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

Updated by: Dattatray-Patil

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:

City	Capacity	Excel Image
London	100	
Brussels	250	
Seattle	80	
Vancover	20	
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

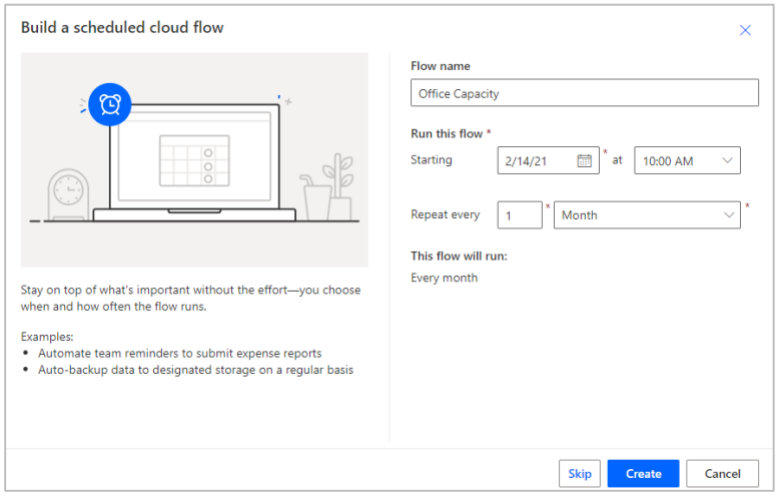
SL [redacted] Today, 3:10 PM

This message was sent with low importance.

The biggest office is : Toronto
Its capacity is : 400
The total capacity is :
List of offices

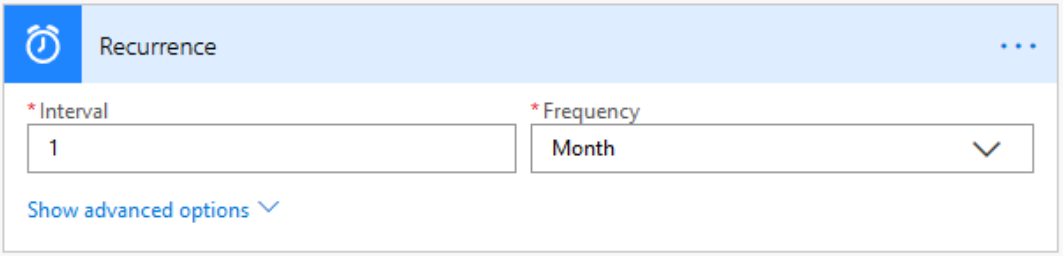
City	Capacity
London	100
Brussels	250
Seattle	80
Vancover	200
Toronto	400
Antwerpen	15
Warsaw	300
Paris	54
Berlin	70
Amsterdam	60
Montreal	78

2. Create a flow to generate this email report.
    a. Navigate to the **Microsoft Flow portal**.
    b. **New flow** > **Scheduled cloud flow**
    c. Use the following screenshot to name the flow, and set the flow frequency.

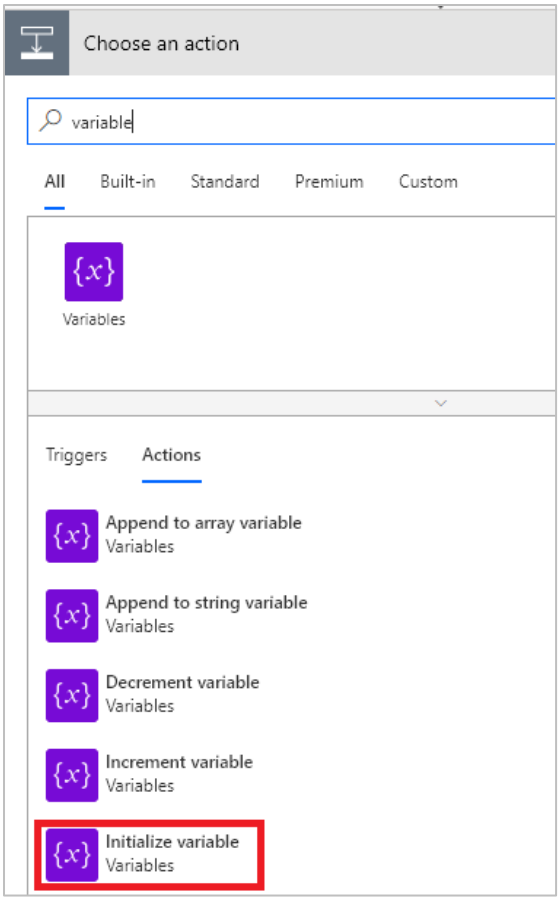


    d. Click **Create**.

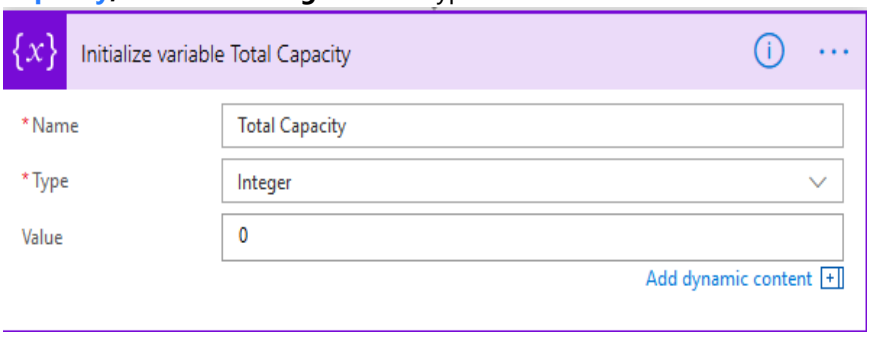
The following flow will be generated:



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. Select **New Step** and select **Initialize variable**:



   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**:

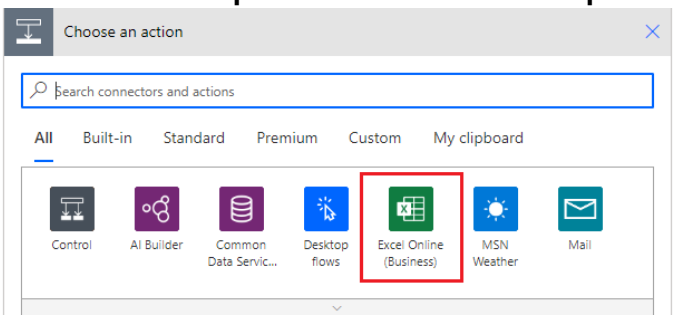


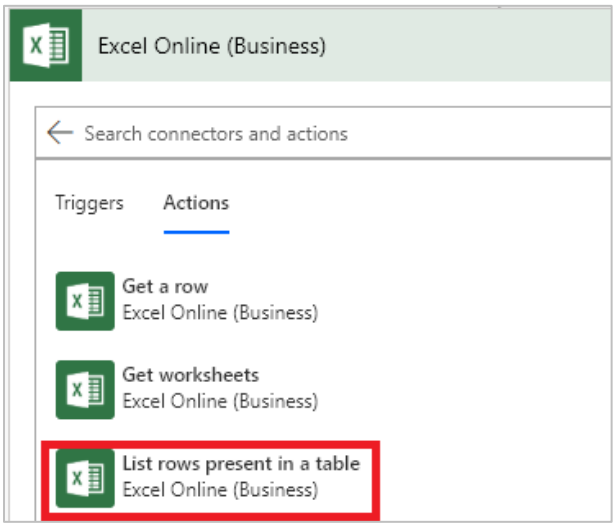
4. Click on **Save** button to **Office Capacity** flow .

Task 4.2: Extend the flow to loop through all offices

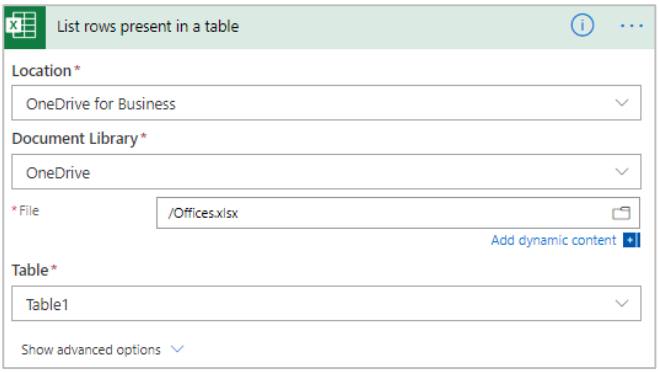
In this task you will update the flow; make it loop through all offices, retrieve their capacity, and increment the Global Capacity variable to calculate the total capacity.

1. To retrieve the list of offices.
    a. **Select New step** > **Excel Online** > **List rows present in a table**:

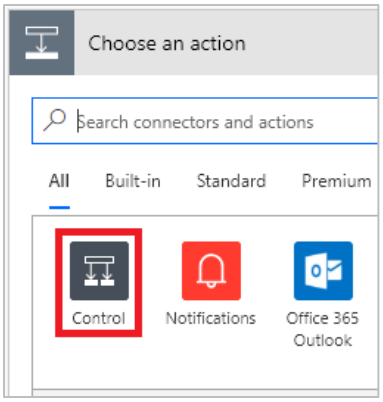




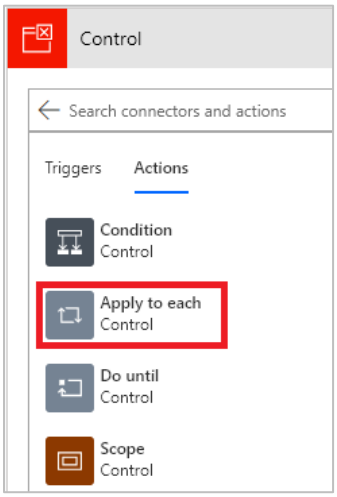
    b. Set the Excel action's properties as per the next screenshot:



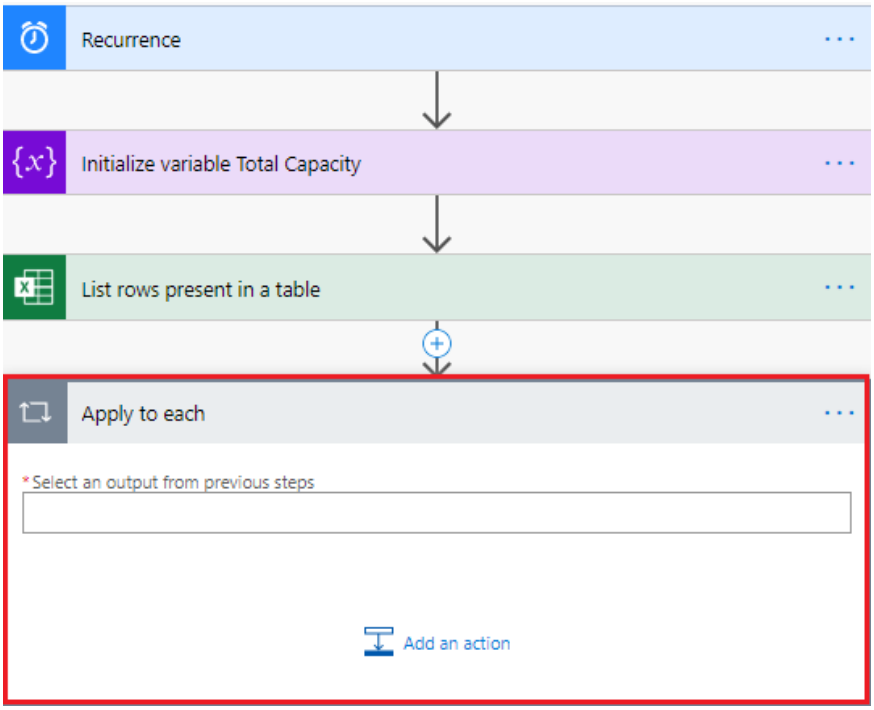
2. Loop through the cities
    a. **Select New step** > **Control**



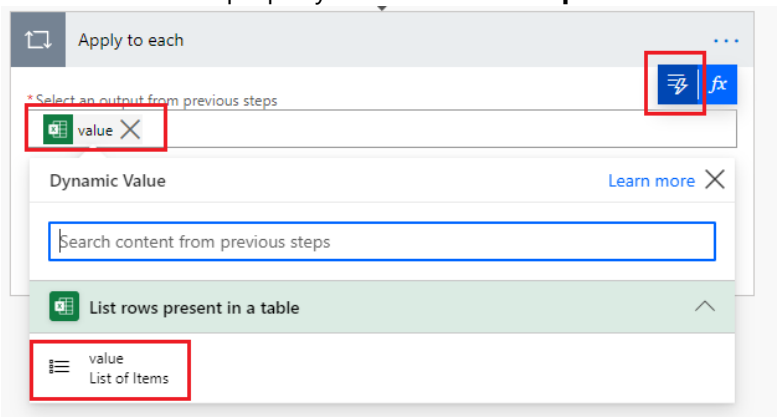
b. Click **Apply to each:**



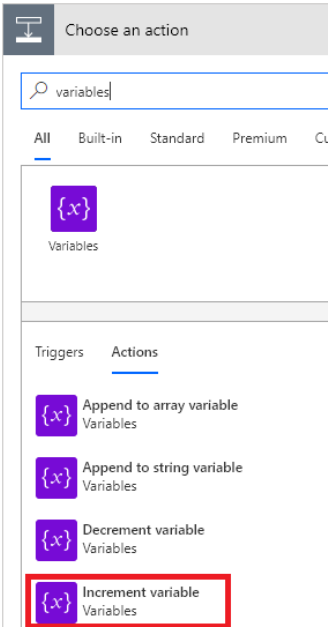
Your Flow should now look like this:



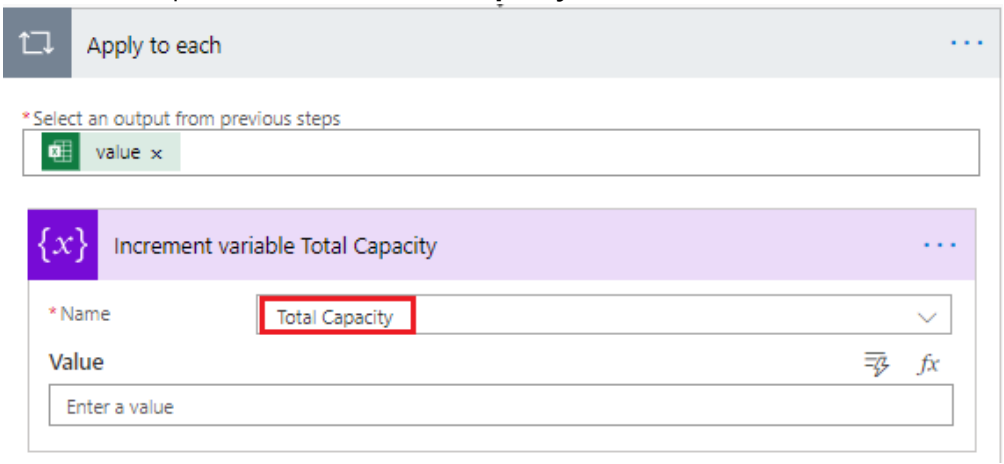
3. Configure the **Apply to each** action (it expects a list of values), using the "Add a dynamic value" to select the **value** property from the **List rows present in a table** action.



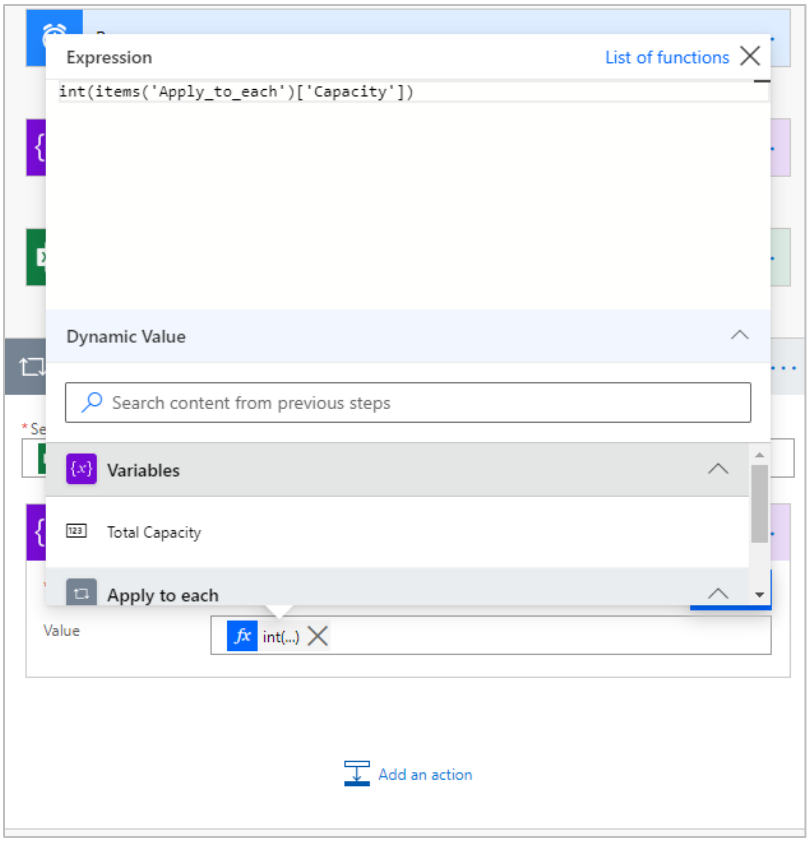
4. Calculate the current office capacity using a variable and an expression.
    a. In the **Apply to each** action, click **Add an action > Increment variable**:



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(items('Apply_to_each')['Capacity'])` and click **Add an expression :**

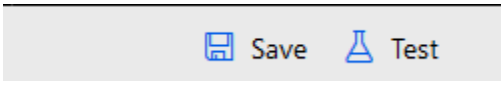


Note:

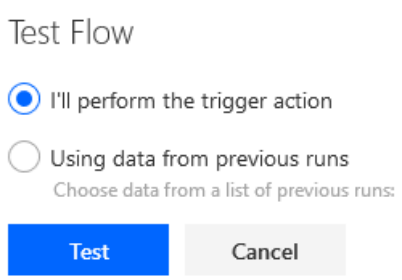
The **item('Apply_to_each')** expression retrieves the current record information in the current loop and **['Capacity']** provides the field name to retrieve. **Item('Apply_to_each')['Capacity']** returns a string. To transform a string to an integer (because we need to increment it), we use the **int()**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

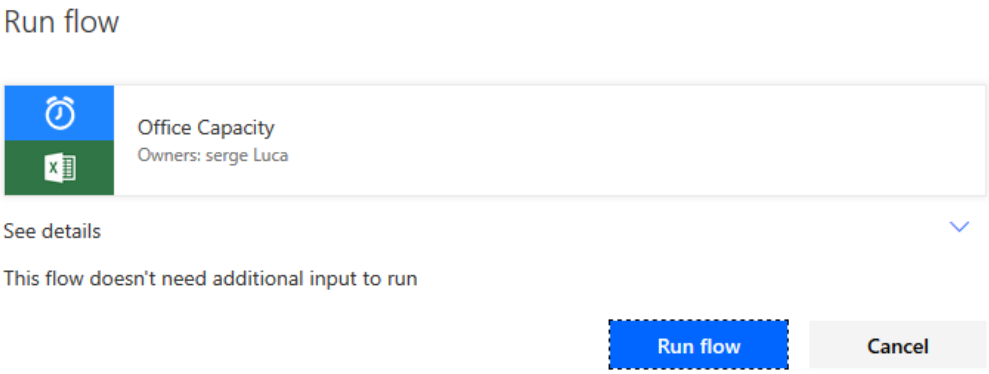
5. To test the flow, without waiting one month before the flow starts, use the **Test** button to manually start the flow on demand (in test mode). This is convenient for testing and debugging purposes.
    a. Click **Test**.



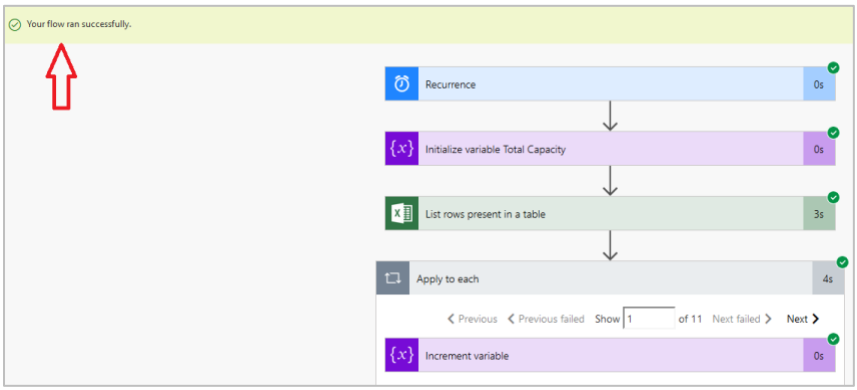
    b. Select **I'll perform the trigger action**:



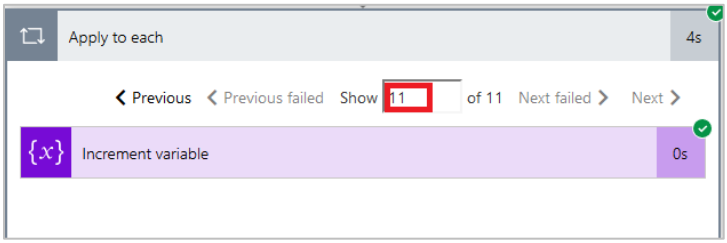
c. Click on **Run Flow**:



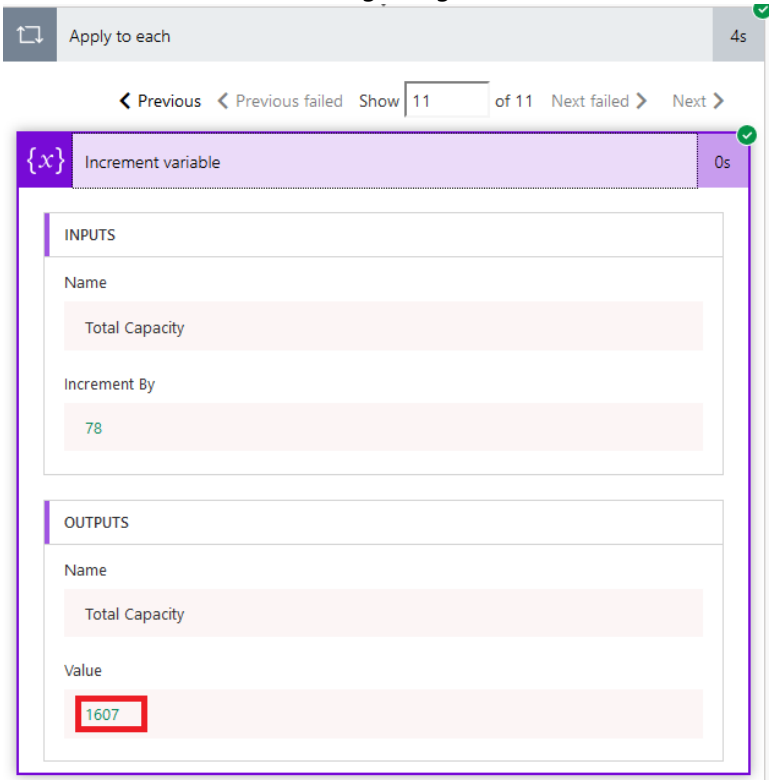
d. Wait until you get the message: **Your Flow ran successfully.**



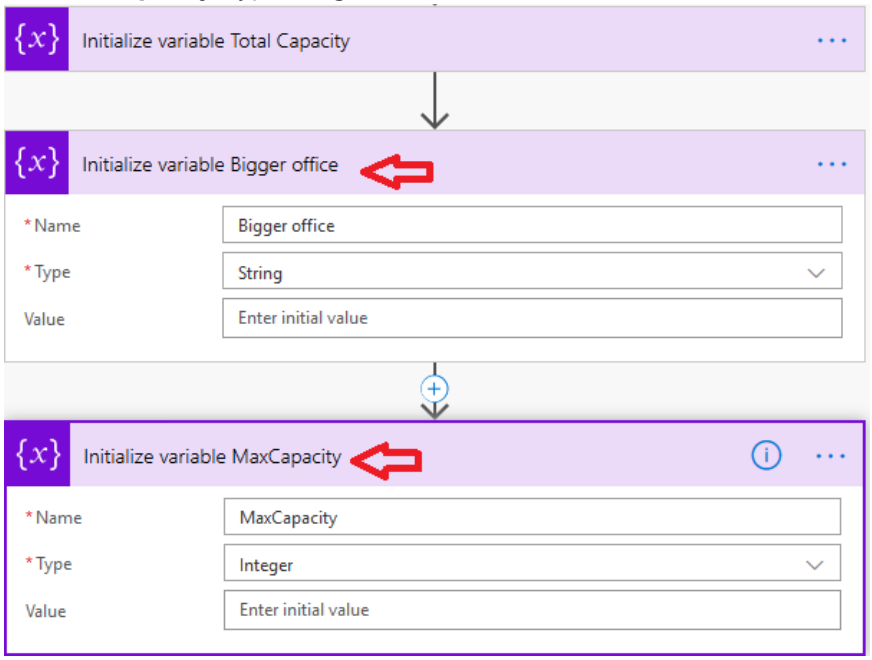
e. To check the **Total Capacity** value, you can examine the value of **Total Capacity** for each step. For example, in our case, we will check its value once it has completed the loop 11 times: so, type **11** In the **Show** textbox:



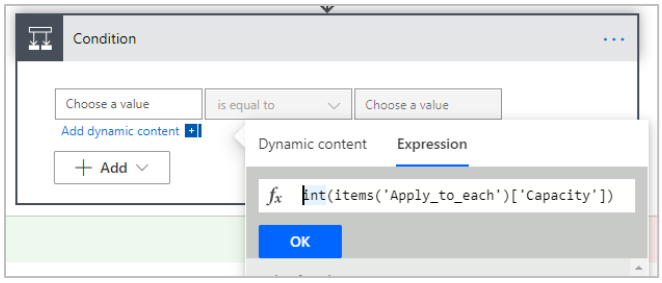
f. Click **Increment variable** to display a value of **1607** (if you use the values in the Excel workbook as defined at the beginning of the lab.



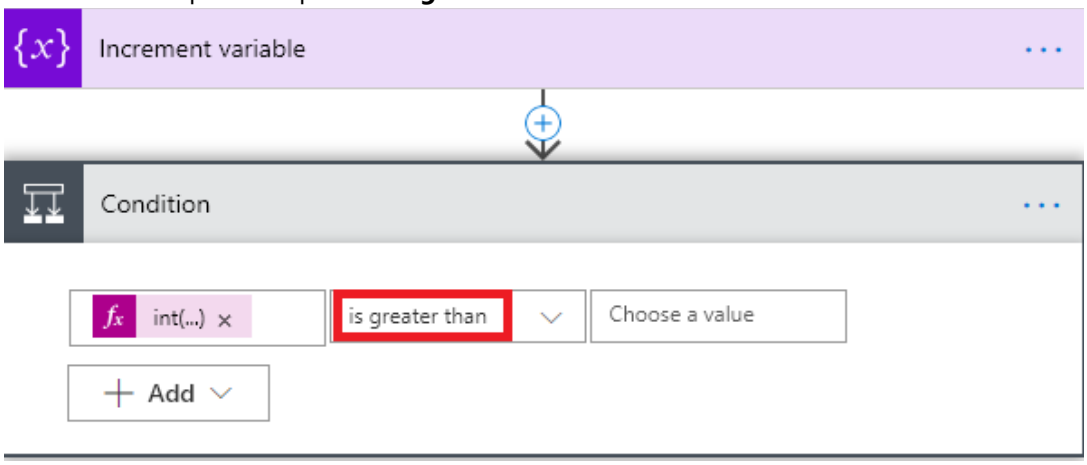
6. Define 2 new variables
   a. Below the variable, **Total capacity** and before the loop, add two new variables named:
      - **Bigger Office** (type string)
      - **MaxCapacity** (type integer)



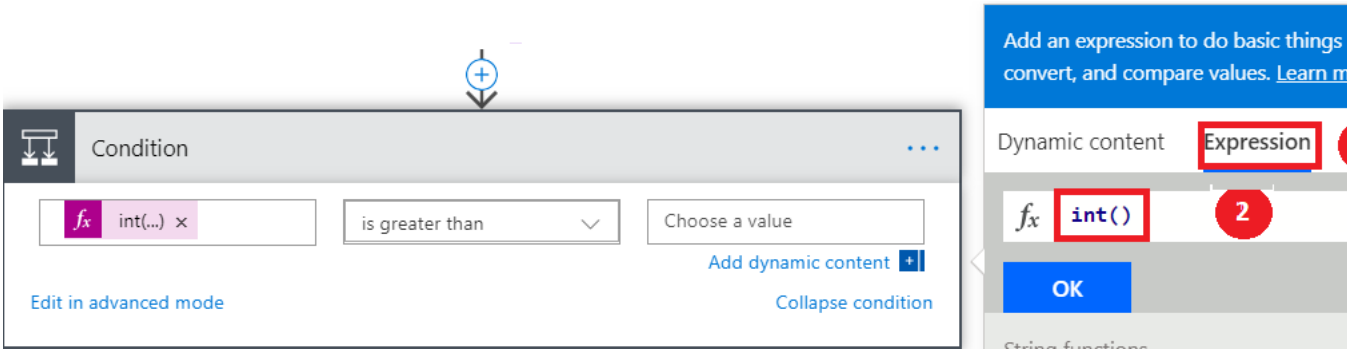
7. **Add a Condition** (from) the Control connector) in the **Apply to each** action:
8. The goal is to compare 2 numbers and to select the larger one. In order to do so, we need to transform our capacity values into integers. In the left side of the condition, click **Choose a value** and click on **Expression**.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



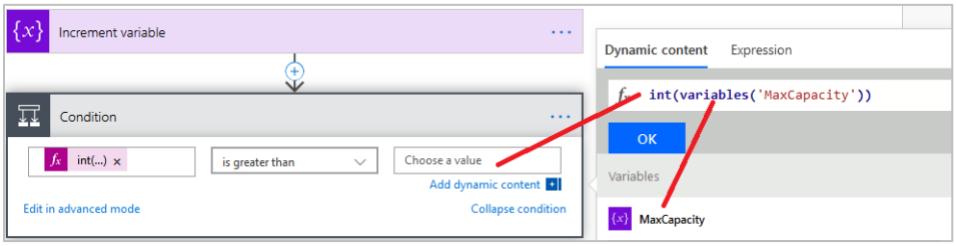
9. Click **Ok.**
10. Select the comparison operator **is greater than**:



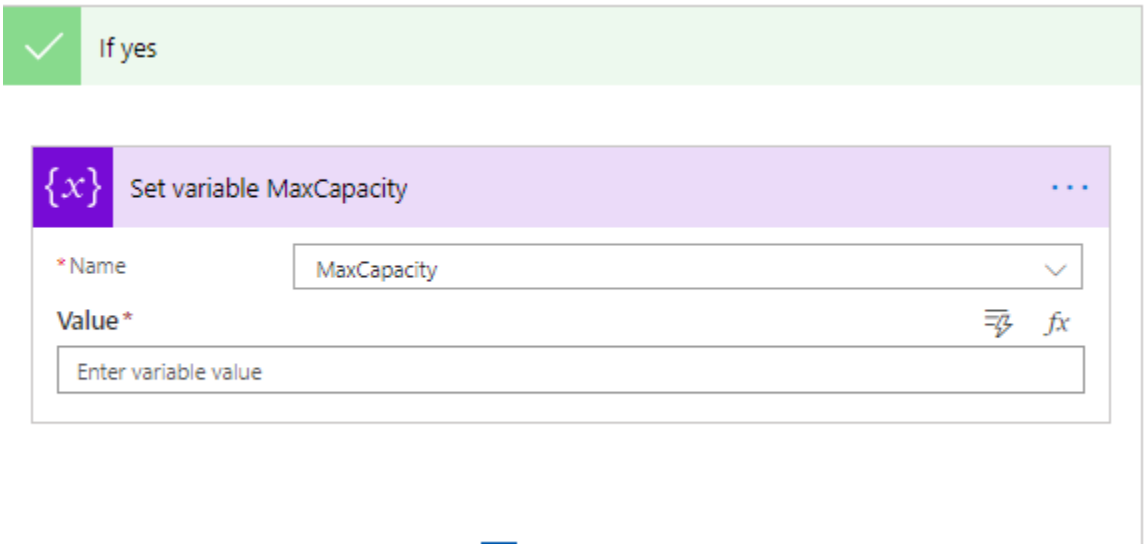
11. In the **Choose a value** textbox, we will include an expression much as we did before by using the **int()** expression, but with a small variation. Click choose a value, click on **Expression** and type **int()**



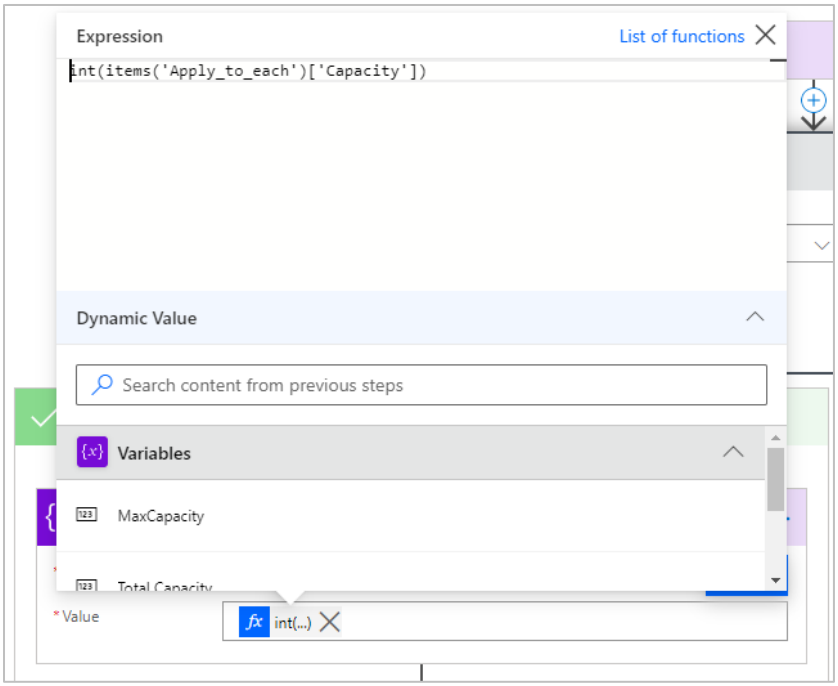
12. Move the cursor within the **int()** parentheses.
13. Click **Dynamic content expression**, select the **MaxCapacity** variable: the editor will automatically generate the expression int(variables('MaxCapacity')). Click **Ok** and **Save** the Flow.



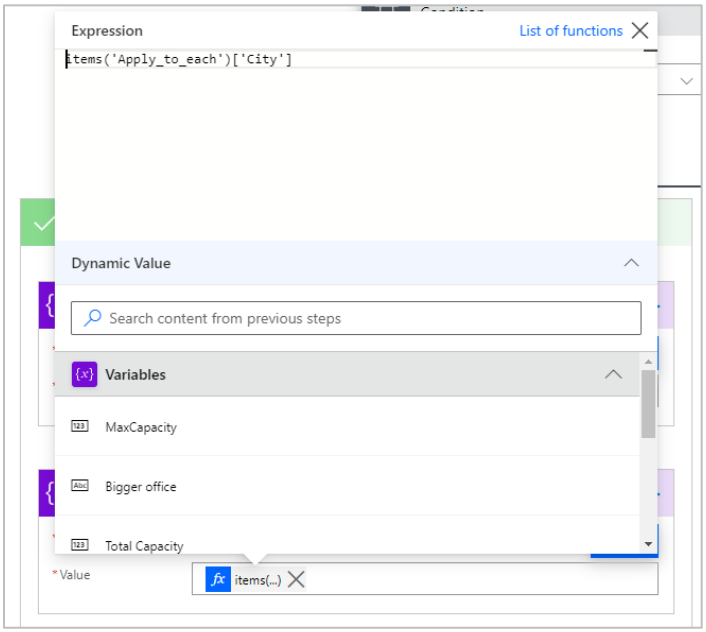
14. Now, in the left **If yes** branch, add a new action **Variables – Set variable** for our **MaxCapacity** variable.
15. Rename the action **Set variable MaxCapacity:**



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

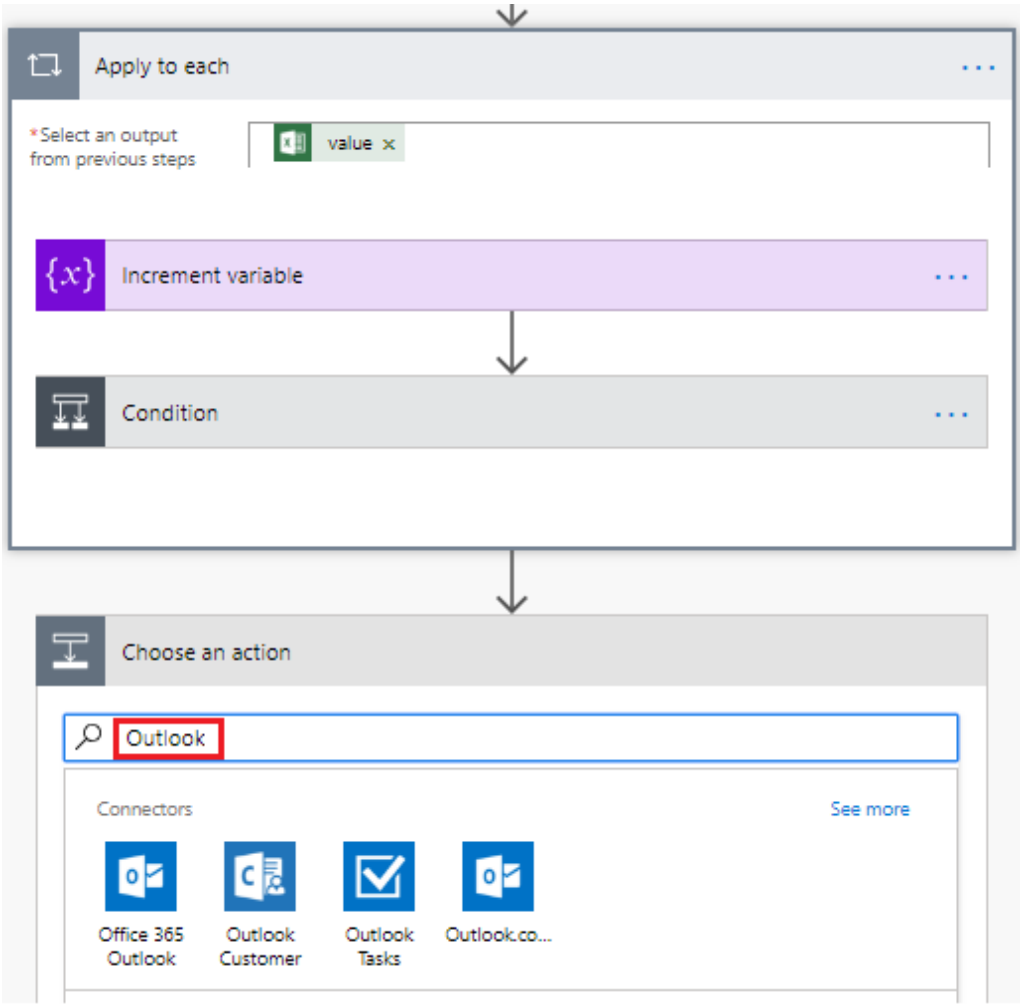


17. In the same left branch of the condition, add another **set variable action** and select the variable **Bigger office** and assign it a value of **City**. Click on the Dynamic value button to retrieve city:

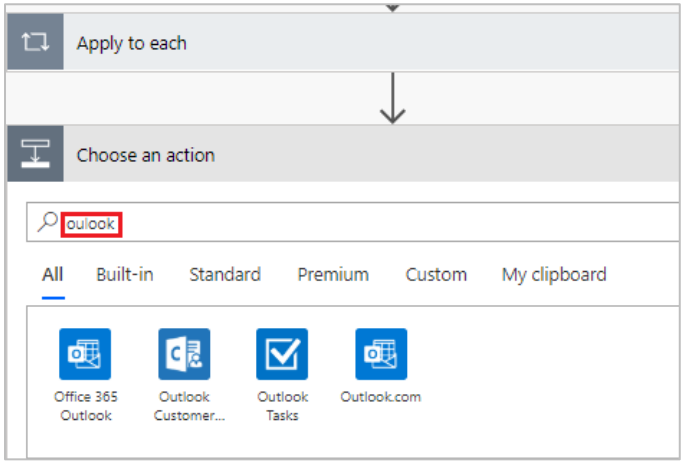


18. Save and test the Flow to figure out which city has the bigger capacity (Toronto in our case). You can debug the Flow or add a notification (or send an e-mail to yourself).
19. Next, let's send an e-mail by adding an **Outlook 365 Outlook - Send an email (v2)** action **after the Apply to each:**

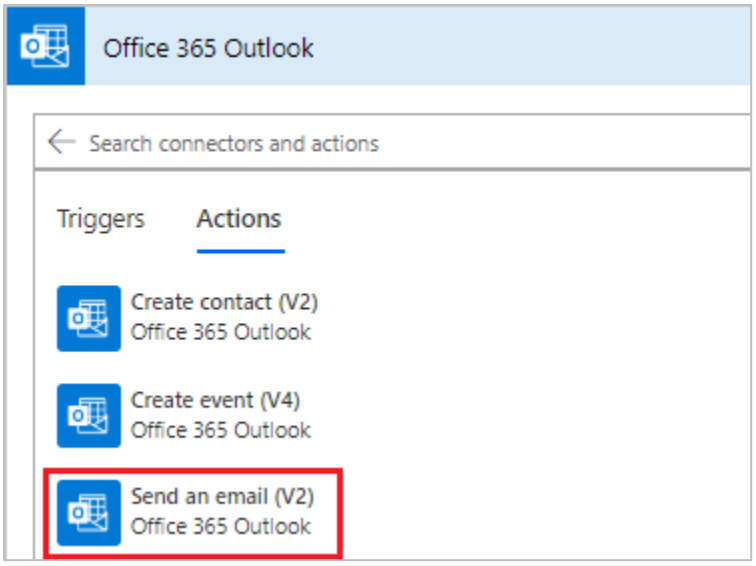
a. Find the action by typing Outlook:



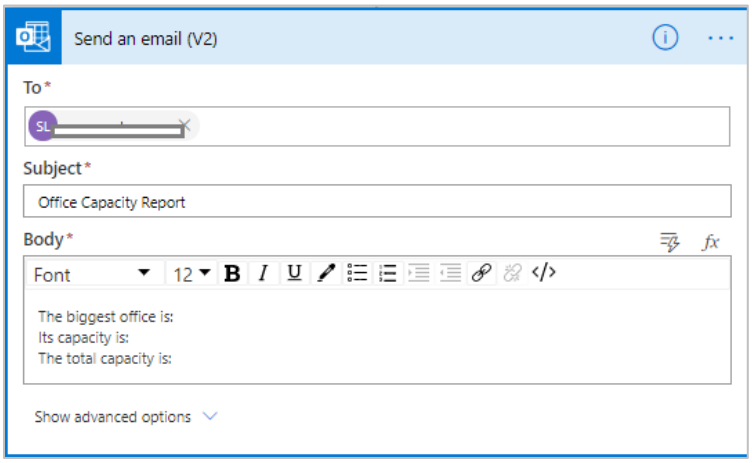
b. In the Connectors list click **Office 365 Outlook**:



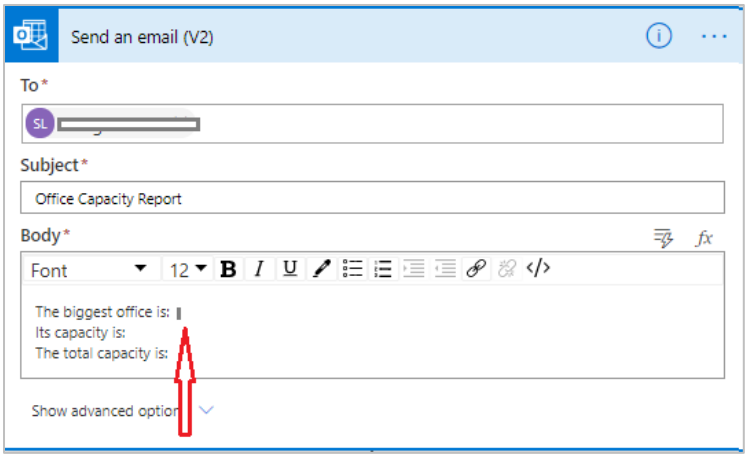
c. Select the action **Office 365 Outlook – Send an email**:



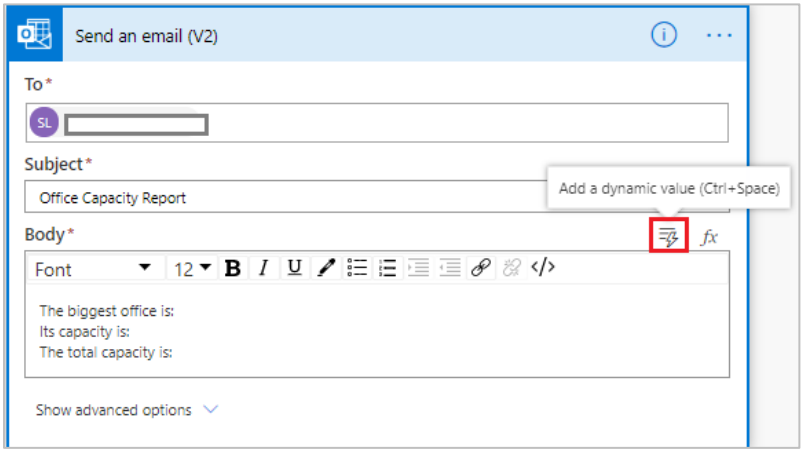
d. Fill-in the Send an email action with the following values
   i. In the **To field** provide your e-mail address
   ii. In the **Subject**, type "**Office Capacity Report**."
   iii. In the **Body** type the following text:



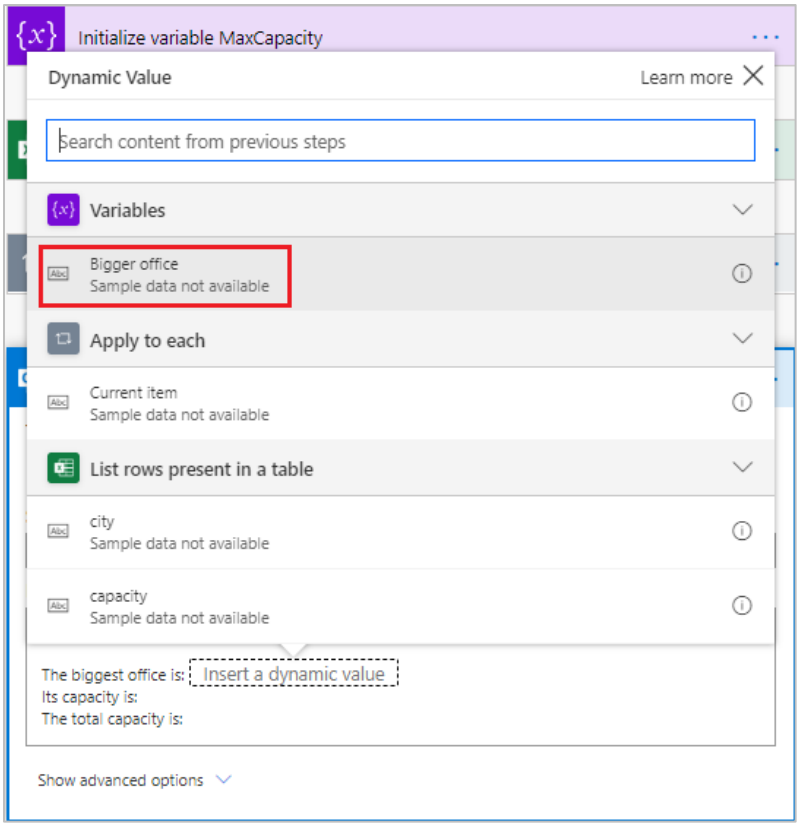
   iv. We will now add the variable's value directly in the **Body**
   v. Move the cursor just after the colon of *The biggest office is:*



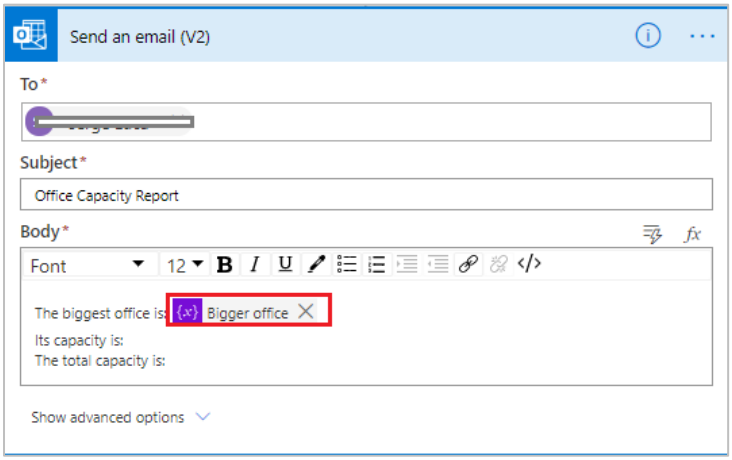
vi. Click Add a Dynamic value :



vii. In the Variables section select “Bigger office”:

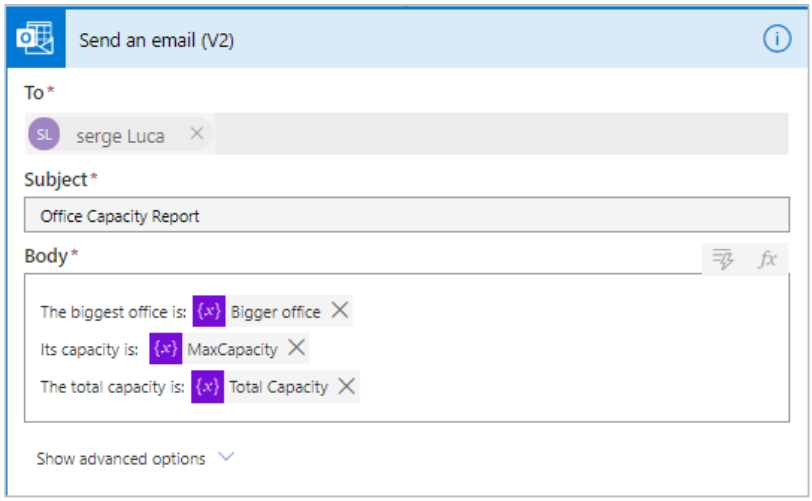


viii. The variable name **Bigger office** should now be visible in the body:

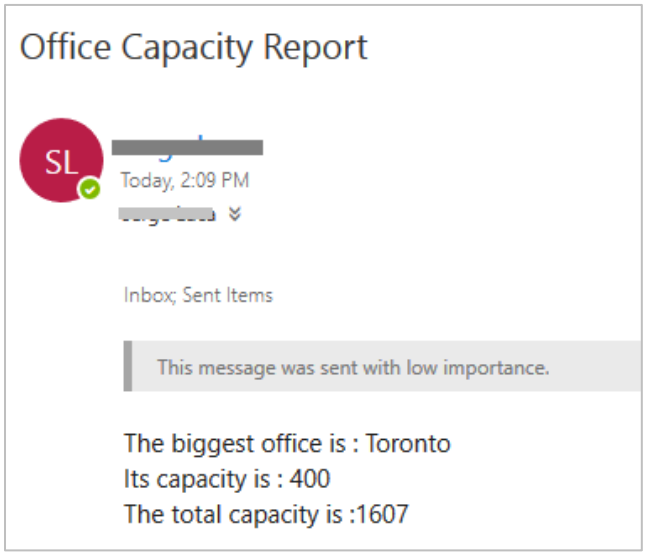


ix. Proceed the same way with the other variables **MaxCapacity** and **Total Capacity**.

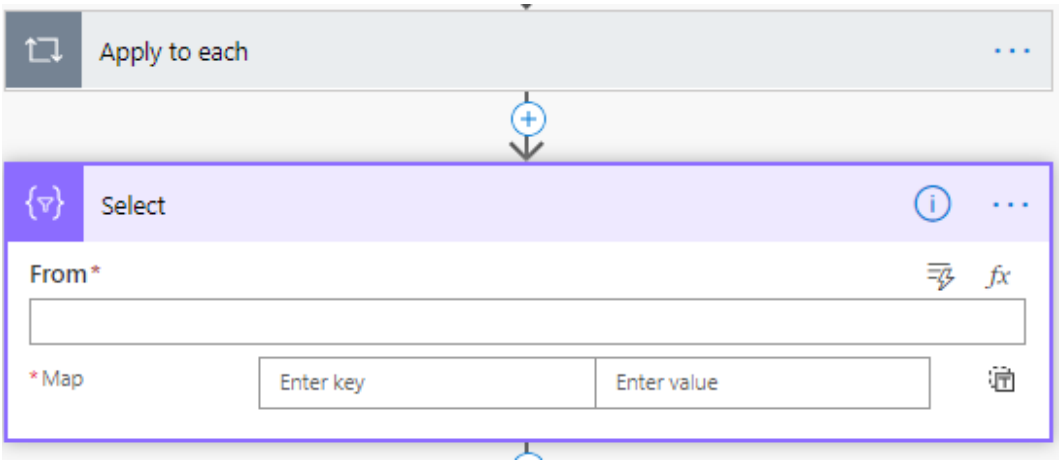
x. Eventually, the e-mail body should look like this:



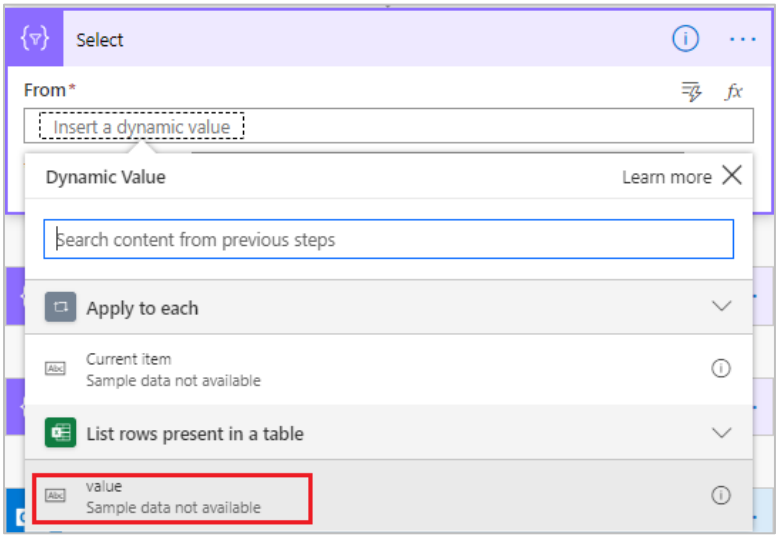
20. Save your Flow and test it.
21. Check your e-mail; you should receive something like this:



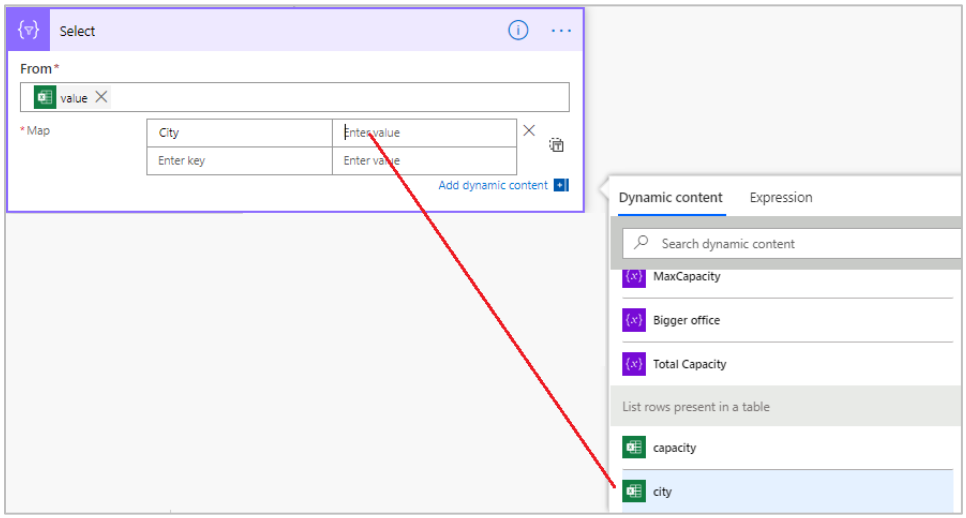
22. In the next steps, we will display the list of offices, so we will have to define a list formatting logic and create an HTML table based on this logic.
23. Let's define the list formatting logic. Before the **Send an email action,** add a **Data Operations – Select** action:



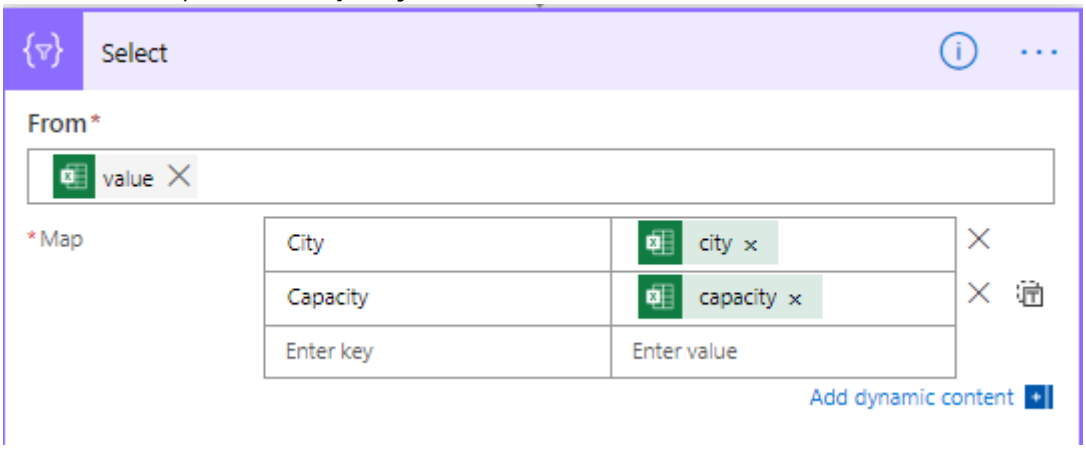
24. Move the cursor in the From field and select the dynamic value associated with **the List rows present in the table** action:



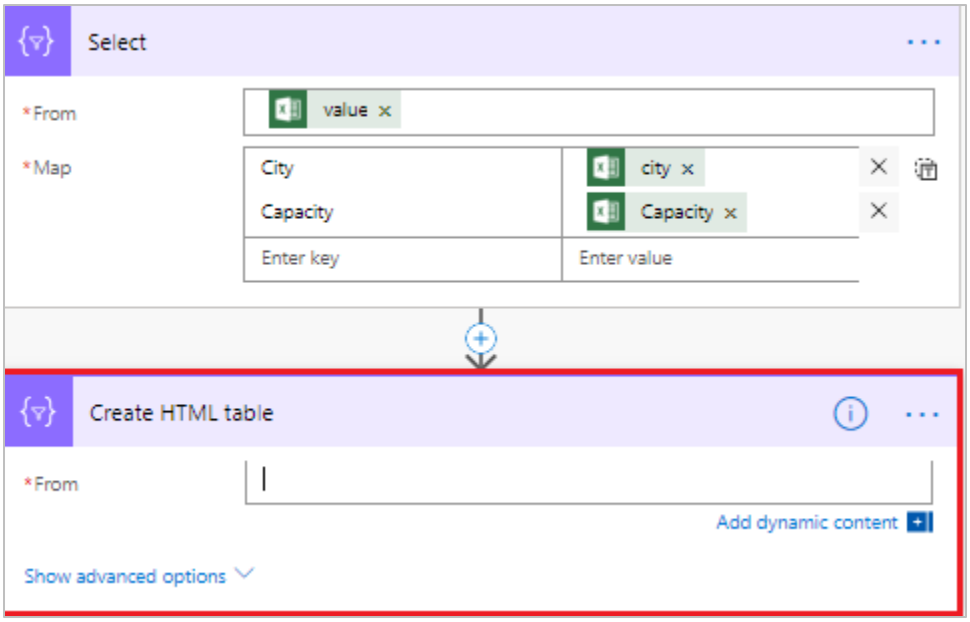
25. In the map field, add the following values: the key field should be **City,** and the value should be the **city** value in the **dynamic value**:



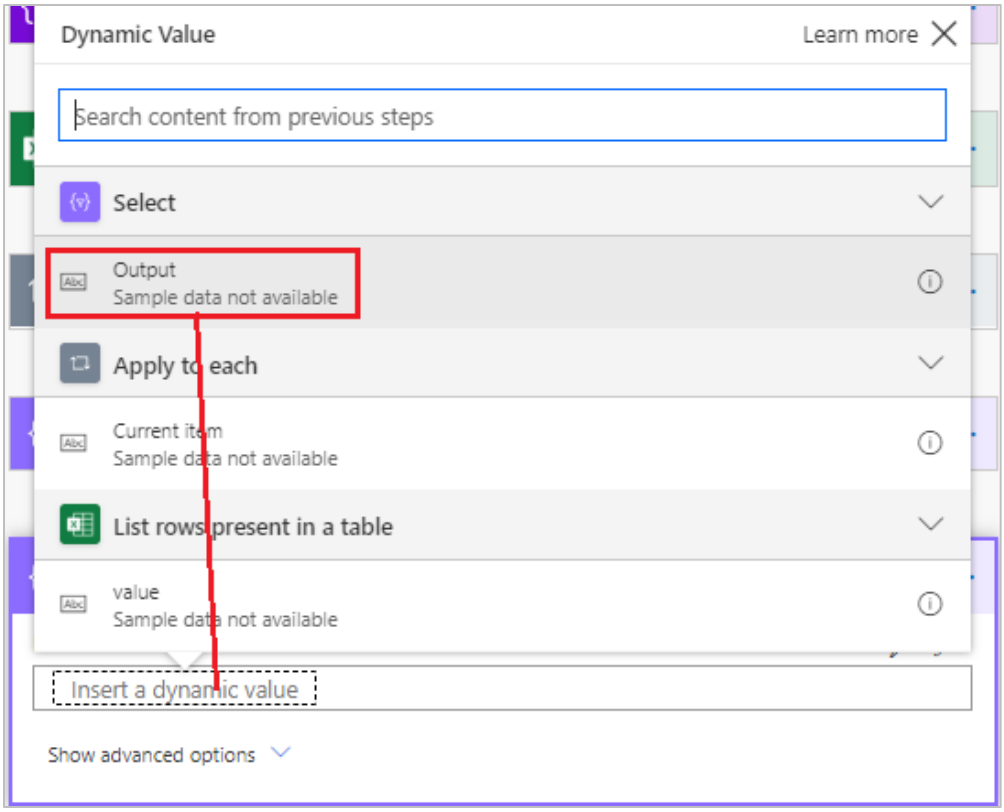
26. Add another map field for **Capacity**:



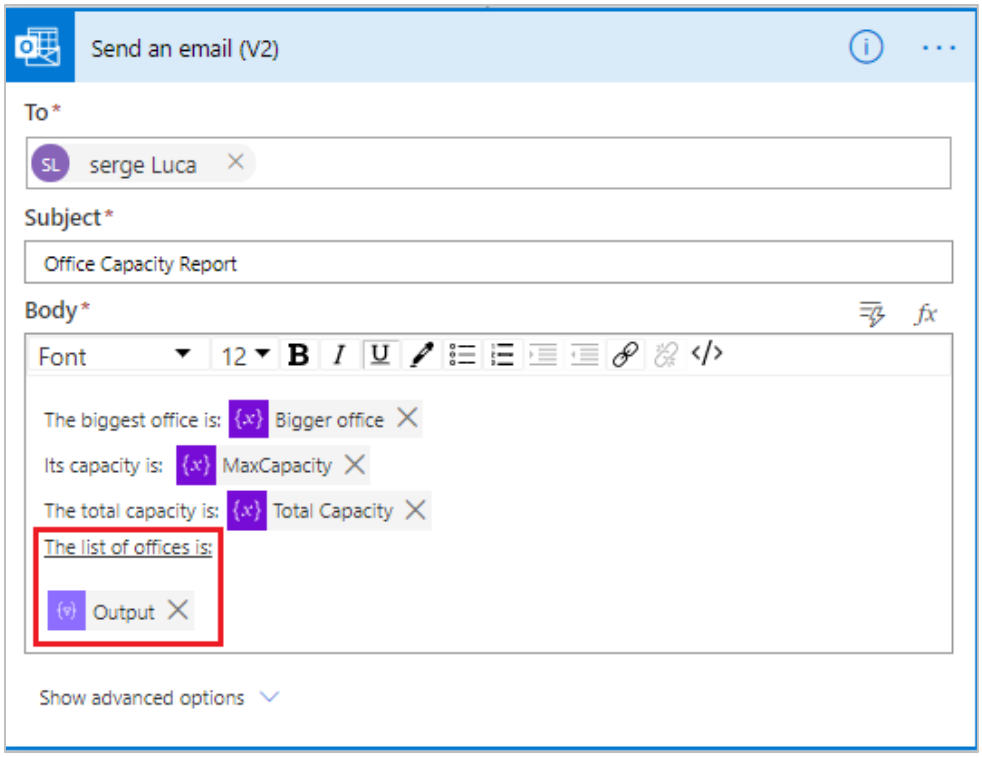
27. Just **after the Select action,** add a **Data Operations - Create HTML table** action**:**



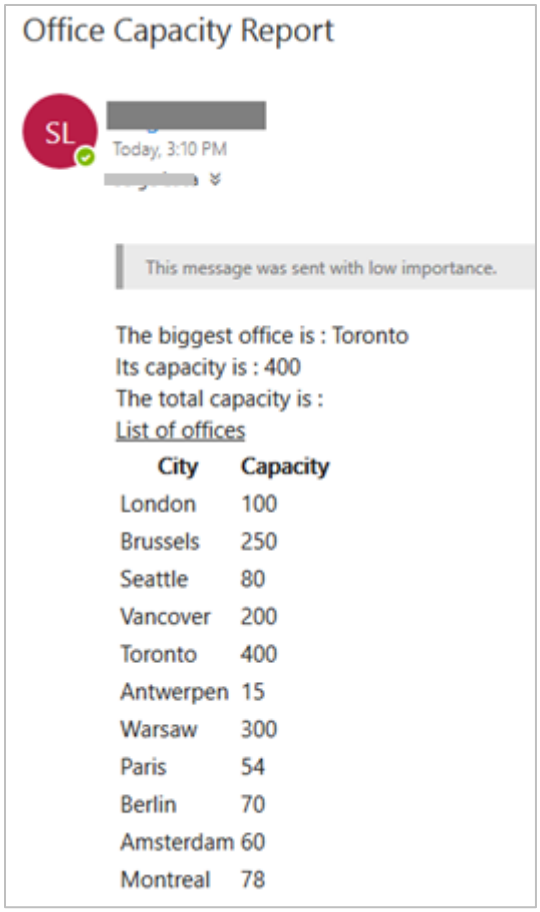
28. Move the cursor to the From field to show the **dynamic value** panel and click the **Output** value of the **Select** action:



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



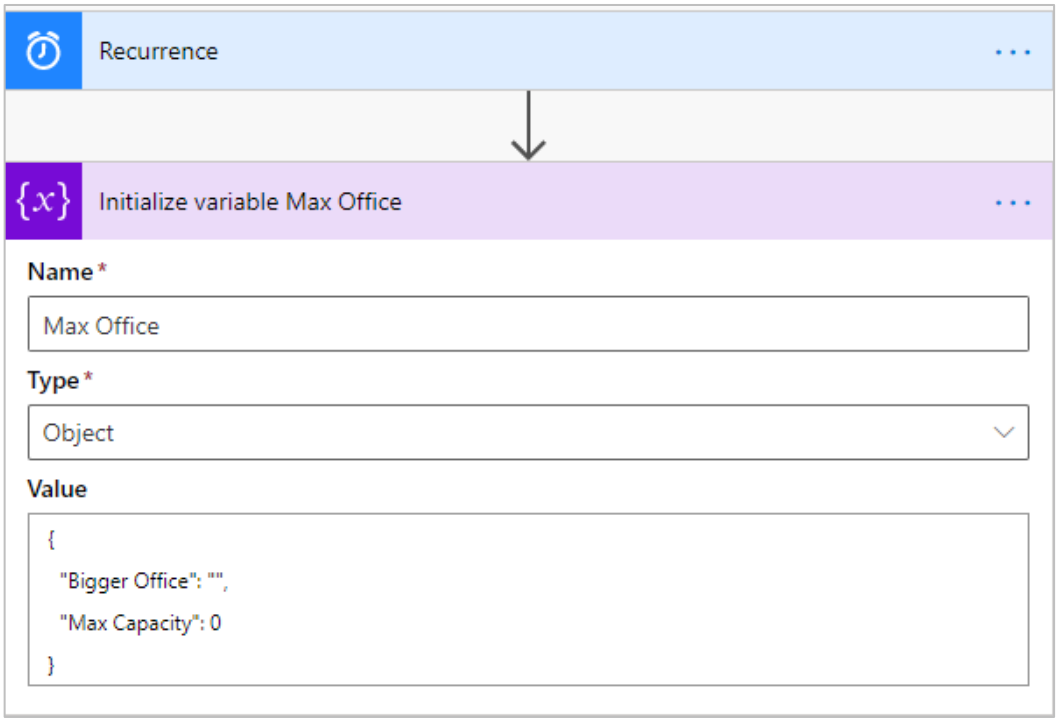
30. Test your Flow and check your e-mail:



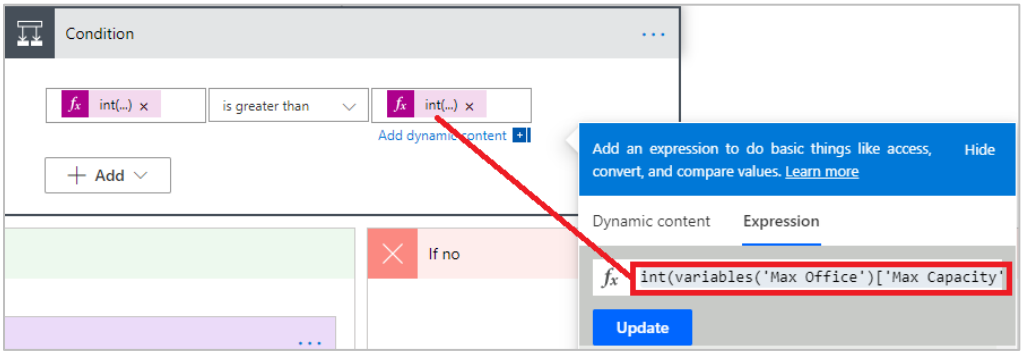
Optional exercise if time permits use an object instead of dedicated variables.

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

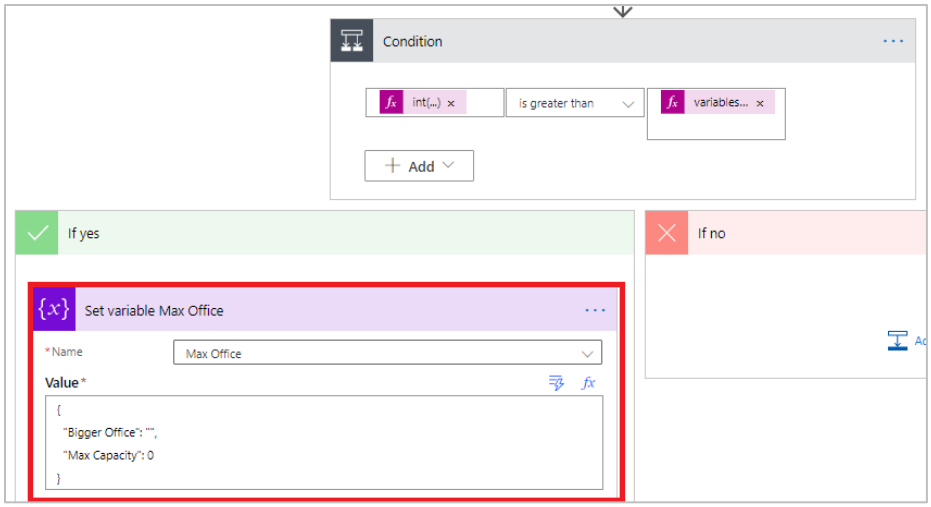
Create a variable **Max Office** as a variable of type "Object" containing the following JSON data:



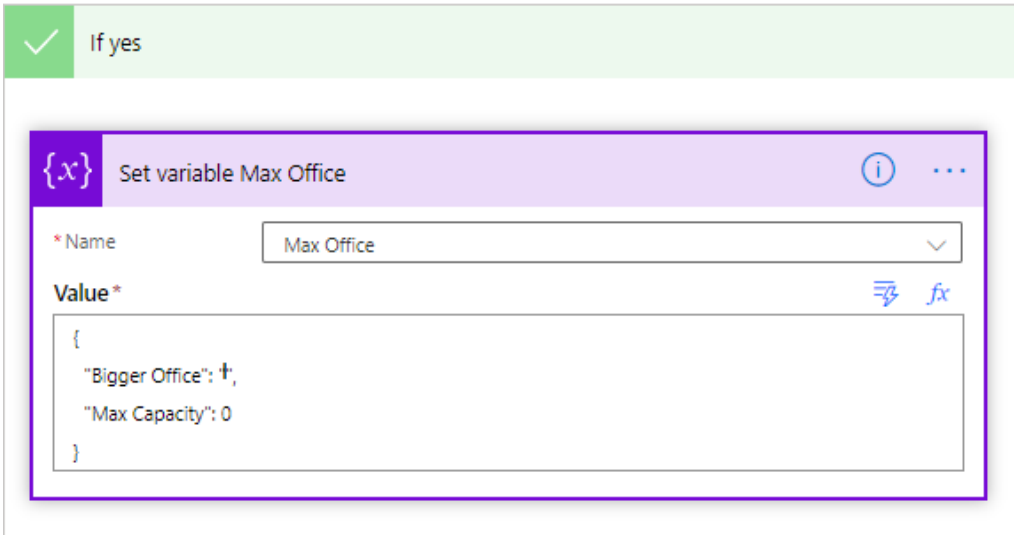
31. In the condition (In the existing **Apply to each**), use this **Max Office** variable content (do not forget to use the **int** function as well:



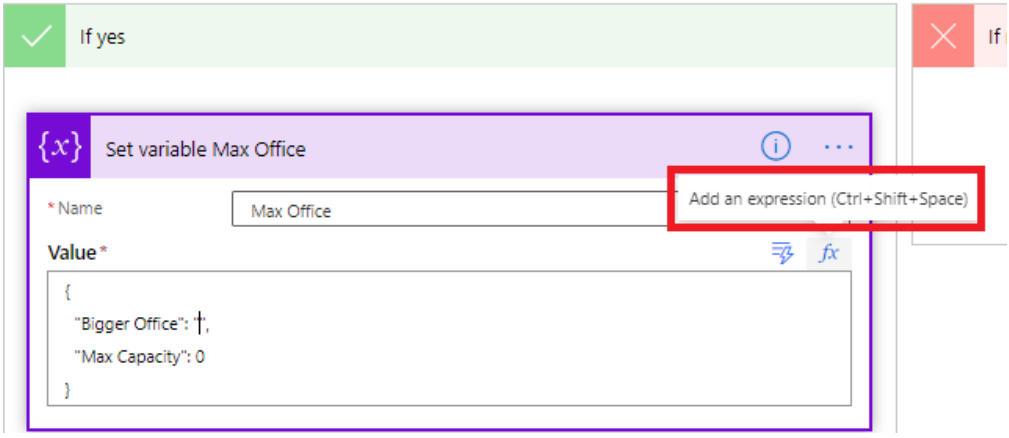
32. In the **if yes** branch condition, use a **Set variable** that will update **Max Office**:



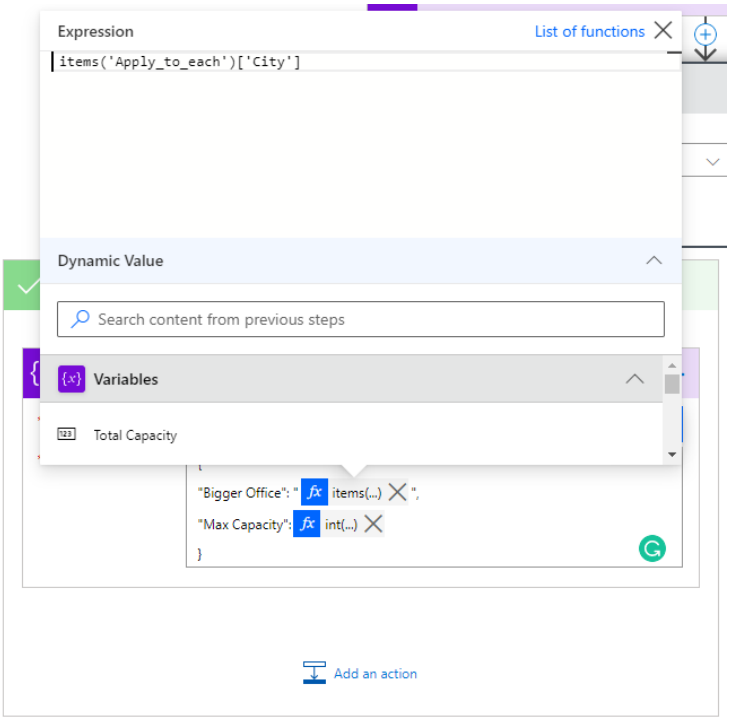
33. Now let set the Bigger Office value of the object (within the double quotes): move the cursor between the quotes:



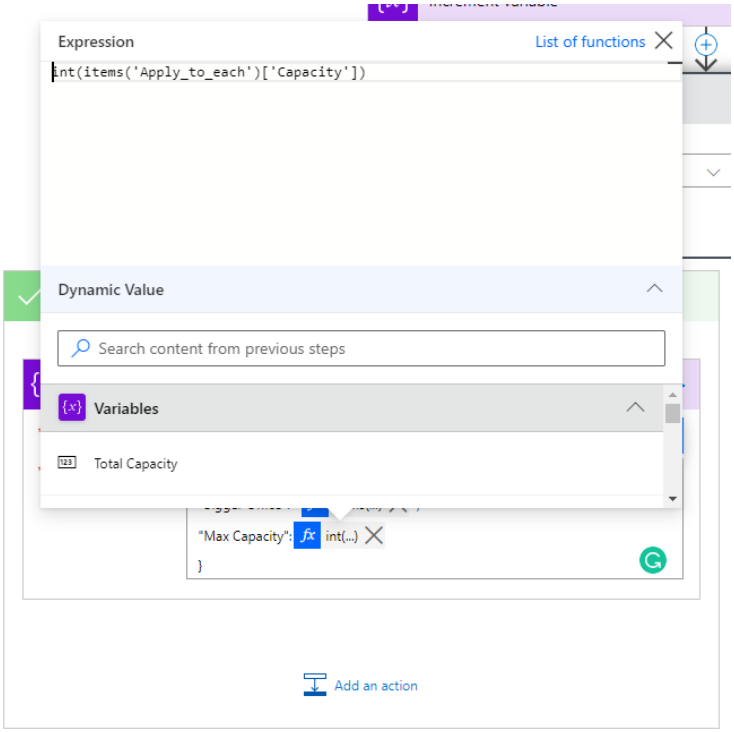
34. Click **Add An expression**:



35. Type the following expression:

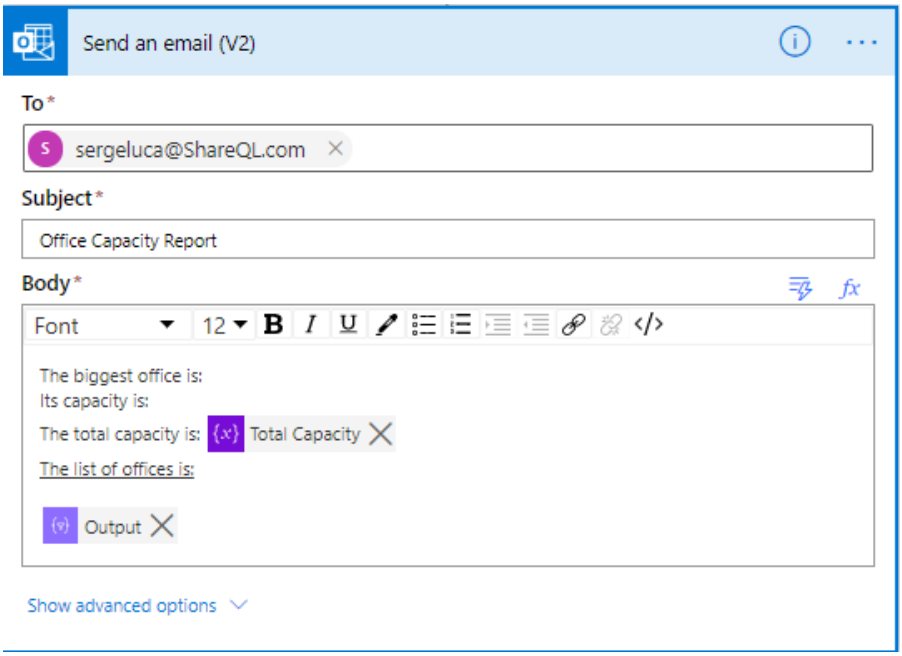


36. Do the same with the field **Max Capacity** of the object (we do not need to have double quotes around the value here):

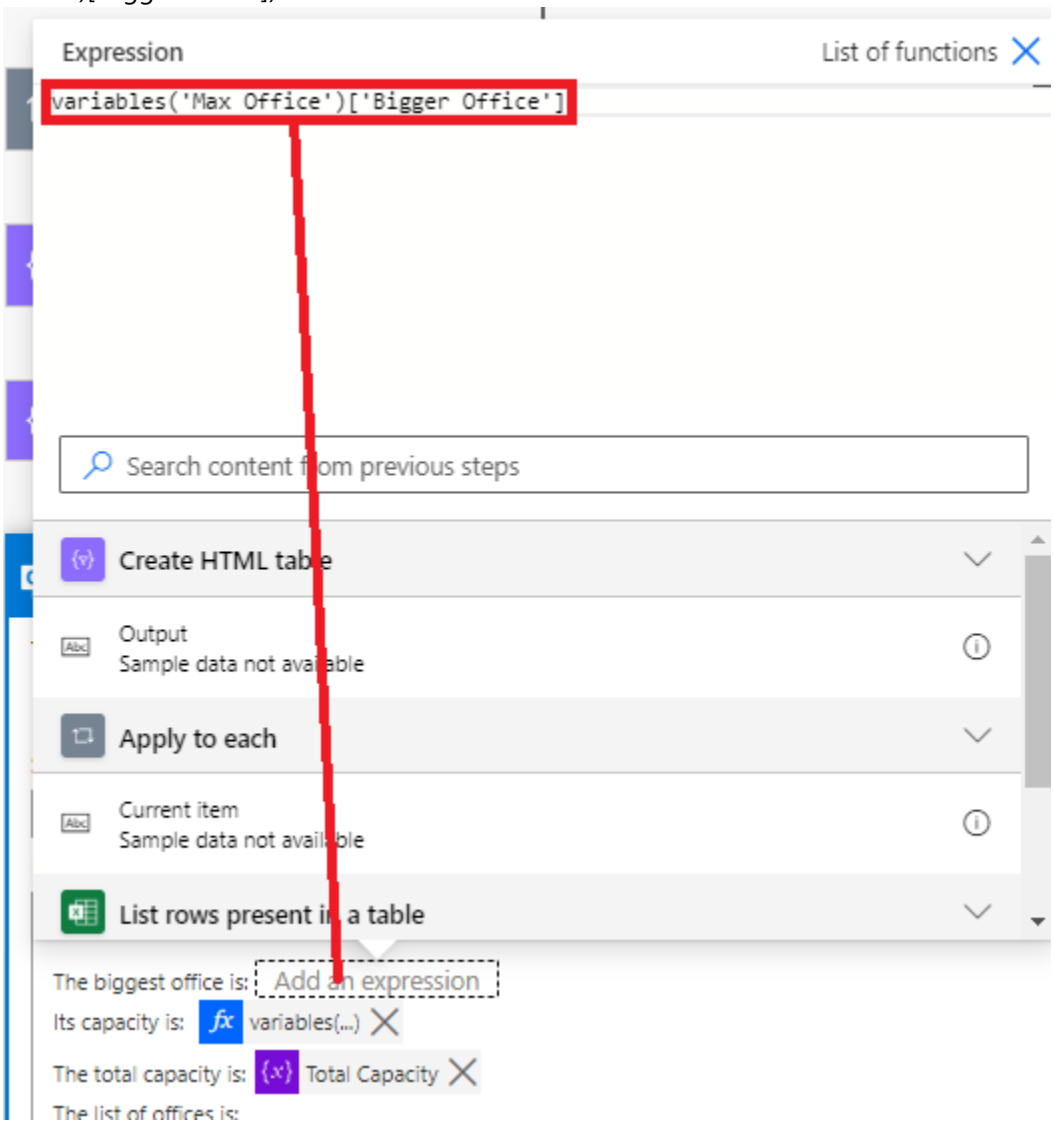


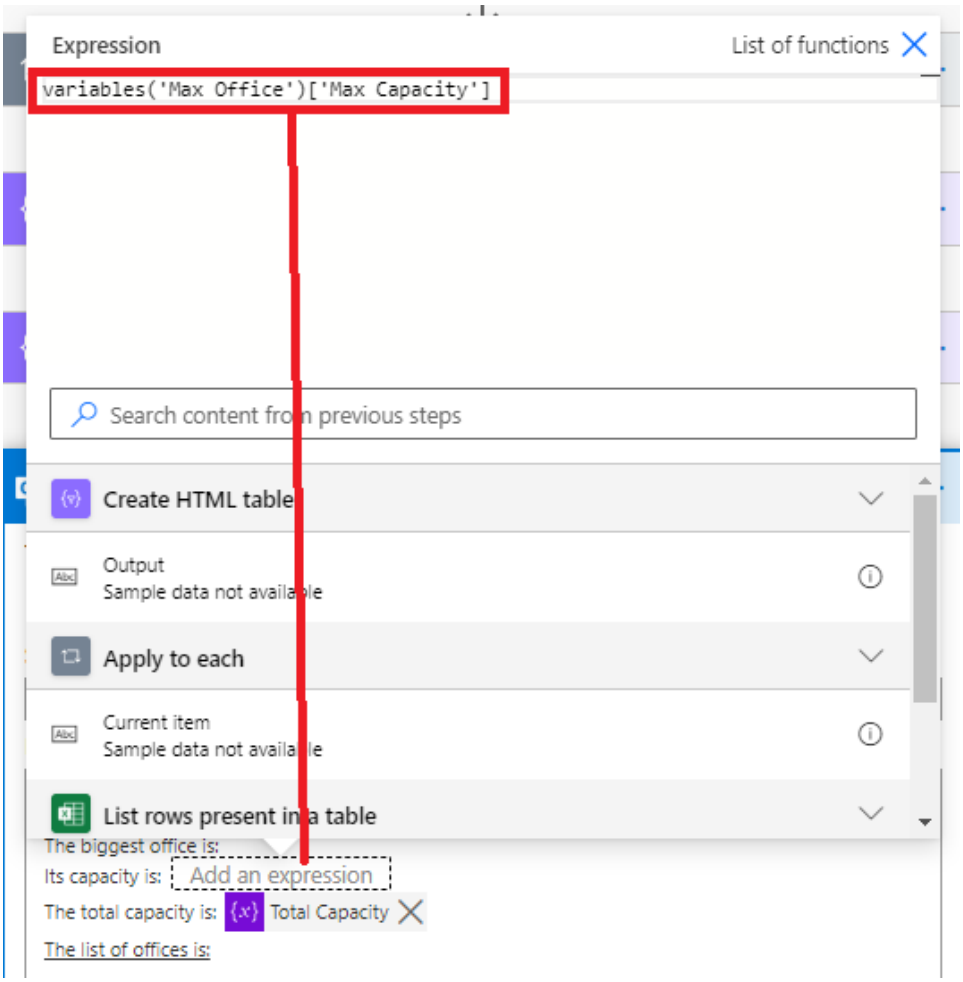
37. Remove the actions **Set Variable MaxCapacity** and **set variable Bigger office** previously defined.

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):



39. Set the value of the biggest office in the mail: add an expression like this (variables('Max Office')['Bigger Office']):

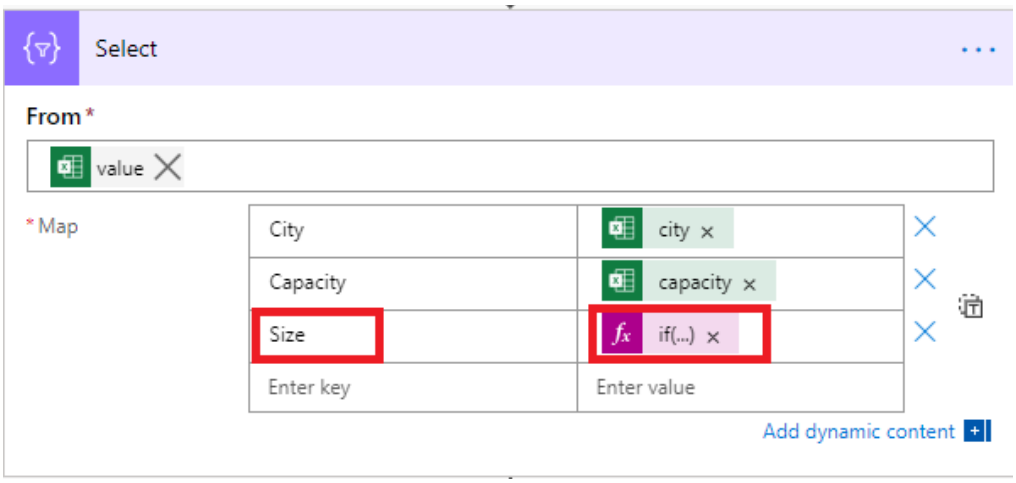




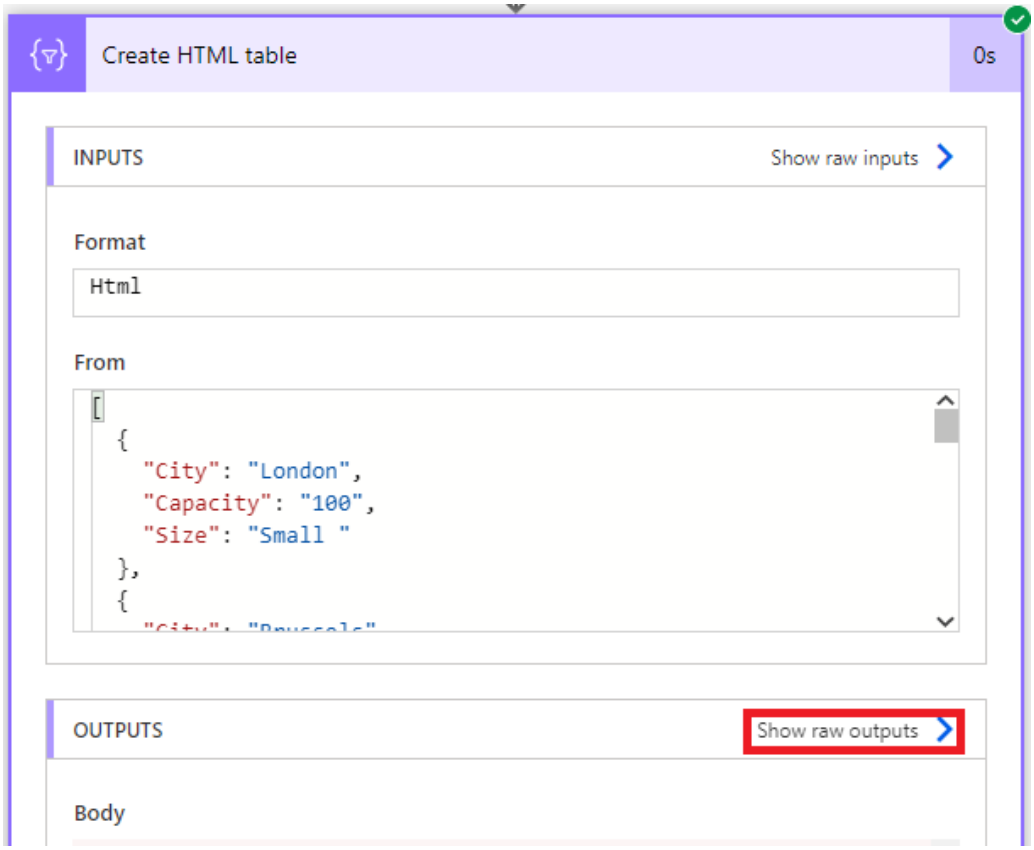
40. Test the flow :-).
41. Question: how would you update the report in order to add a new column named "Size" displaying "Small" if < 100 and "Big" if >= 100 ?

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

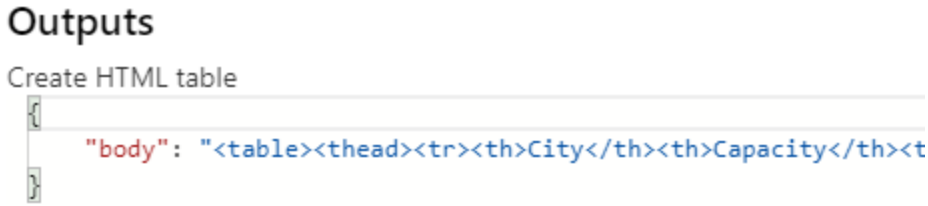
```
if ( greater(int(item()['Capacity']), 100),'Big','Small')
```



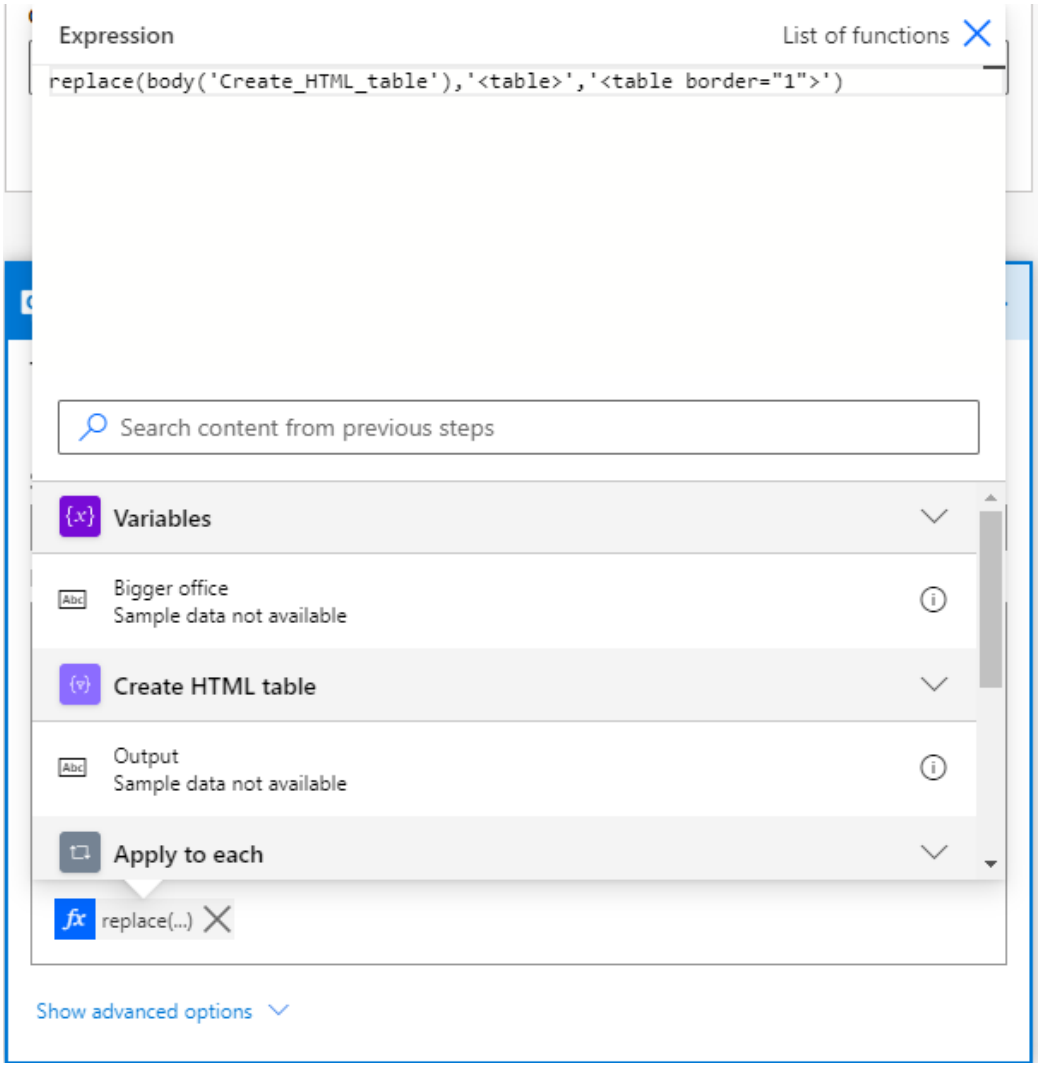
42. Question: how would you format the generated Html table to define a border?
Answer: first let us analyze the html generated code by clicking on an existing flow run and then click the option "Show raw outputs' of the Create HTML Table action as illustrated below:



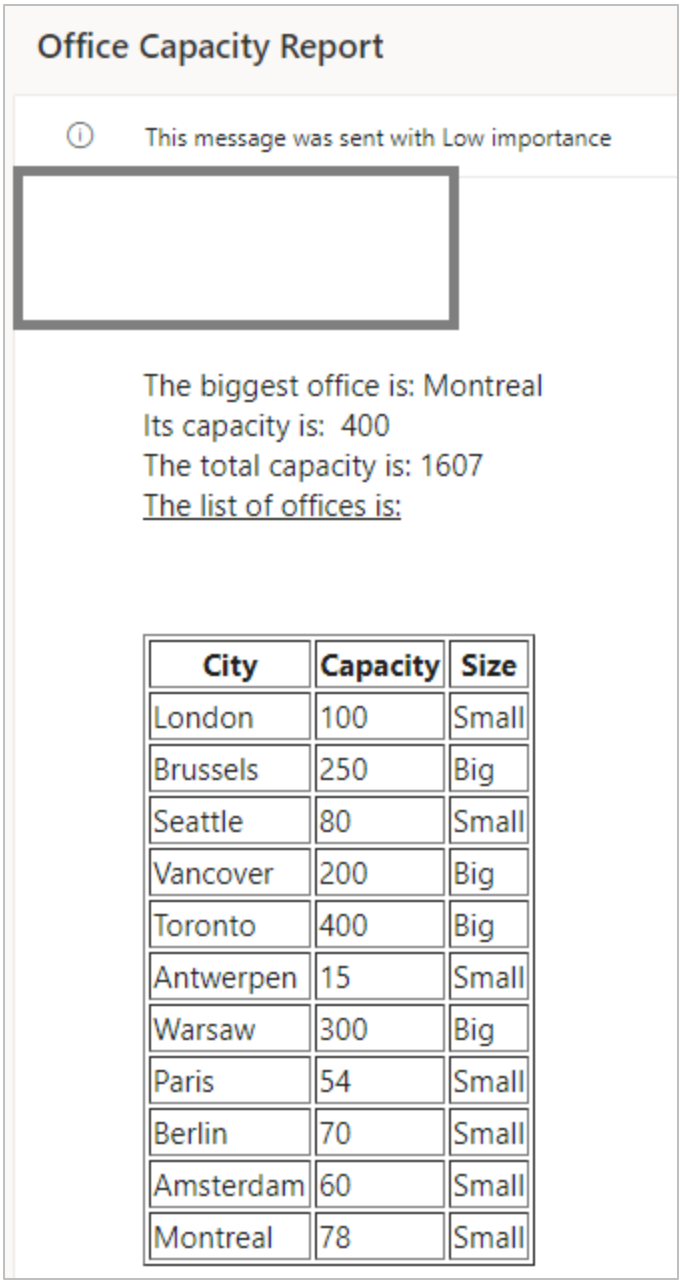
The html code looks like this:



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



The generated e-mail will look like this:



City	Capacity	Size
London	100	Small
Brussels	250	Big
Seattle	80	Small
Vancover	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