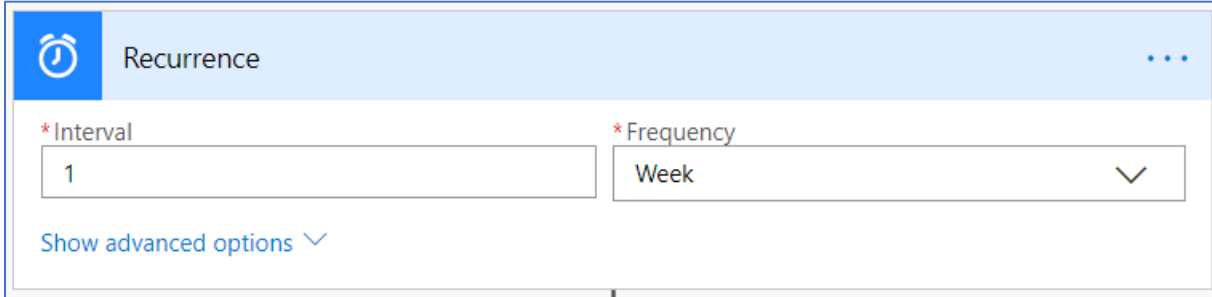


Get Old Documents

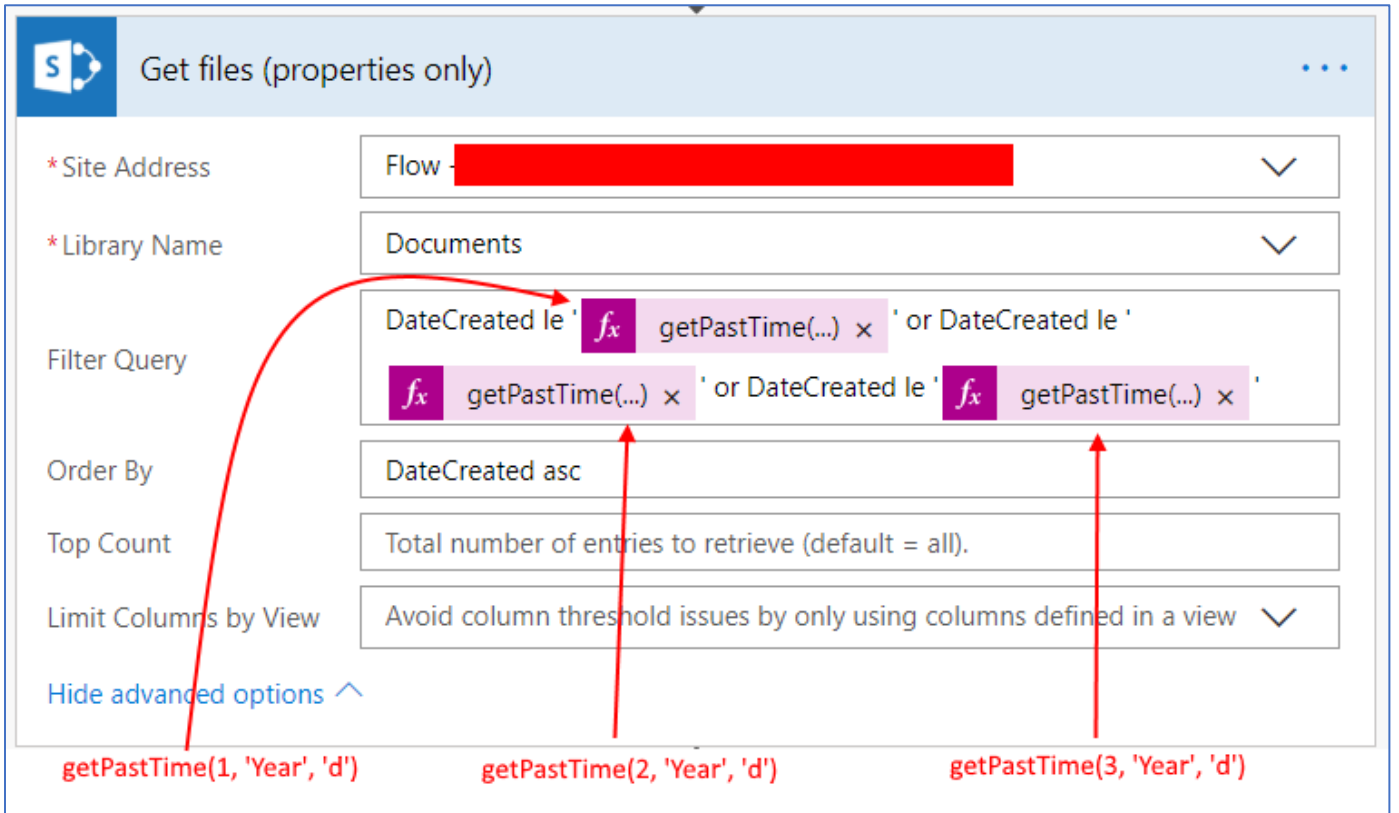
The approach I decided to take with this Flow is to first retrieve all the documents with the parameters set in the Filter Query, since I only need the documents that were created one year, two years, and three years ago from today's date, and that have the specific Interval, such as 1, 2, or 3. Once all these documents are retrieved, then I need to get all the names in the Created By field from all the documents and create an array from these names. These names will then be used to select all the documents that pertain to each Created By and create an HTML list to send via email to each person. This approach will prevent Flow from sending one email per document to the Created By retrieved in the queries mentioned above. If you follow all the steps below, you will be able to achieve the same outcome I got.

1. The trigger for this Flow is a **Recurrence**, which I set to run every week. You can always modify this to fit your needs.



2. The next step is a **SharePoint – Get files (properties only)** action. Set the value to the **Site Address** and the **Library Name**. Click on **Show advanced options** and set the **Filter Query** like the one in the image below. In my case, I filtered the returned items if the value of the **DateCreated** column is **less than or equals to** Today's date a year, two, or three years ago. See the **Filter Query** below. The highlighted text below are the expressions to evaluate the **DateCreated**. Additionally, I set the **Order By** field to the value of **DateCreated** ascending: **DateCreated asc**.

```
DateCreated le 'getPastTime(1, 'Year', 'd')' or DateCreated le 'getPastTime(2, 'Year', 'd')' or DateCreated le 'getPastTime(3, 'Year', 'd')
```



3. The next step is an **Initialize variable**, which I renamed to **Initialize arrayCreatedBy Variable**. I set the **Name** to **arrayCreatedBy** and the **Type** to **Array**. You can set your own variable name if you'd like.

{x} Initialize arrayCreatedBy Variable

* Name

* Type

Value

4. The next step is also an **Initialize variable**, which I renamed to **Initialize arrayDocuments Variable**. I set the **Name** to **arrayDocuments** and the **Type** to **Array**. You can set your own variable name if you'd like.

{x} Initialize arrayDocuments Variable

* Name

* Type

Value

5. The next step is a **Data Operations – Filter array**, which I renamed to **Filter Documents Array**. This action filters the output from the **SharePoint – Get files (properties only)** action to only show the items where the **Intervals** column value is **equals to 1, 2, or 3**. The filter has to be edited in **Advanced mode** to build the expression as shown below:

```
@or(equals(item()?['Intervals'], '1'), equals(item()?['Intervals'], '2'), equals(item()?['Intervals'], '3'))
```

{v} Filter Documents Array

* From

Dynamic content Expression

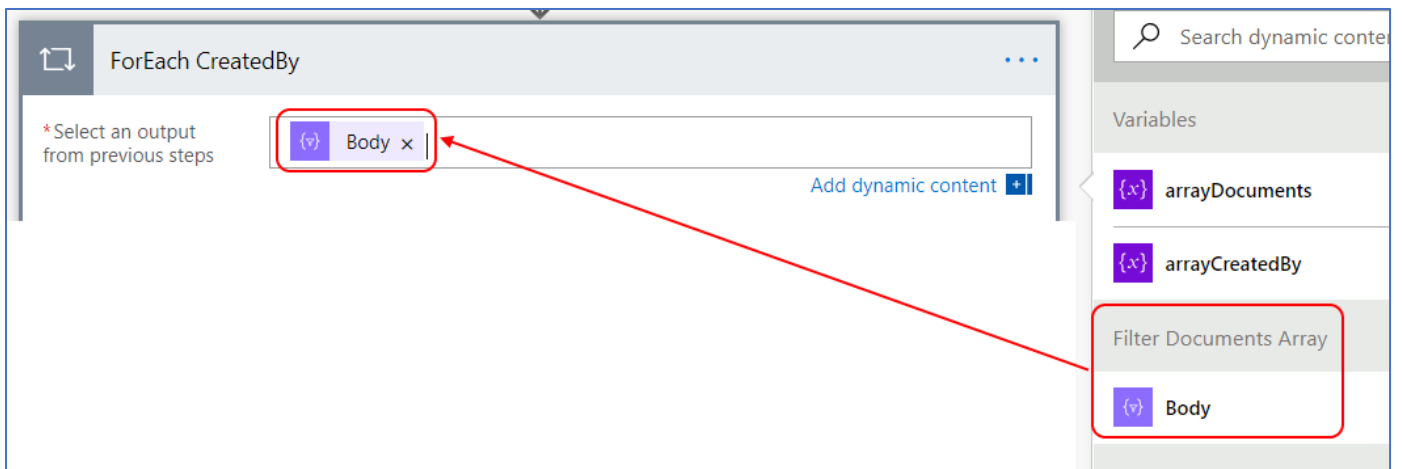
Search dynamic content

Variables

- {x} arrayDocuments
- {x} arrayCreatedBy
- Get files (properties only)
- value List of Items

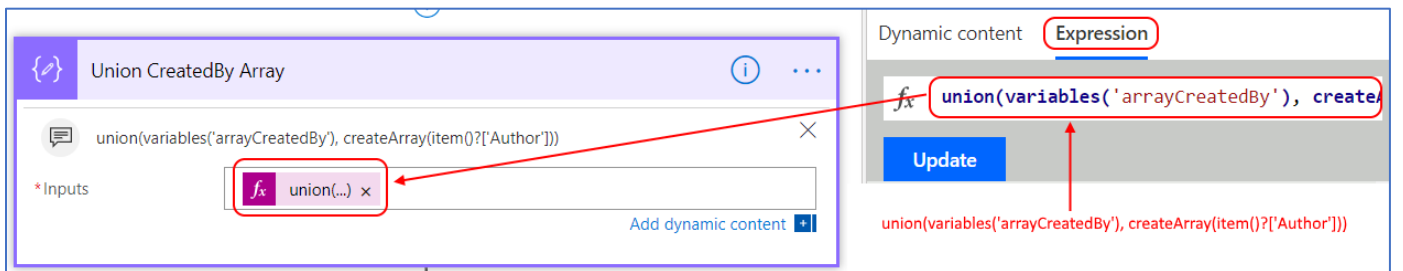
@or(equals(item()?['Intervals'], '1'), equals(item()?['Intervals'], '2'), equals(item()?['Intervals'], '3'))

6. The next step is an **Apply to each**, which I renamed to **ForEach CreatedBy**. For the ***Select an output from previous step**, I selected the **Output** from the **Filter Documents Array** step above.

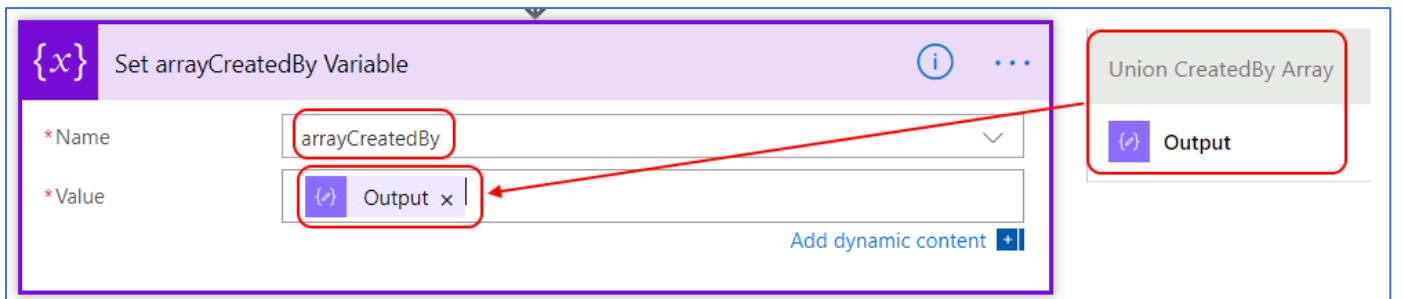


7. The next step is a **Data Operations – Compose**, which I renamed to **Union CreatedBy Array**, and it's inside the **ForEach CreatedBy** loop above. Set the ***Inputs** to the expression below. This expression joins all the names in the **Created By** columns and creates an array. The output from this step will be used in the next one.

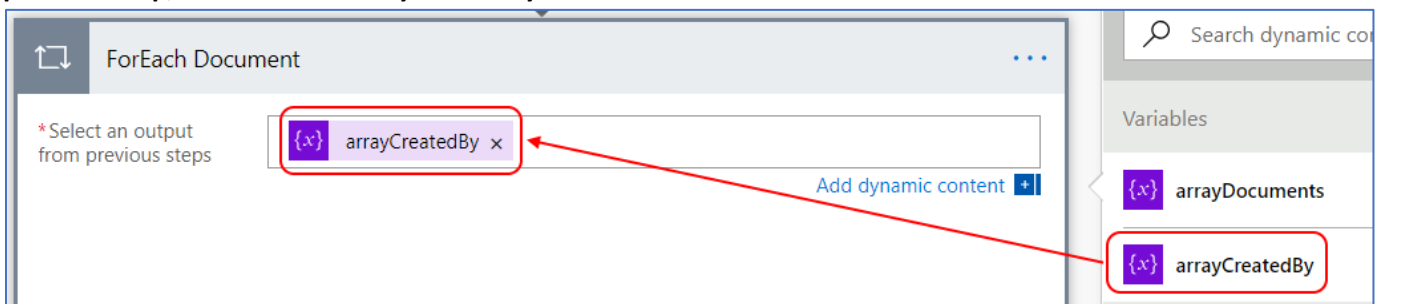
```
union(variables('arrayCreatedBy'), createArray(item()?['Author']))
```



8. The next step is a **Set variable**, which I renamed to **Set arrayCreatedBy Variable**, and it's inside the **ForEach CreatedBy** loop above. I selected the **arrayCreatedBy** from the dropdown and set the **Value** to the **Output** from the **Union CreatedBy Array** step above.

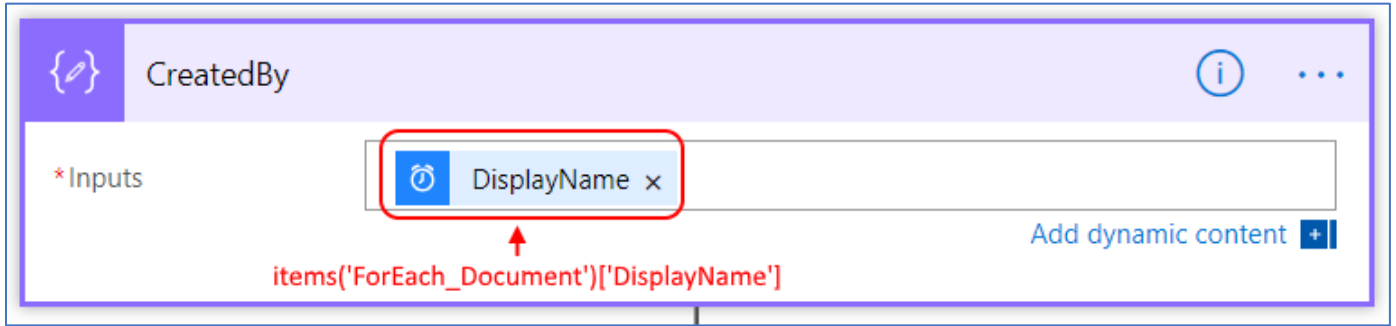


9. The next step is another **Apply to each**, which I renamed to **ForEach Document**. For the ***Select an output from previous step**, I selected the **arrayCreatedBy**.

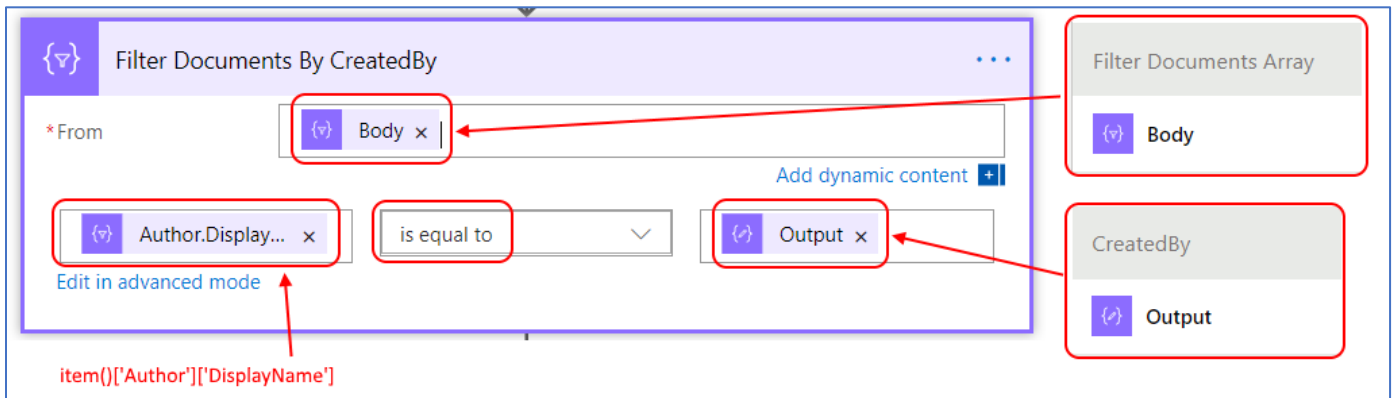


10. The next step is a **Data Operations – Compose**, which I renamed to **CreatedBy**, and it's inside the **ForEach Document** loop above. This action selects the **DisplayName** of each person in the **arrayCreatedBy** array, and to achieve this, the expression below needs to be used.

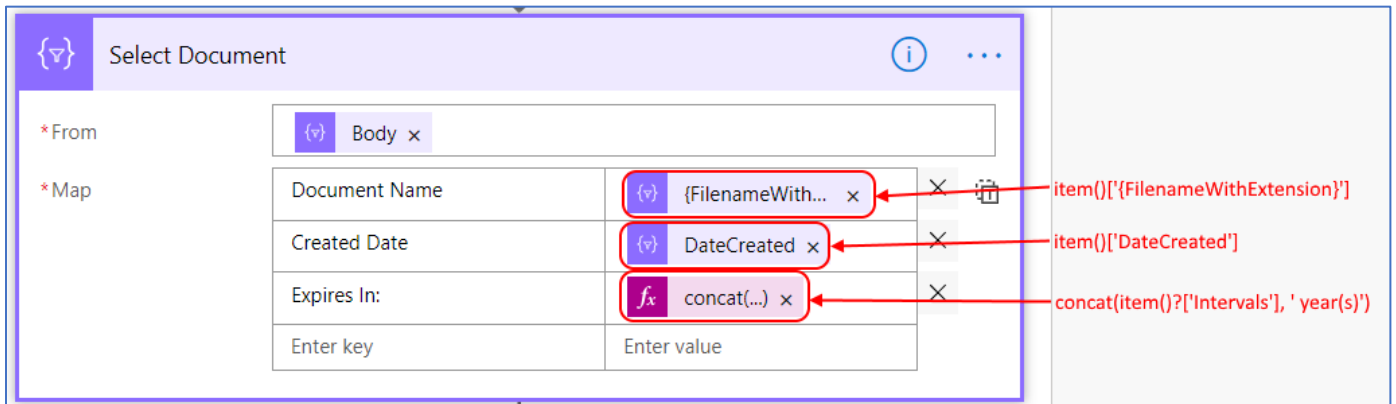
```
items('ForEach_Document')['DisplayName']
```



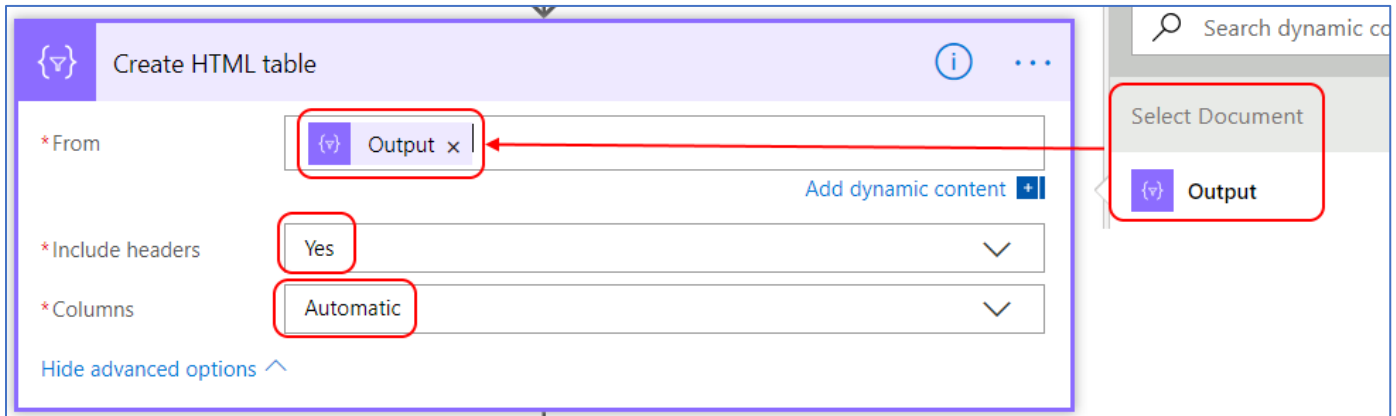
11. The next step is a **Data Operations – Filter array**, which I renamed to **Filter Documents By CreatedBy**, and it's inside the **ForEach Document** loop above. This action filters the output from the **Filter Documents Array** step above. Set the ***From** to the **Body** from the **Filter Documents Array** step, and then set the value to evaluate to the expression `item()['Author']['DisplayName']`. Select **is equal to** from the dropdown and then set the next value to the **Output** from the **CreatedBy** step.



12. The next step is a **Data Operations – Select**, which I renamed to **Select Document**, and it's inside the **ForEach Document** loop above. This step is used to select certain properties you want to display in the email body. Set the ***From** field to the **Body** from the **Filter Documents By CreatedBy** step, and ***Map** to the properties you want. In my case, I used **Document Name**, **Created Date**, and **Expires In**. The key can be set to whatever you want, but the value has to be the property available from the **Body** of the **Filter Documents By CreatedBy** step. The image below contains the expression used in each key value.



13. The next step is a **Create HTML table**, which goes inside the **ForEach Document** loop above. It's used to create the HTML content that will go in the email body. Set the ***From** field to the **Output** from the **Select Document** step above. Click on **Show advanced options** to set the ***Include headers** to **Yes** and ***Columns** to automatic.



14. And the last step is an **Outlook – Send an email (V2)** action, but you can use the main one available in the Outlook connector. Set the ***To** field to the expression `items('ForEach_Document')['Email']`, then set the ***Subject** to whatever you need. For the ***Body**, use the **Output** from the **Create HTML table** action above, and set the **Importance** to **Normal**. Save the Flow and run it to make sure it works as expected.

