
}
```

Expression List of functions X

item()['city']

36. Do the same with the field **Max Capacity** of the object (we do not need to have double quotes around the value here):

✓ If yes

{x} Set variable Max Office

* Name Max Office




Value *

```
{  
  "Bigger Office": "city",  
  "Max Capacity": "Add an expression"  
}
```



Expression List of functions

item()['capacity']

37. Remove the actions **Set Variable MaxCapacity** and **set variable Bigger office** previously defined.
38. Now you will update the **send an e-mail** action with the new variable. Remove the reference to the old variables (just keep the reference to Total Capacity and the output):



 Send an email (V2)  













To*

 sergeluca@ShareQL.com 



Subject*

Office Capacity Report



Body*  


Font  12           

The biggest office is:
Its capacity is:

The total capacity is:  Total Capacity 


The list of offices is:

 Output 


[Show advanced options](#) 



39. Set the value of the biggest office in the mail: add an expression like this:


Expression


List of functions 



`variables('Max Office')['Bigger Office']`


 Search content from previous steps


 Create HTML table 



 Output

Sample data not available 

 Apply to each 

 Current item



Sample data not available 

 List rows present in a table 



The biggest office is:

Add an expression

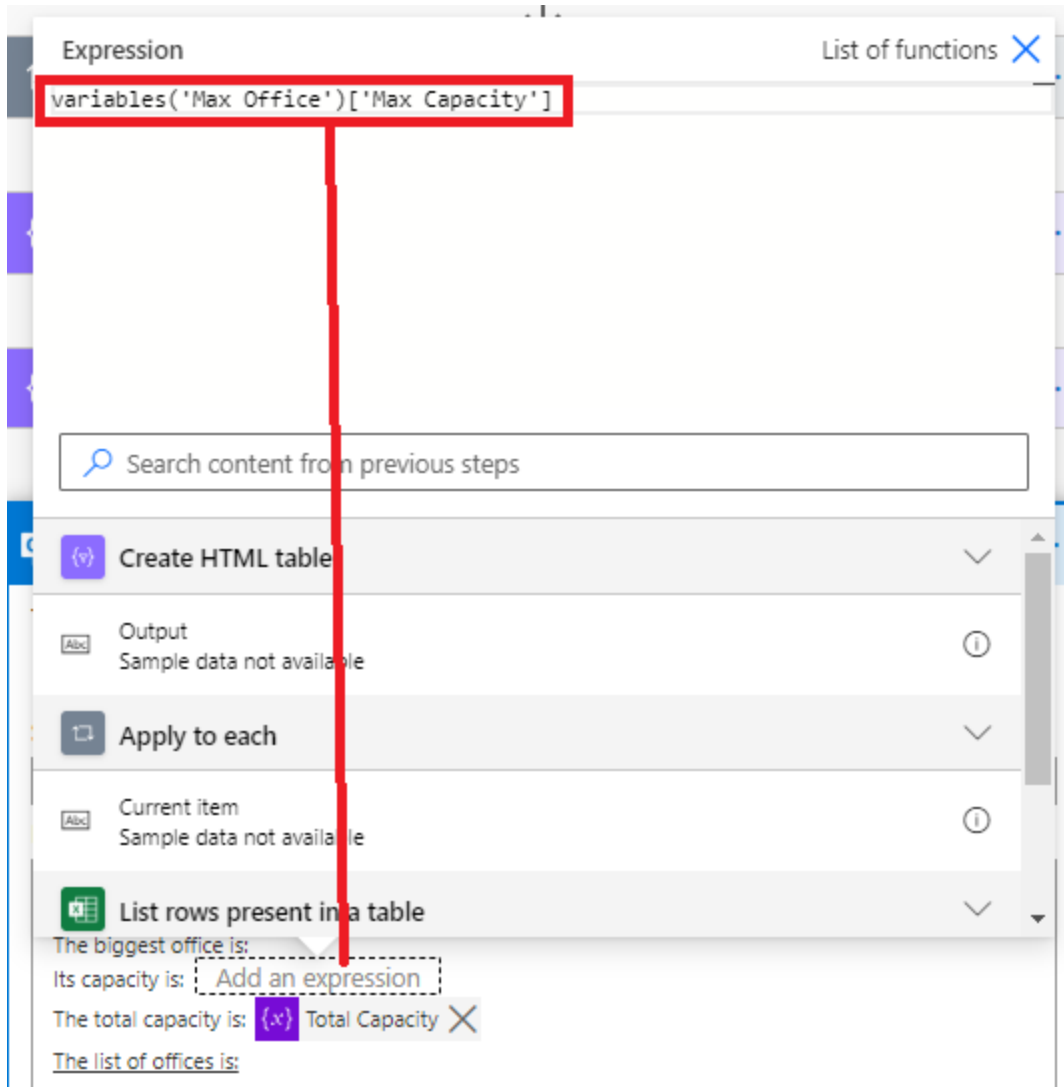
Its capacity is:

 variables(...) 

The total capacity is:

 Total Capacity 

The list of offices is:



40. Test the flow :-).

41. Question: how would you update the report in order to add a new column named "Size" displaying "Small" if < 100 and "Big" if >= 100 ?

Answer: in the Select Action, add a new row "size" associated with the following expression:

```
if ( greater(int(item()['Capacity']), 100), 'Big', 'Small')
```

{}

Select

...

From*

value

×

* Map

City	city	×
Capacity	capacity	×
Size	if(...)	×
Enter key	Enter value	

Add dynamic content +

42. Question: how would you format the generated Html table to define a border?

Answer: first let us analyze the html generated code by clicking on an existing flow run and then click the option "Show raw outputs" of the Create HTML Table action as illustrated below:

{}

Create HTML table

0s

INPUTS

Show raw inputs >

Format

Html

From

```
[
{
  "City": "London",
  "Capacity": "100",
  "Size": "Small "
},
{
  "City": "Brussels"
```

OUTPUTS

Show raw outputs >

Body

The html code looks like this:

Outputs

Create HTML table

```
{  
  "body": "<table><thead><tr><th>City</th><th>Capacity</th><t
```

You just need to replace **<table>** with **<table border="1">**. You can do it by using the replace function:

The screenshot shows the 'Expression' step editor in Power Automate. The 'Expression' field contains the formula: `replace(body('Create_HTML_table'),'<table>','<table border="1">')`. Below the editor is a search bar labeled 'Search content from previous steps'. A list of previous steps is shown, including 'Variables', 'Bigger office', 'Create HTML table', 'Output', 'Apply to each', and 'replace(...)'. The 'replace(...)' step is highlighted at the bottom of the list.

Expression List of functions ✕

`replace(body('Create_HTML_table'),'<table>','<table border="1">')`

Search content from previous steps

- Variables
- Bigger office
Sample data not available
- Create HTML table
- Output
Sample data not available
- Apply to each
- replace(...) ✕

[Show advanced options](#) ✓

The generated e-mail will look like this:

Office Capacity Report



This message was sent with Low importance



The biggest office is: Montreal

Its capacity is: 400

The total capacity is: 1607

The list of offices is:

City	Capacity	Size
London	100	Small
Brussels	250	Big
Seattle	80	Small
Vancouver	200	Big
Toronto	400	Big
Antwerpen	15	Small
Warsaw	300	Big
Paris	54	Small
Berlin	70	Small
Amsterdam	60	Small
Montreal	78	Small

We need your feedback

Do you want to report an issue or to suggest something? We need your feedback:

<https://github.com/Power-Automate-in-a-day/Training-by-the-community/issues>