

Lab 4. Flow control, variables, expressions

Lab 4. Flow control, variables, expressions

Author: Serge Luca, aka "Doctor Flow"

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 like 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

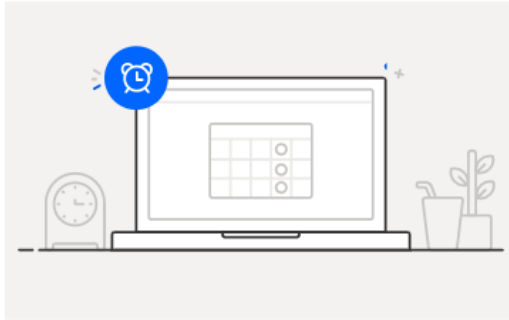
The total capacity is :

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 e-mail report.
 - a. Create a **New flow** > **Scheduled cloud flow**
 - b. Use the following screenshot to name the flow, and set the flow frequency.

Build a scheduled cloud flow



Stay on top of what's important without the effort—you choose when and how often the flow runs.

Examples:

- Automate team reminders to submit expense reports
- Auto-backup data to designated storage on a regular basis

Flow name

Office capacity

Run this flow *

Starting 2/5/23 at 10:00 AM

Repeat every 1 Month

This flow will run:

Every month

Skip

Create

Cancel

c. Click **Create**.

The following flow will be generated:

Recurrence

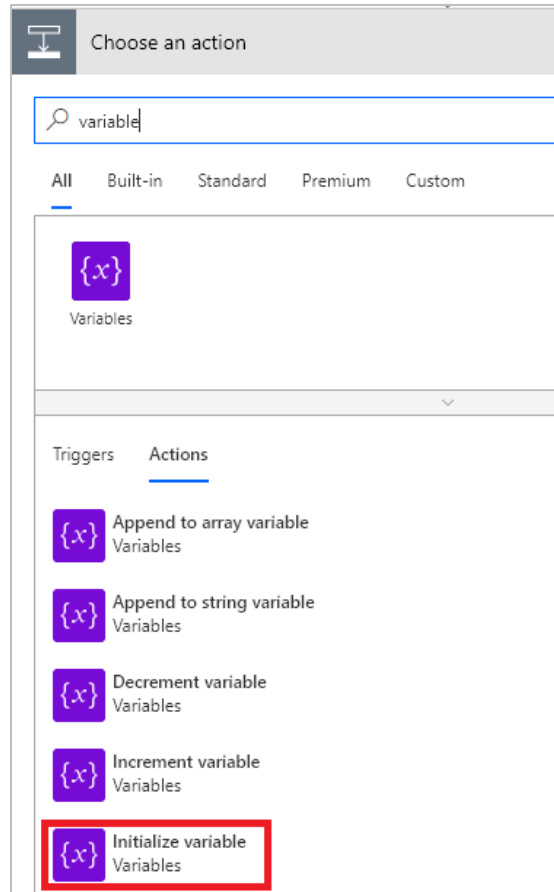
* Interval1

* FrequencyMonth

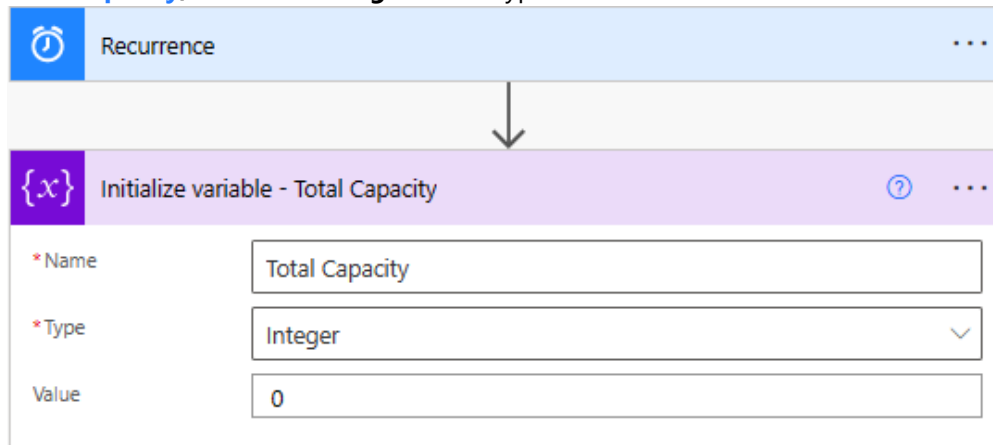
Show advanced options

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**:



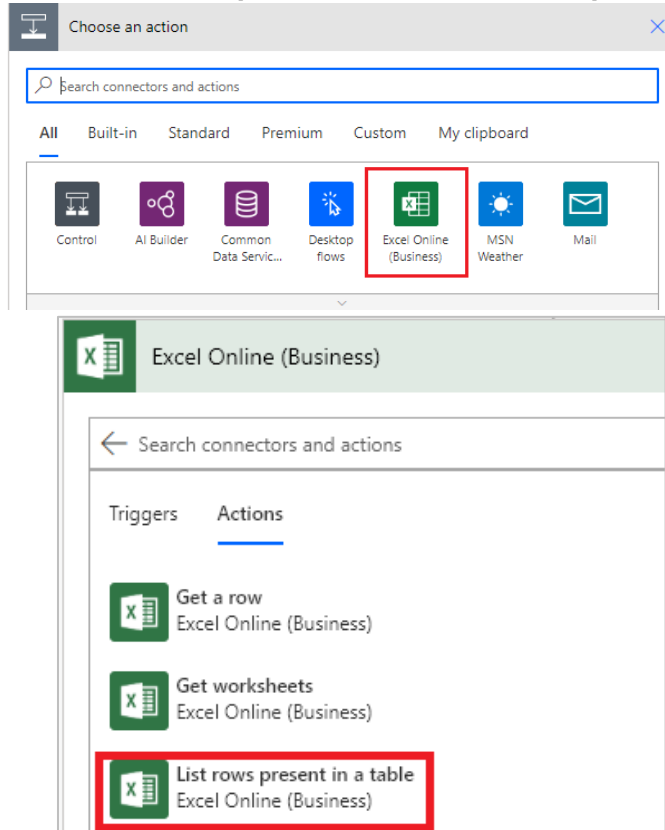
4. Click on **Save** button to **Office Capacity** flow .

Task 4.2: Extend the flow to loop through all offices

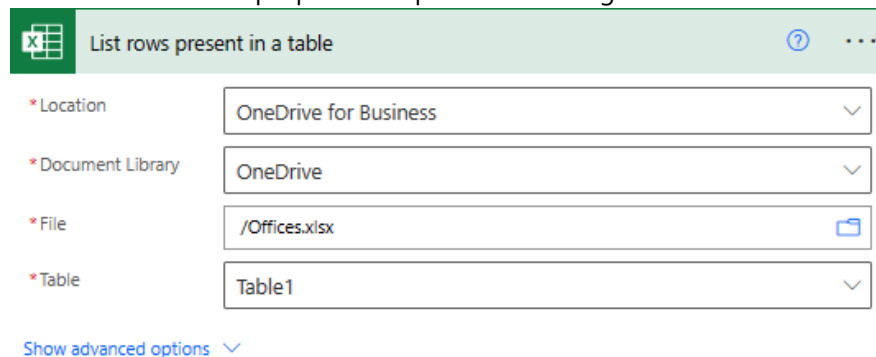
In this task, you will make it loop through all offices, retrieve their capacity, and increment the Global Capacity variable to calculate the total capacity.

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 following screenshot:



c. Click **Show Advanced option** and type **capacity DESC** in the **Order By** field:


 List rows present in a table ? ...

* Location	OneDrive for Business
* Document Library	OneDrive
* File	/Offices.xlsx
* Table	Table1
Filter Query	An ODATA filter query to restrict the entries returned.
Order By	capacity DESC
Top Count	Total number of entries to retrieve (default = all).
Skip Count	The number of entries to skip (default = 0).
Select Query	Comma-separated list of columns to retrieve (first 500 by default).
DateTime Format	Select an item


[Hide advanced options](#) ^


2. Loop through the cities


a. **Select New step > Control**

 Choose an action

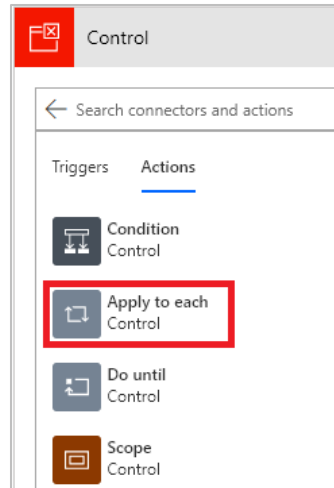
All Built-in Standard Premium


Control

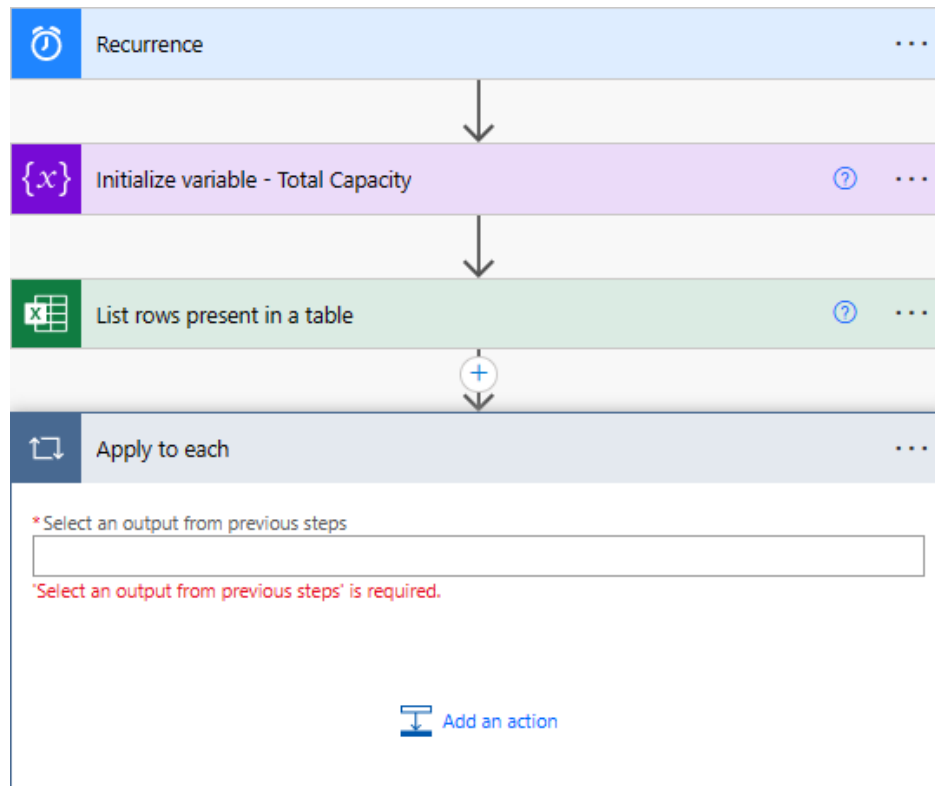

Notifications


Office 365 Outlook

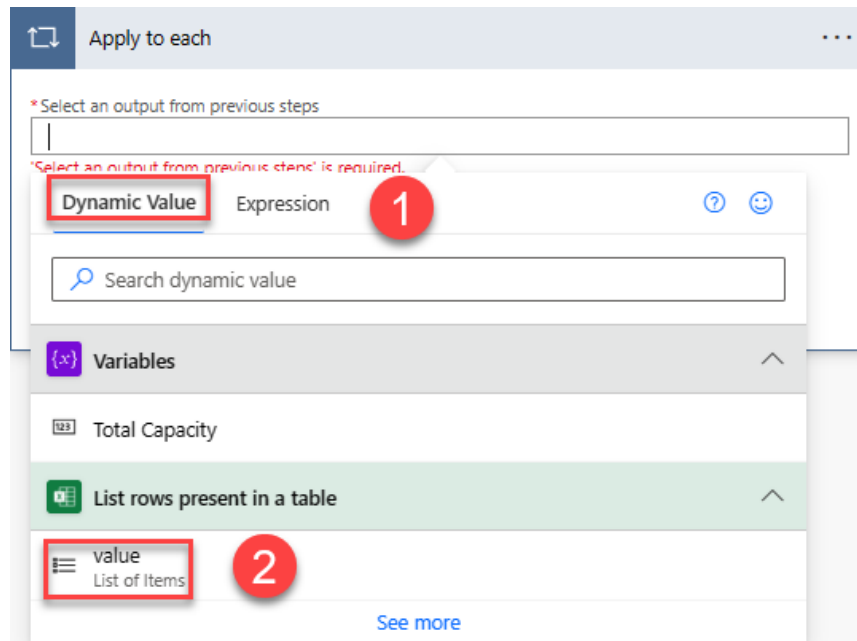
b. Click **Apply to each**:



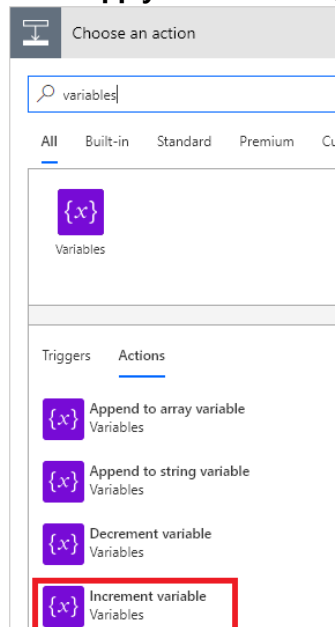
Your flow should now look like this:



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**, and rename the action:

Apply to each

*Select an output from previous steps

value

{x} Increment variable - Total Capacity

* Name Total Capacity

Value Enter a value

- c. Click inside the **Value** text box, and click the **Expression** tab to add an expression

List rows present

Dynamic Value Expression

Search dynamic value

{x} Variables

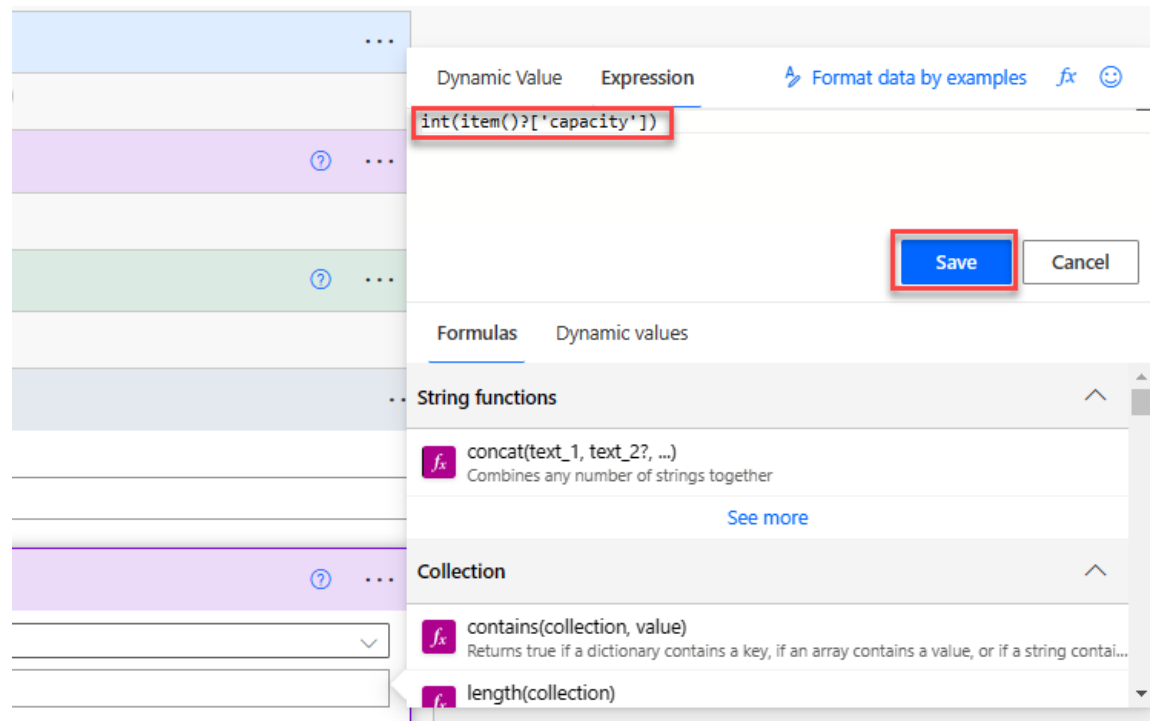
Total Capacity

Apply to each

Current item
Current item

Value Enter a value

- d. Type the following expression: and don't forget the click **Save**:
int(item()?['capacity'])

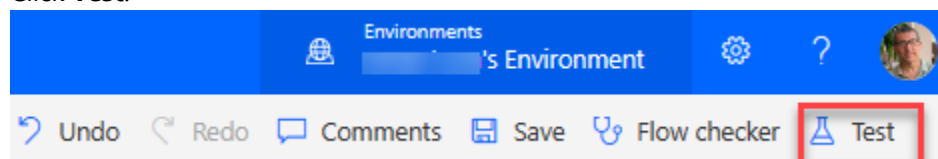


Note:

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. The question mark **?** makes your code more robust by avoiding it crashing if there is no such field (here 'capacity') in the record. The information coming from Excel is a string, and we need to transfer it as a number to increment it to the variable **Total capacity**. To transform a string to an integer (), we use the **int()** function.

There are many other expressions available in Power Automate, and we encourage you to read the documentation related to expressions after doing the labs. You can start from the following web page, <https://powerautomate.microsoft.com/en-us/blog/use-expressions-in-actions/>. The list (reference) of all functions can be found here: <https://learn.microsoft.com/en-us/azure/logic-apps/workflow-definition-language-functions-reference>

5. To test the flow, without waiting one month before it 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 **Manually**:

Test Flow

☒ Manually

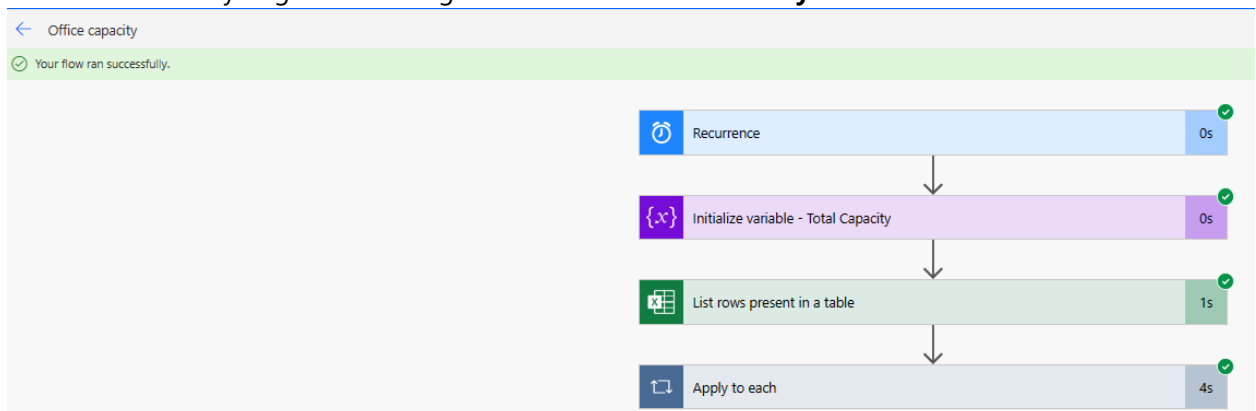
☐ Automatically

Test

Cancel

c. Click on **Run Flow**, and **Done**.

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



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

- e. Click **Increment variable – Total Capacity** 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 of 11 [Next failed >](#) [Next >](#)

 Increment variable - Total Capacity 0s

INPUTS [Show raw inputs >](#)

Name

Total Capacity

Increment By

78

OUTPUTS [Show raw 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:

- **Biggest office** (type string)
- **Biggest capacity** (type integer)

The screenshot shows three sequential steps in a workflow editor, each with a purple header bar and a red border. The first step is 'Initialize variable - Total Capacity'. The second step is 'Initialize variable - Biggest Office', with fields for Name (Biggest Office), Type (String), and Value (Enter initial value). The third step is 'Initialize variable - Biggest Capacity', with fields for Name (Biggest Capacity), Type (Integer), and Value (0). Arrows indicate the flow from the first step to the second, and from the second to the third.

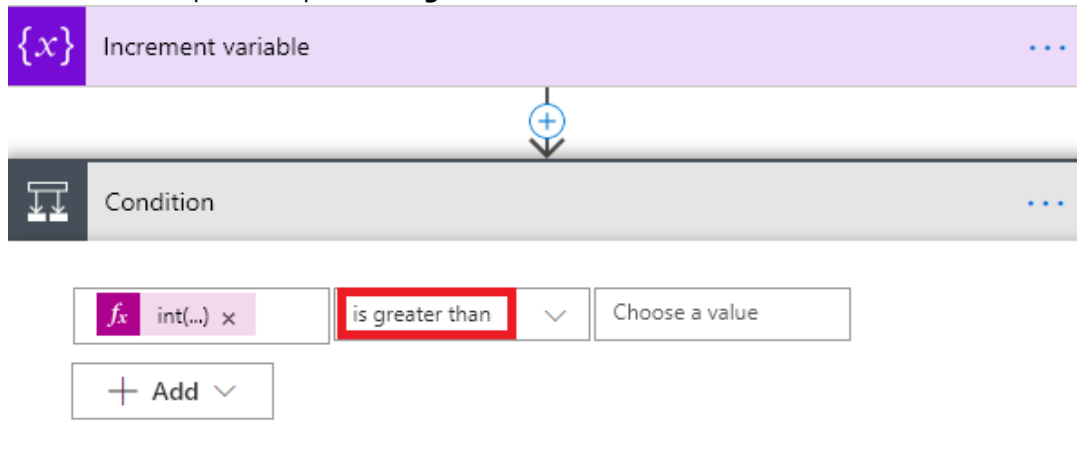
7. **Add a Condition** (from) the Control connector) in the **Apply to each** action:

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

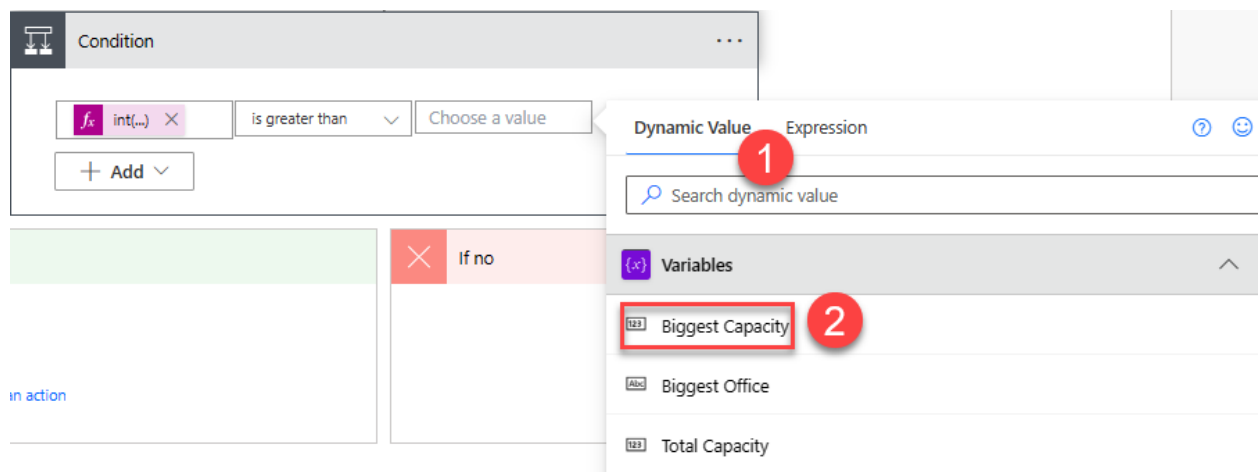
The screenshot shows the 'Condition' editor interface. On the left, there is a 'Choose a value' button and an 'Add' button. The main area has tabs for 'Dynamic Value' and 'Expression'. The 'Expression' tab is selected, and the text 'int(item()?['Capacity'])' is entered. At the bottom right, there are 'Save' and 'Cancel' buttons.

9. Click **Save**.

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

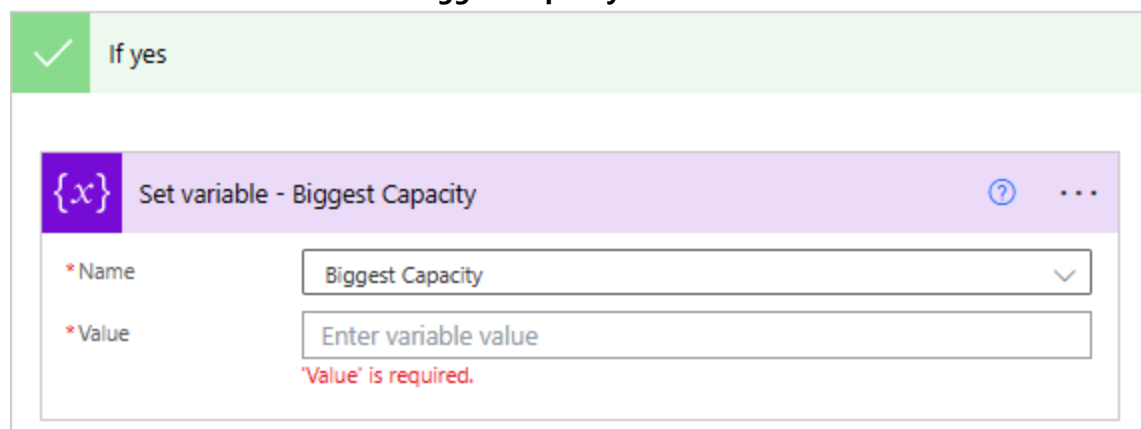


11. In the **Choose a value** textbox, select the **Dynamic value – Variables – Biggest Capacity** :

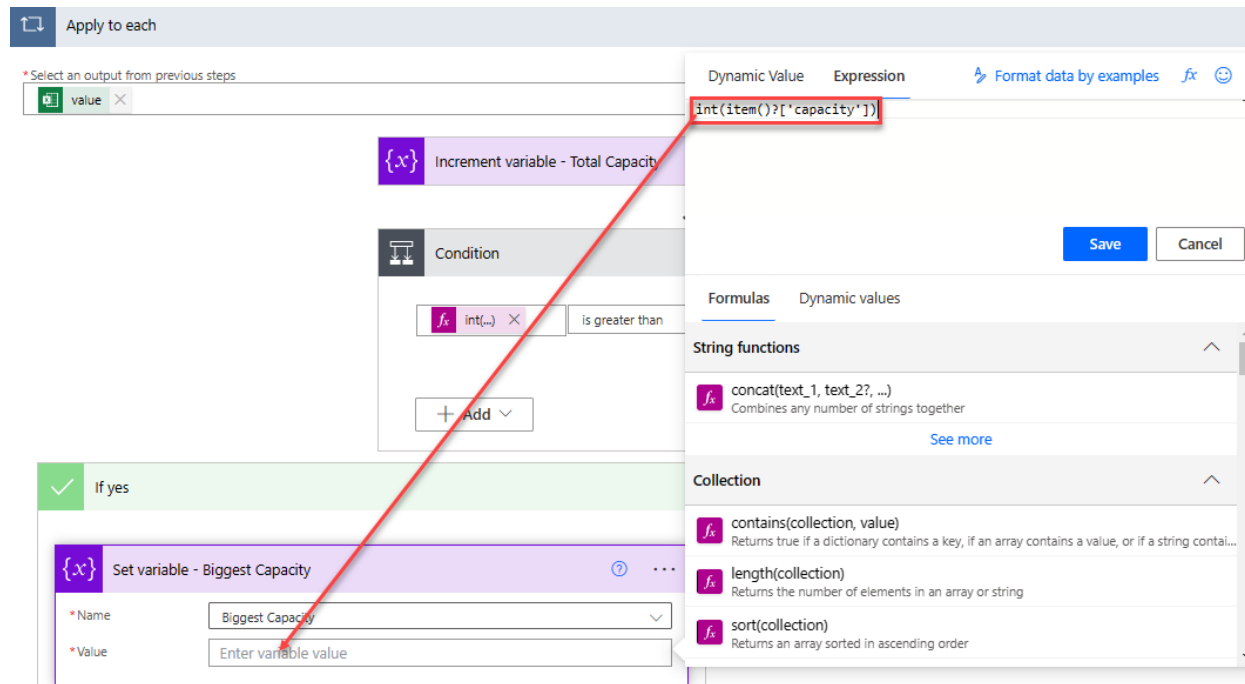


12. Now, in the left **If yes** branch, add a new action **Variables – Set variable** for our **MaxCapacity** variable.

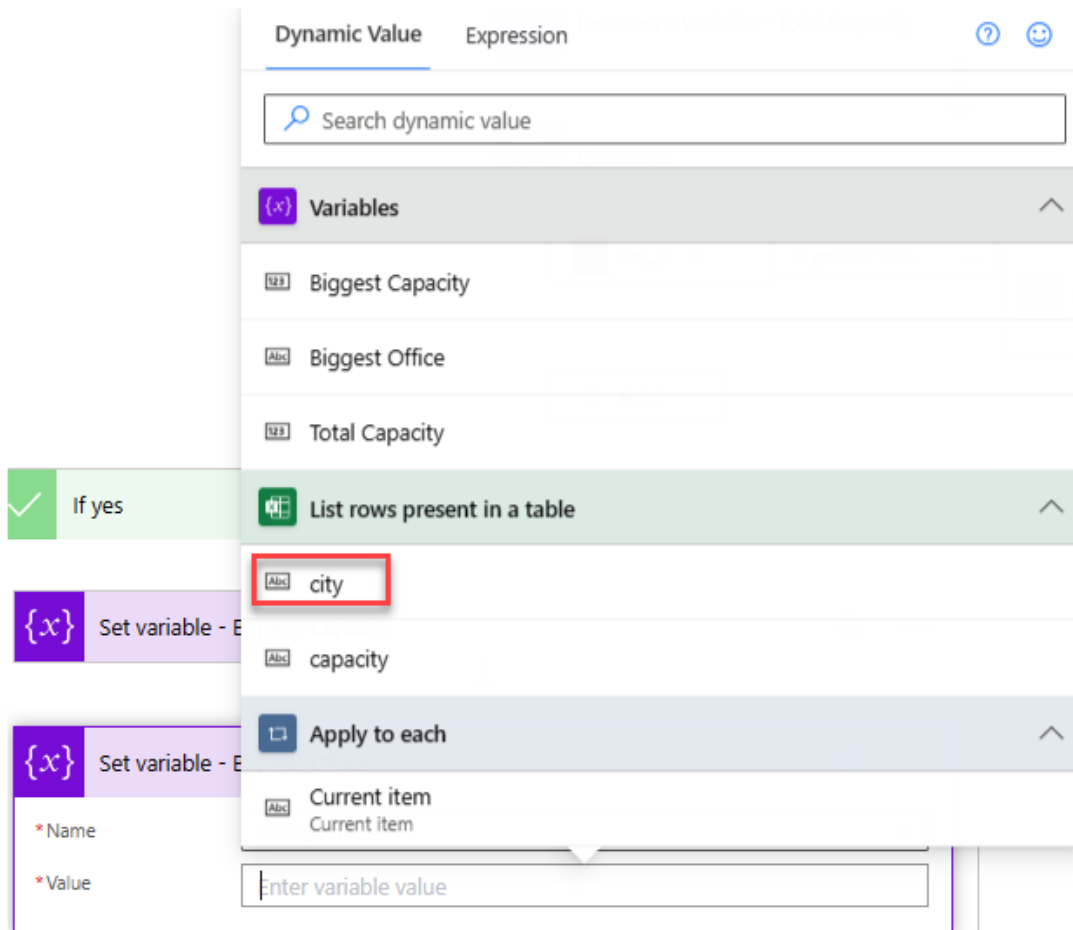
13. Rename the action **Set variable – Biggest Capacity**:



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

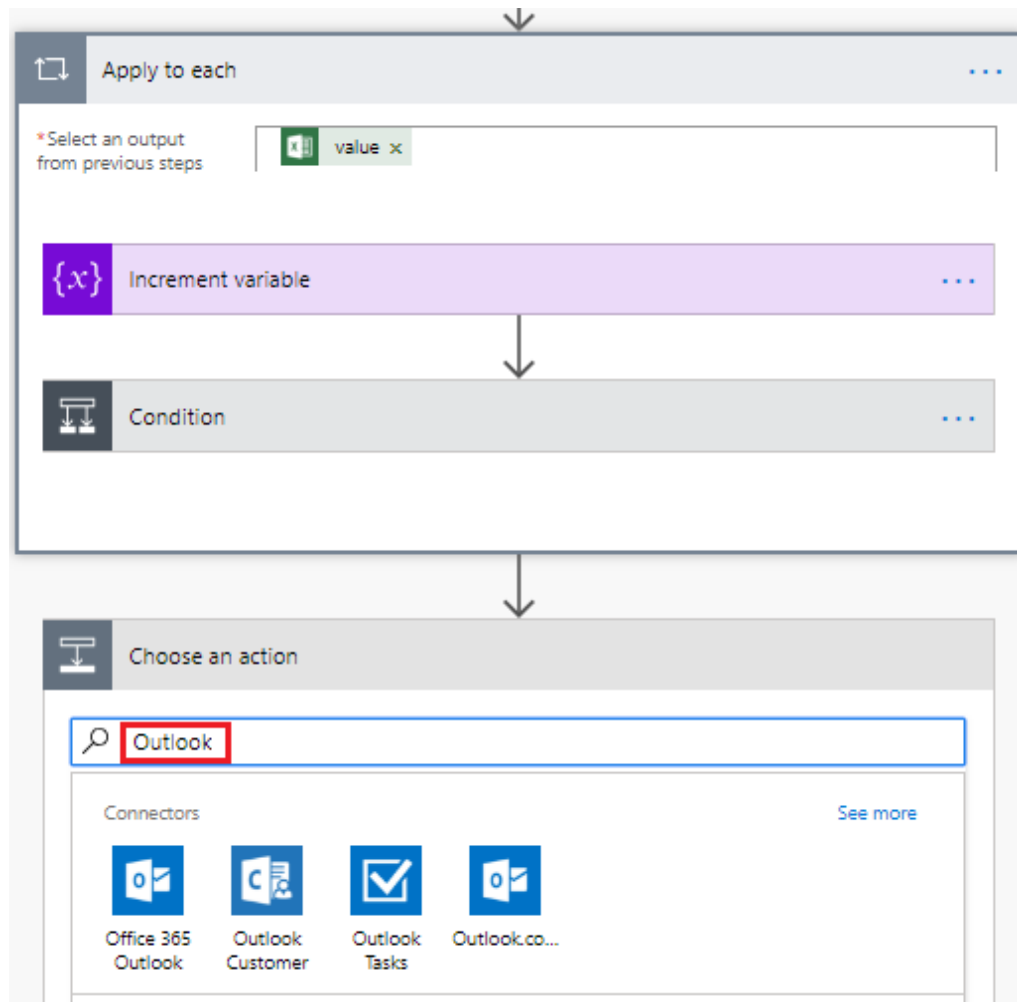


15. In the same left branch of the condition, add another **set variable action** and select the variable **Bigger Office** and assign it a value of **city**. Click on the Dynamic value button to retrieve city:

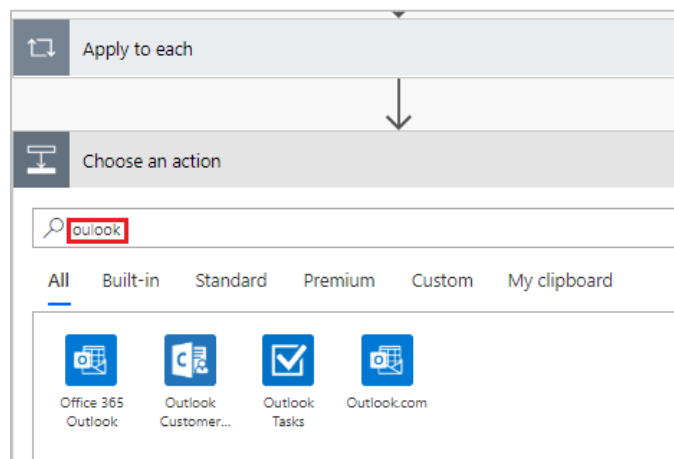


Or, as an alternative, you can create an expression with **item()?['city']**

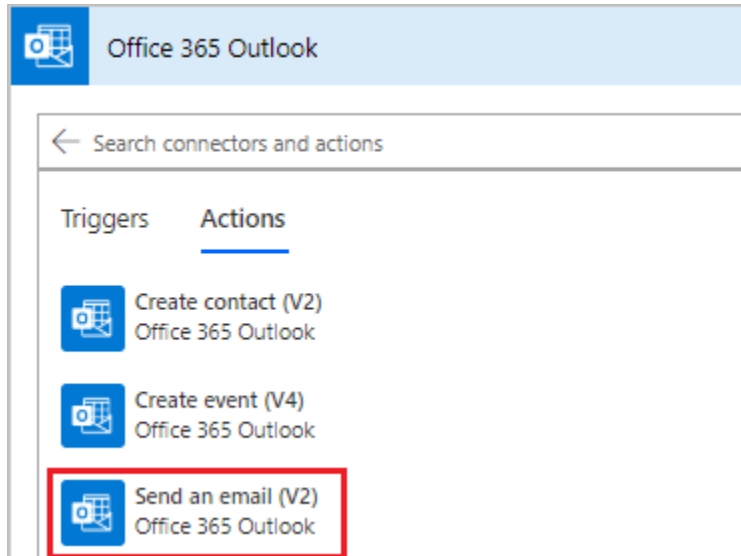
16. Save and test the flow to determine which city has the bigger capacity (Toronto in our case). You can debug the flow or add a notification (or send an e-mail to yourself).
17. Next, let's send an e-mail by adding an **Outlook 365 Outlook - Send an e-mail (v2)** action **after the Apply to each:**
 - 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 e-mail (V2)**:



- d. Fill-in the Send an e-mail 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:

The screenshot shows the configuration form for the 'Send an email (V2)' action. The form has three main sections: 'To*', 'Subject*', and 'Body*'. The 'To*' field contains the text '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, there is a 'Show advanced options' link with a downward arrow.

- 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:*

Send an email (V2)

To*
SL

Subject*
Office Capacity Report

Body*

Font 12 **B** *I* U [List Icons] [Link Icon] [Image Icon] [Code Icon]

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

Show advanced options

vi. Click Dynamic value and select the variable Biggest office:

Dynamic Value Expression

Search dynamic value

Variables

- Biggest Office**

Apply to each

- Current item
Current item

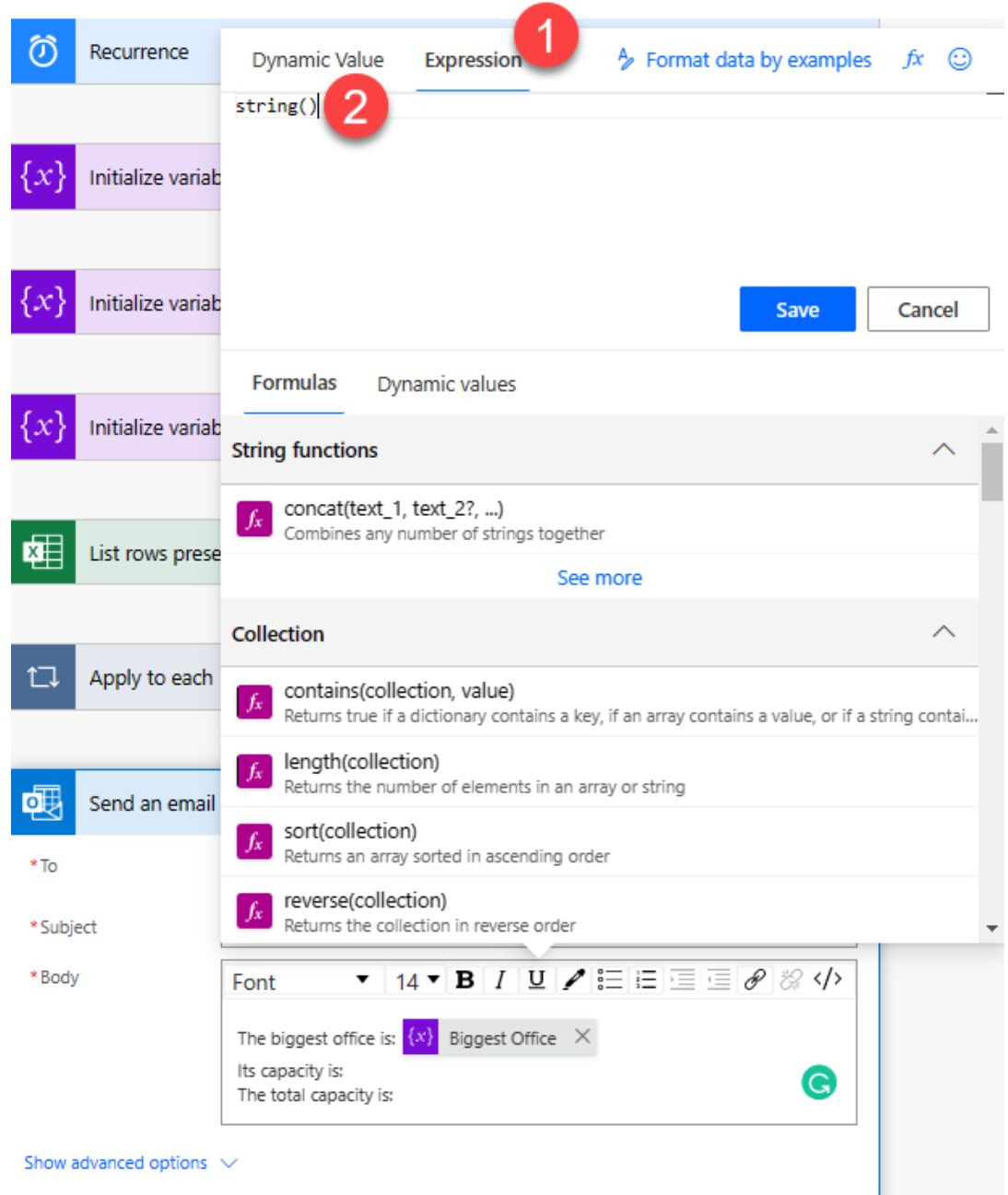
List rows present in a table

- city
- capacity

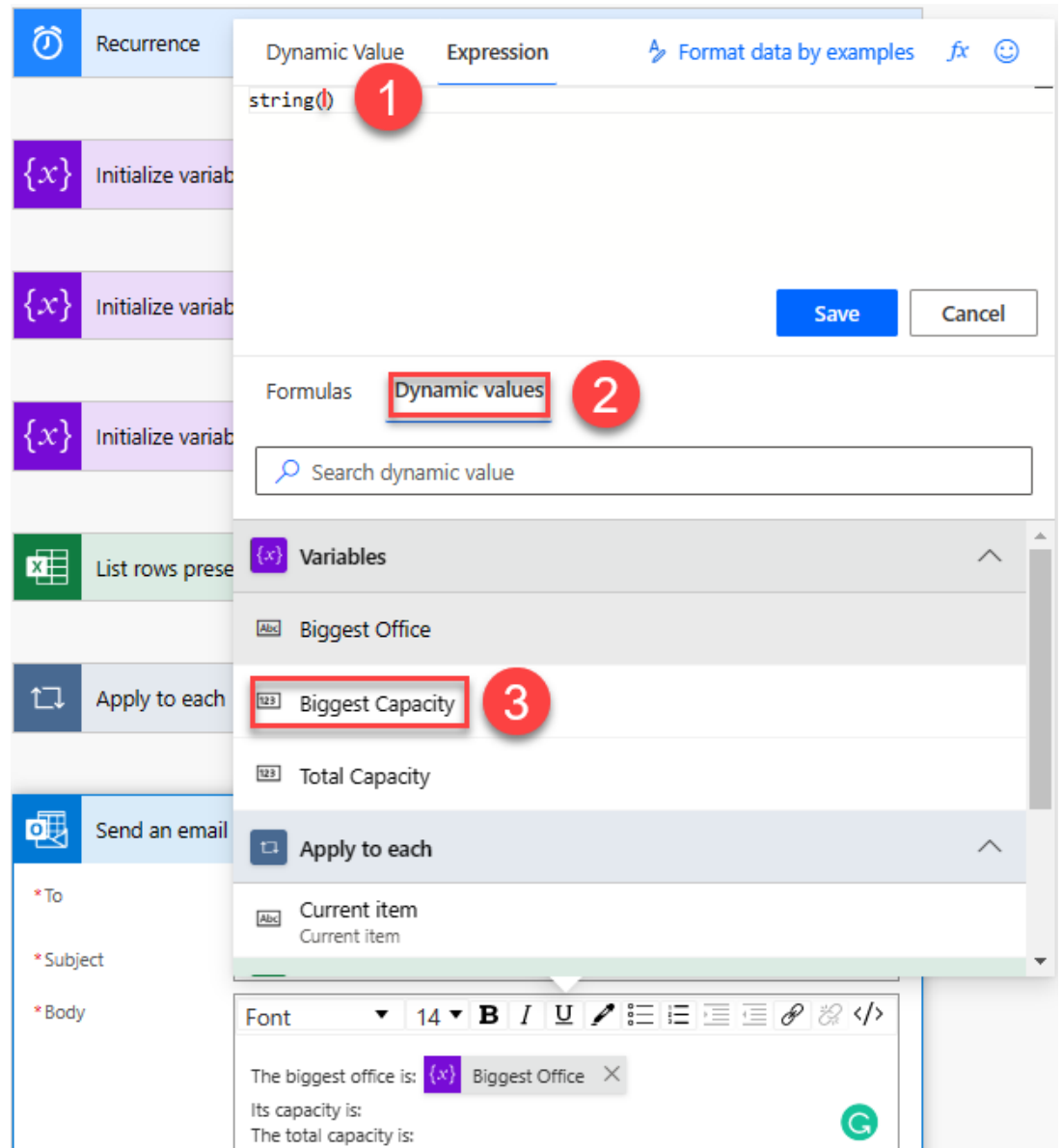
Font 14 **B** *I* U [List Icons] [Link Icon] [Image Icon] [Code Icon]

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

- vii. Unfortunately the other variables **Biggest Capacity** and **Total Capacity** are numbers, and the Send an e-mail designer expects strings; you will have to rely on a custom expression: move the cursor next to its capacity and click Expression, where you can type string



Move the cursor within the string () parentheses and click dynamic value to grab the variable Biggest Capacity (and click more if you don't find it in the list of variables):



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

Send an email (V2)

*To: [Redacted] X

*Subject: Office Capacity Report

*Body:

Font 14 B I U [Icons]

The biggest office is: {x} Biggest Office X

Its capacity is: f_x string(...) X

The total capacity is: f_x string(...) X

Show advanced options V

18. Save your flow and test it.
19. Check your e-mail; you should receive something like this:

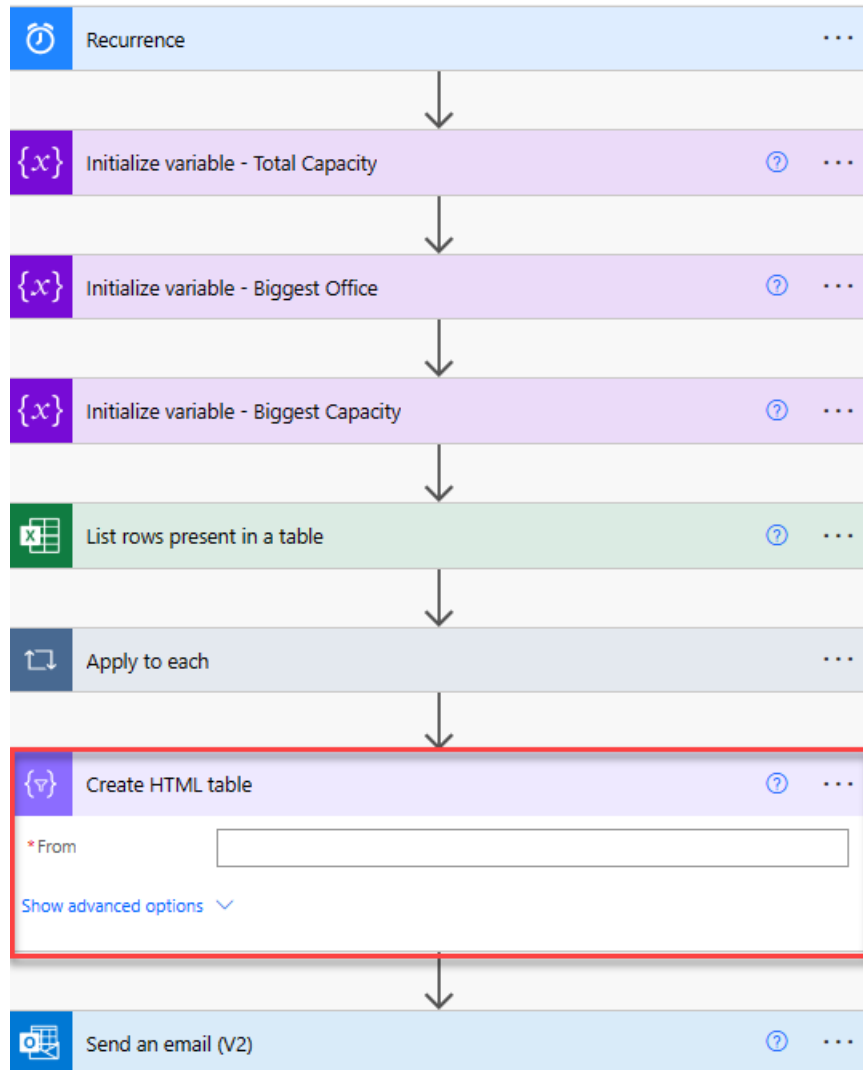
Office Capacity Report



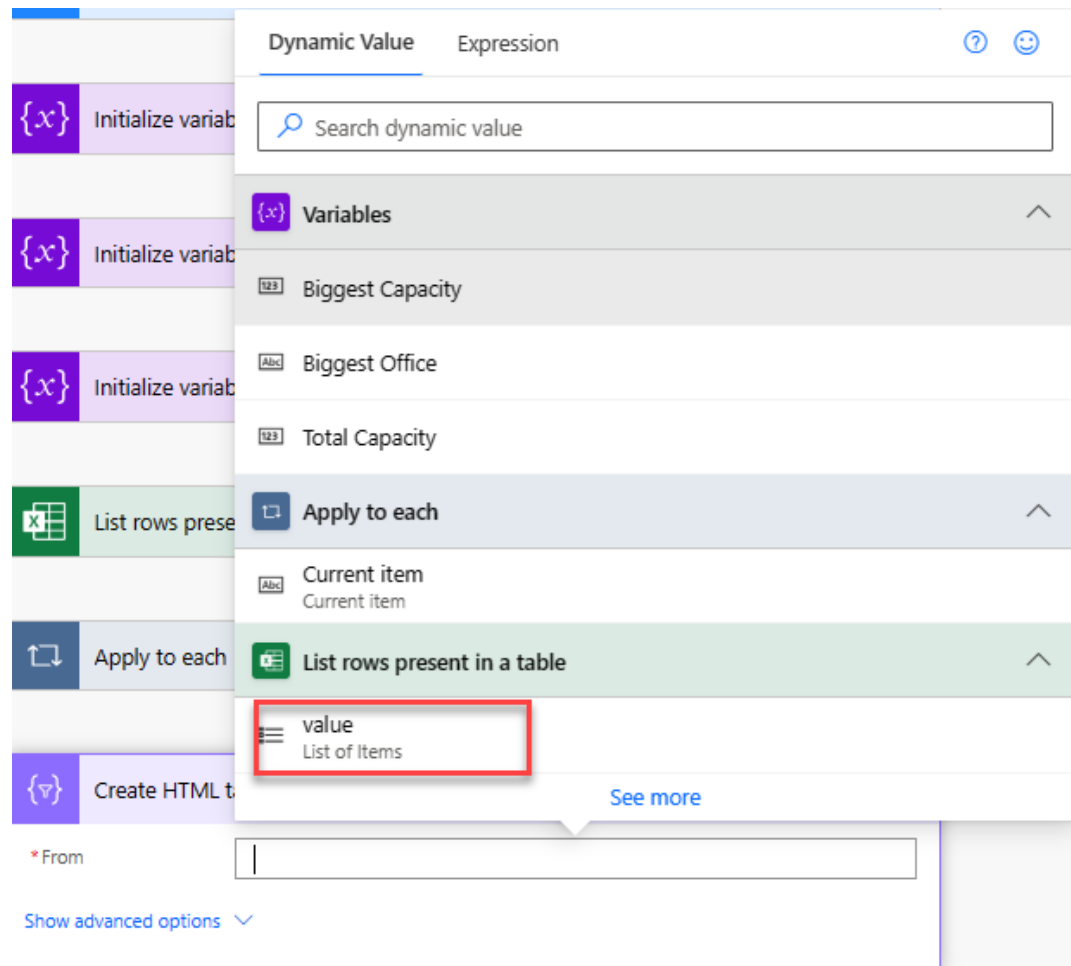
The biggest office is: Toronto
 Its capacity is:400
 The total capacity is:1607

Reply Forward

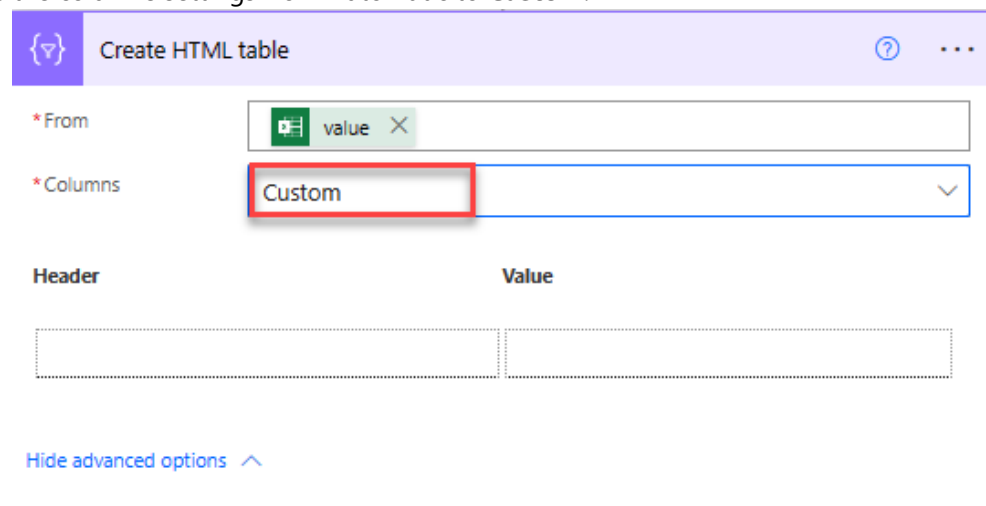
20. In the following steps, we will display the list of offices, so we will have to define a list formatting logic and create an HTML table based on this logic.
21. Let's define the list formatting logic. Before the **Send an e-mail action**, add a **Data Operations – Create HTML table** action:



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



23. Change the columns settings from Automatic to **Custom**:



24. You can now define the logic of your report : the header name (in **Header**) and the dynamic value or the expression associated with each header in **Value**.

25. In the **Header** field, type **City**, and in **Value** select **the dynamic value city** :

The screenshot shows the PowerApps interface with the 'Create HTML table' step selected. The 'Header' row is being defined, and the 'City' column is being added. The 'Dynamic Expression' pane is open, showing a list of variables and functions. Red circles 1, 2, and 3 highlight the 'City' input, the 'Dynamic Expression' pane, and the 'city' variable selection respectively.

Dynamic Expression

Search dynamic value

Variables

- Biggest Capacity
- Biggest Office
- Total Capacity

Apply to each

- Current item

List rows present in a table

- value
- body
- body/value - Item
- city

Create HTML table

* From: value

* Columns: Custom

Header

City

Hide advanced options

26. Follow the same procedure for Capacity:

{ }

Create HTML table

?

...

* From

value

×

* Columns

Custom

▼

Header

Value

City

city

×

×

Capacity

capacity

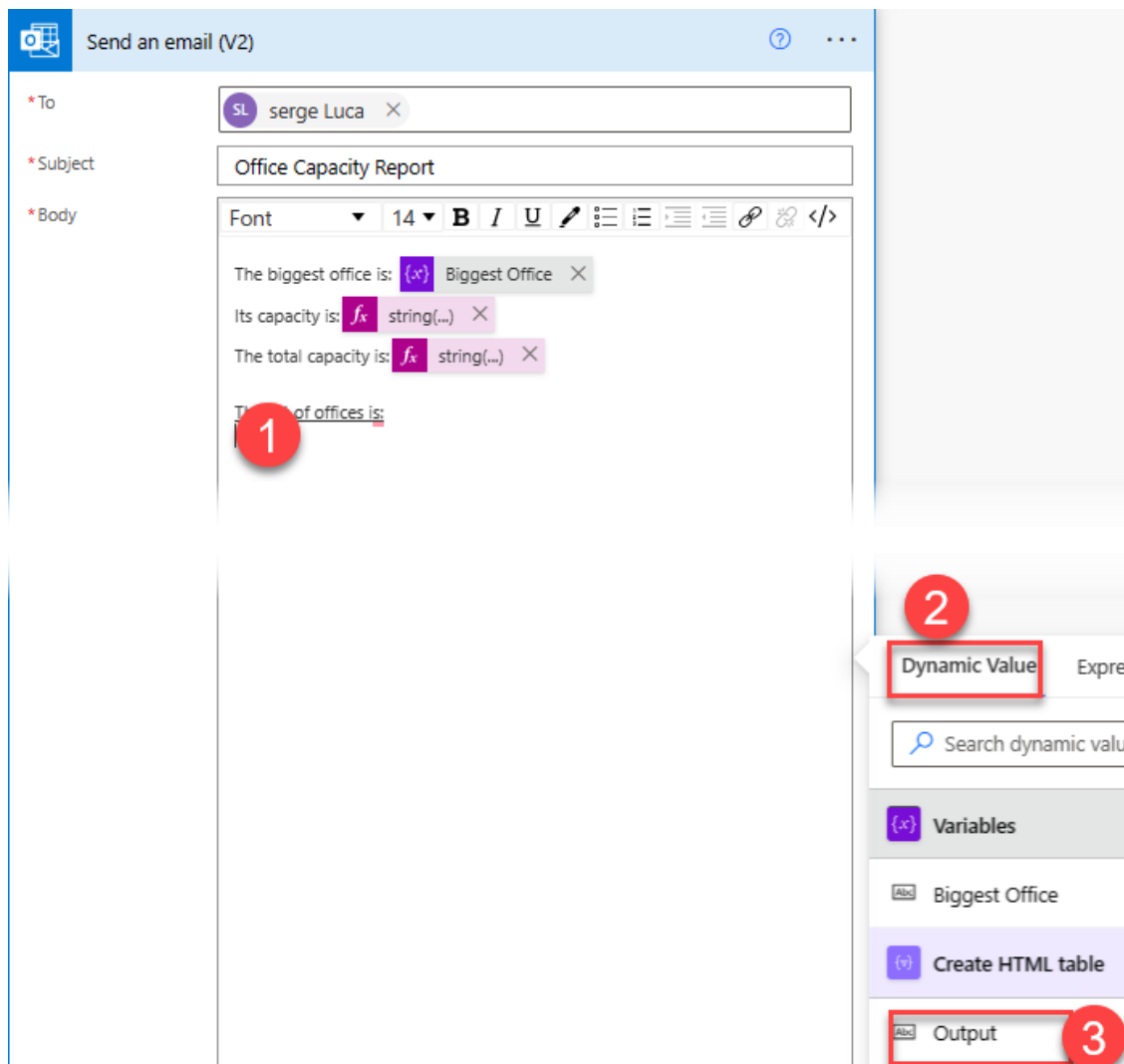
×

×

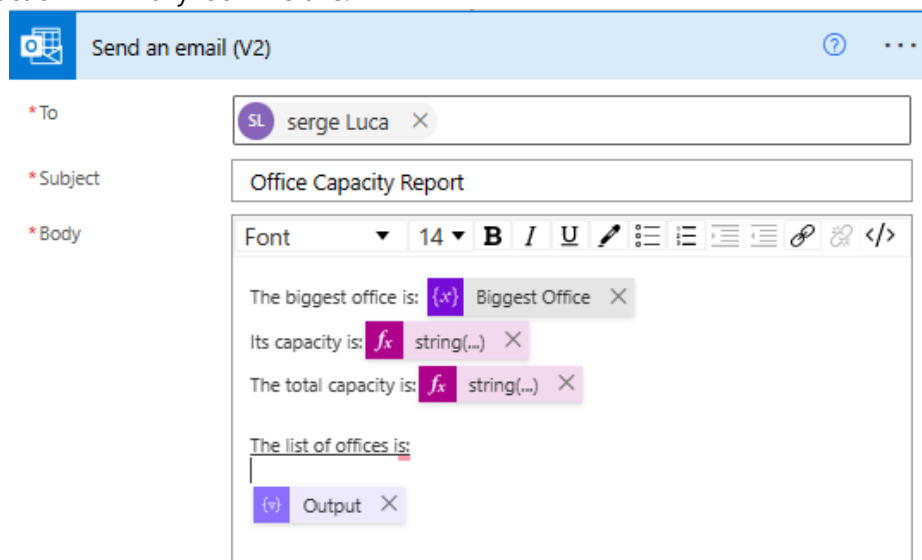
Hide advanced options

^

27. Go back to the **Send an e-mail** action and update the **Body** text box to include the Create HTML Output value: just type the Total list of offices is : and add the output of The create HTML table action:



28. The action will finally look like this:



29. Test your flow and check your e-mail:

Office Capacity Report



[Redacted]

To: [Redacted]

The biggest office is: Toronto

Its capacity is:400

The total capacity is:1607

The list of offices is:

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

The rest of the exercise will be about customizing step by step the returned table, which will eventually look like this :

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

30. Let's extend our flow by adding a new column named Size in our report: the logic is that if the quantity is smaller than 100, we would like to see "Small", otherwise we will see "Big".

Answer; you will need to use the following functions **if()** and **greater()**. The expression is :

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

The screenshot displays a workflow editor with the following steps:

- Initialize variable - Biggest Office
- Initialize variable - Biggest Capacity
- List rows present in a table
- Apply to each
- Create HTML table

The 'Create HTML table' step is expanded, showing a table structure with the following headers and values:

Header	Value
City	city
Capacity	capacity
Size	if(...)

The 'Size' header is highlighted with a red box and a red circle labeled '1'. The 'if(...)' value is also highlighted with a red box and a red circle labeled '1'.

A 'Dynamic Value' dialog box is open, showing the 'Expression' tab. The expression is: `if (greater(int(item()['Capacity']), 100), 'Big', 'Small')`. The expression is highlighted with a red box and a red circle labeled '3'. The 'Save' button is highlighted with a red circle labeled '4'.

The dialog box also shows a list of functions under the 'Collection' section:

- `contains(collection, value)`: Returns true if a dictionary contains a key, if an array contains a value, or if a string contains a substring.
- `length(collection)`: Returns the number of elements in an array or string.
- `sort(collection)`: Returns an array sorted in ascending order.
- `reverse(collection)`: Returns the collection in reverse order.

The generated mail will look like this:

The biggest office is: Toronto
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

31. Now we would like to customize the generated html code by adding a table border.
- First, let 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:

Send an email (V2)

*To: sergeluca@ShareQL.com

*Subject: Office Capacity Report

*Body:

Font 14 B I U [link icon] [list icon] [table icon] [code icon]

The biggest office is: {x} Biggest Office

Its capacity is: fx string(...)

The total capacity is: fx string(...)

The list of offices is:

Add the cursor where you want to display the html table and click **expression** where you can type the **replace()** function:

*Body:

Font 14 B I U [link icon] [list icon] [table icon] [code icon]

The biggest office is: {x} Biggest Office

Its capacity is: fx string(...)

The total capacity is: fx string(...)

The list of offices is:

Dynamic Value **Expression** [Format data by examples] [fx] [smiley icon]

replace()

Save Cancel

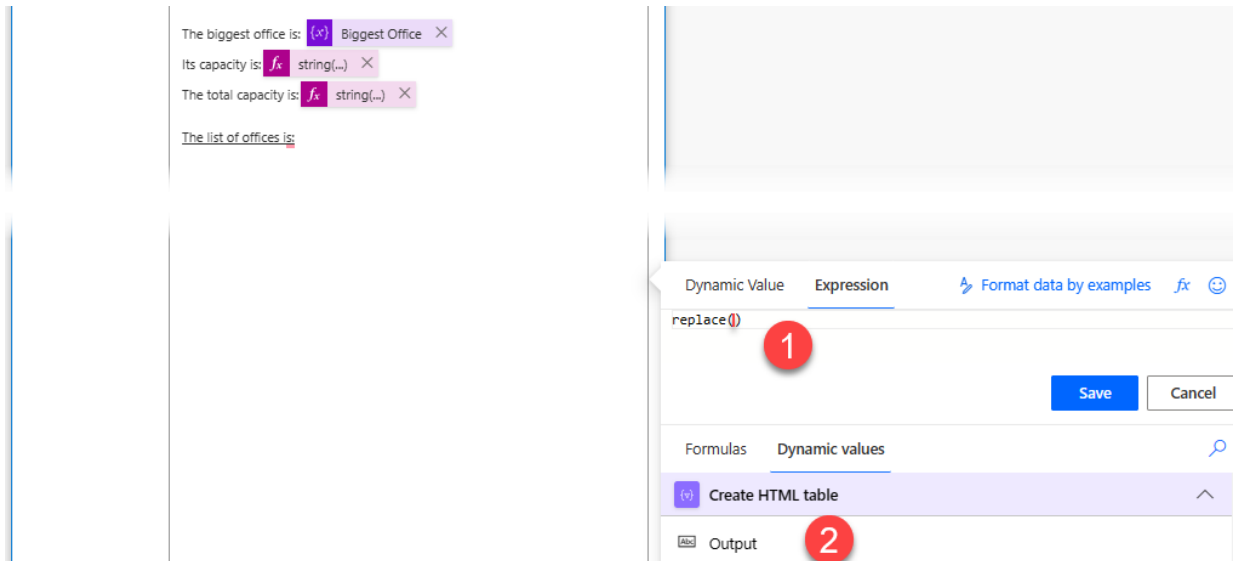
Formulas Dynamic values

String functions

concat(text_1, text_2?, ...)
Combines any number of strings together

[See more](#)

Move the cursor in the replace parentheses, and Click **Dynamic Value** (just down below, not the next one), to grab the Create HTML table – Output:



In replace, type `, '<table>', '<table border="1">'` , the final code should be `replace(body('Create_HTML_table'), '<table>', '<table border="1">')`, and click **Save**.

The mail will look like this:

The biggest office is: Toronto
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

32. A more advanced exercise is to display the capacity number in red if it is greater than 200, and in green otherwise.

- Edit the Create HTML table action, and in the capacity value field, add the following expression just to the left of capacity: (and click **Save**>

The screenshot shows the 'Create HTML table' configuration window. The table structure is as follows:

Header	Value
City	city
Capacity	capacity
Size	if(...)

The 'Capacity' field is selected, and a dynamic value expression is being added. The expression is:

```
if (greater(int(item()['capacity']),200),'<p style="color:red">','<div style="color:green">')
```

The 'Save' button is visible at the bottom right.

- b. Close the Div by adding `</div>` in plain text:

Create HTML table

*From

value

Header	Value
City	<div>city</div>
Capacity	<div><div>if(...)</div><div>capacity</div><div></div></div></div>
Size	<div>if(...)</div>

Show advanced options

The

- c. The generated e-mail will look like this:

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

The list of offices is:

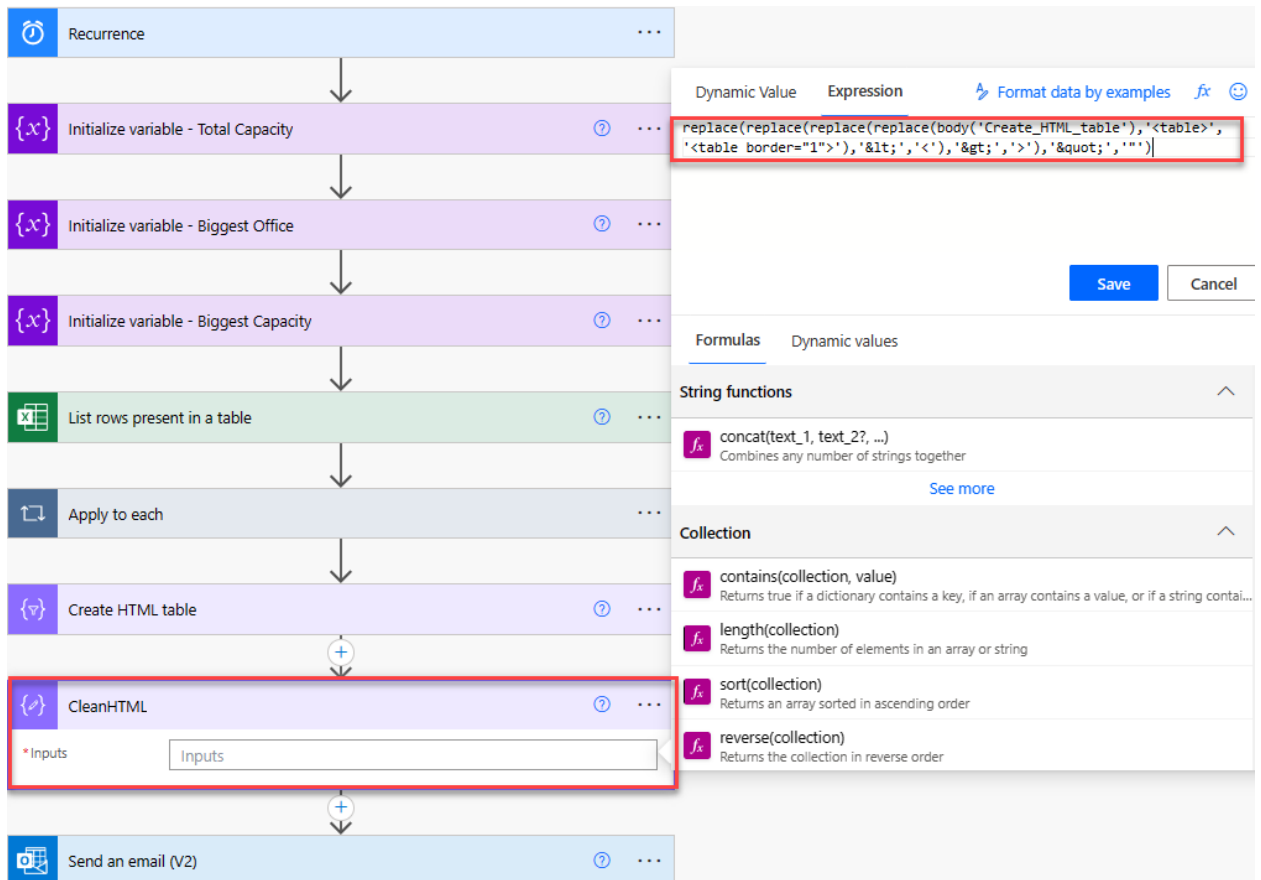
City	Capacity	Size
London	<div style="color:green">100</div>	Small
Brussels	<p style="color:red">250</div>	Big
Seattle	<div style="color:green">80</div>	Small
Vancouver	<div style="color:green">200</div>	Big
Toronto	<p style="color:red">400</div>	Big
Antwerpen	<div style="color:green">15</div>	Small
Warsaw	<p style="color:red">300</div>	Big
Paris	<div style="color:green">54</div>	Small
Berlin	<div style="color:green">70</div>	Small
Amsterdam	<div style="color:green">60</div>	Small
Montreal	<div style="color:green">78</div>	Small

Actually the html mbols <, > and ' are automatically replaced with their encoded value as illustrated in the following table

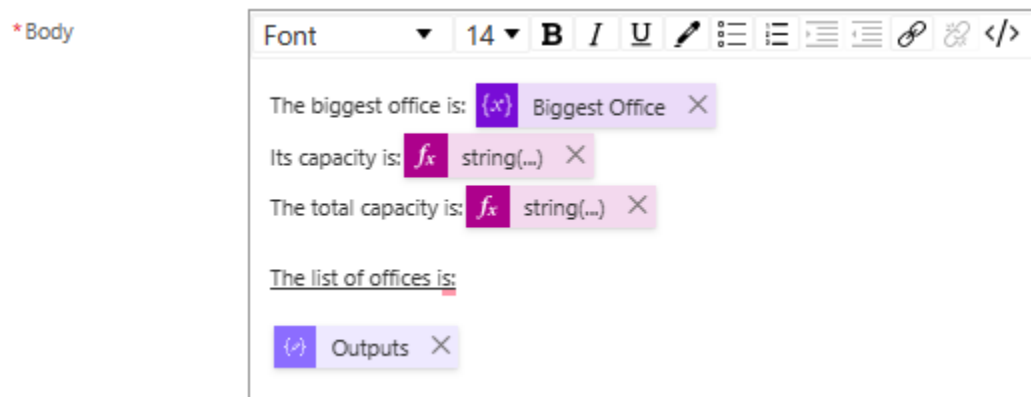
<	<
>	>
'	"

- d. After the create HTML Table action, add a **Compose** action (rename it **CleanHTML**) where you can add the following expression:

```
replace(replace(replace(replace(body('Create_HTML_table'),'<table>','<table border="1">'),'&lt;','<'),'&gt;','>'),'&quot;','"'))
```



e. Send the output of CleanHTM into the Send an e-mail action:



f. Test your flow; the received e-mail will look like this:

The biggest office is: Toronto
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 suggest something? We need your feedback:
<https://github.com/Power-Automate-in-a-day/Training-by-the-community/issues>