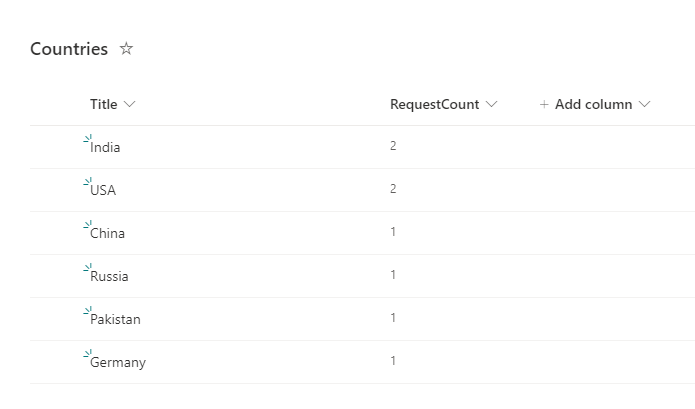
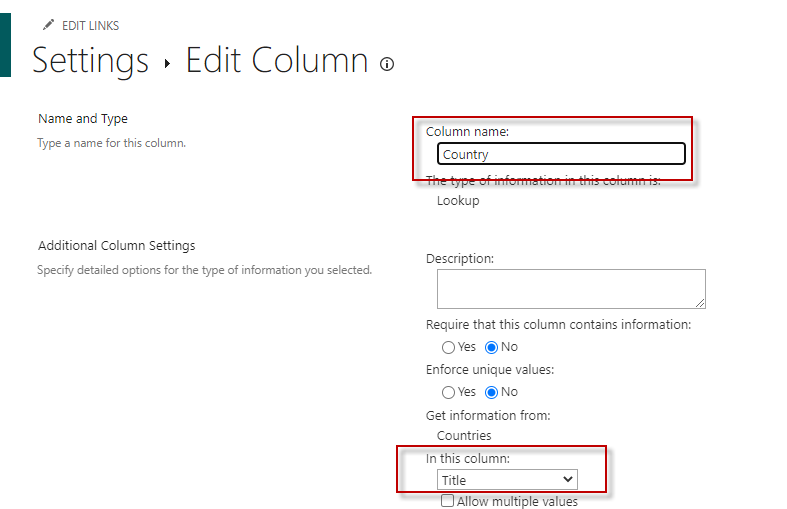
# List Configuration in SharePoint:

Create new lists called ‘Countries’ and populate the lists with some country values. Create RequestCount of data type ‘Number’.

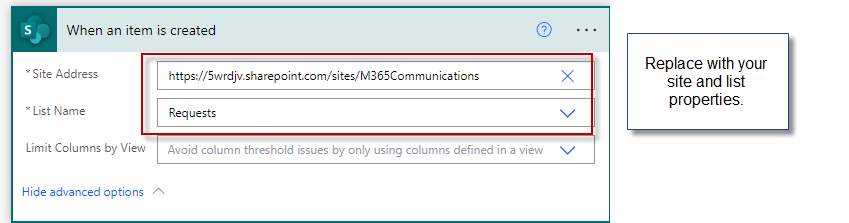


Create new list called ‘Requests’ and here add look up column called ‘Country’and reference to the ‘Title’ field.

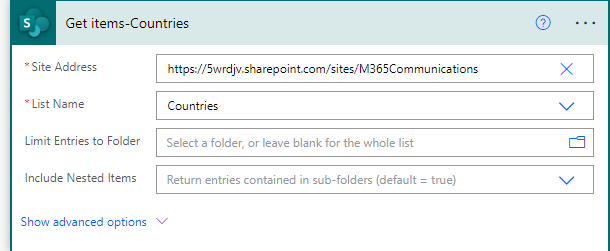


# Flow configuration in ‘Power Automate’:

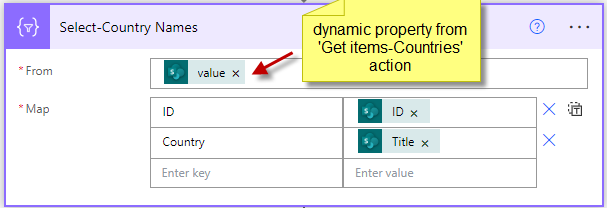
Step 1: create the trigger of the flow on when item is created. Update the parameters with your site name and the list name. Here I am choosing the list ‘Requests’ because this is where I need to have the workflow to trigger when a new item is created.



Step 2: Get the items from the list ‘Countries’. Just replace with your site and list properties and leave everything as default.

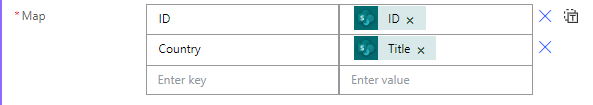


Step 3: Next use the ‘Select’ operation to filter the outputs from the ‘Get items-Countries’ action. Since ‘Get items’ gives as lot of properties, it is best to select the data only we need for this request update operation.

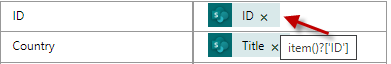


From: It is dynamic property that is from the previous action ‘Get items-Countries’.

Map: This is important step to filter the output. Needs to be given in key value pair. The left side is the Key which can be hard coded. The right side is the value that we need to extract.



For the Key ‘ID’, the expression value is *item()?[‘Title’]*



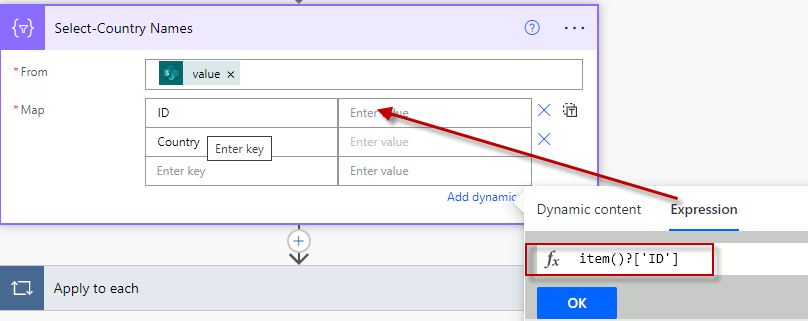
For the Key ‘Country’, the expression value is *item()?[‘Title’]*



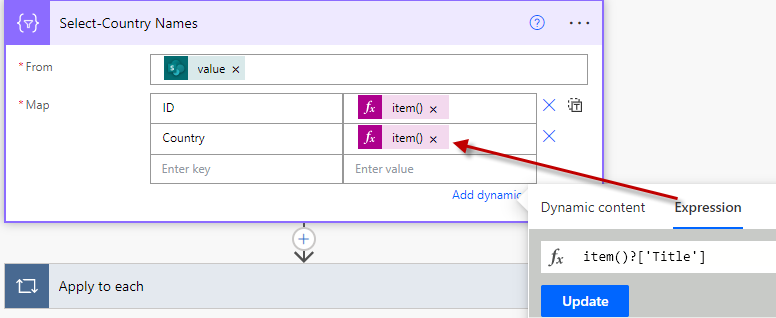
Item(): it is built in function that determines the current item in the collection. From the current item() we want to select specific value from it we can use ‘?’(select operator) and then followed by name of the property in Square brackets in quotation. Below is the syntax.

*Item()?[‘PropertyName’]*

Also please note that the values in the right-hand table needs to be exclusively written in expression. Below is the screen capture for the ID Key. Same steps needs to be done for the ‘Country’ key.

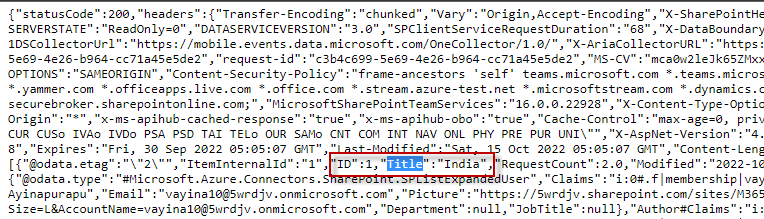


Similarly for the Country,

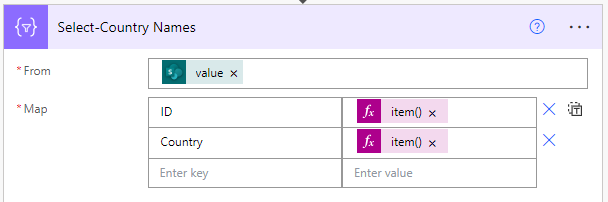


If you have observed the outputs from the previous runs, the values we are getting from the ‘Get items-Countries’ contains these property names ‘ID’ and ‘Title’ in the outputs and hence mentioning the expression is very important step to get the desired values.

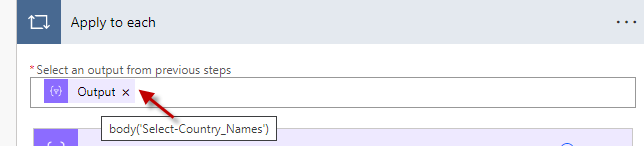
Below is the screen shot of the values that we need from the ‘Get items-Countries’.

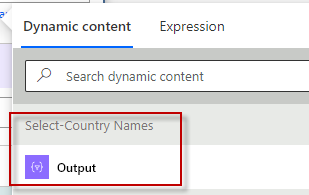


The final action should look like below after configuring the ‘Select’ action.

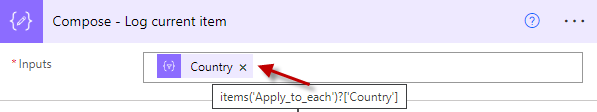


Step 4: Now add ‘Apply to each’ and select the output from ‘Select-Country Names’ action. This is dynamic property which is directly available.





Step 5: Now add ‘Compose’ action inside ‘Apply to each’ loop to log the current item value.

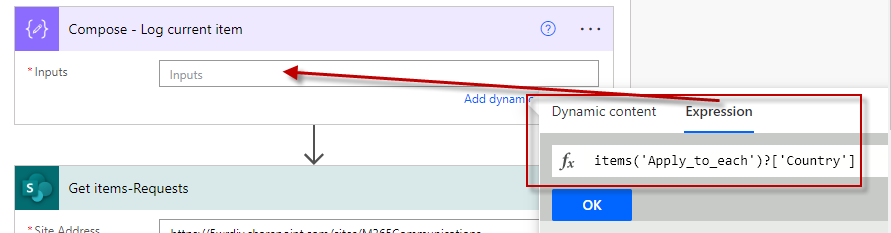


To log the current item value below is the syntax

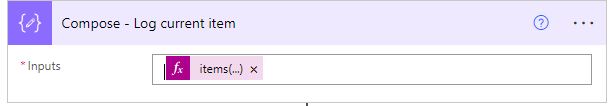
Items(): items is the function that represent the collection of values. To log each value from the collection, below is the syntax

*Items(‘LoopName’)?[‘PropertyName’]*

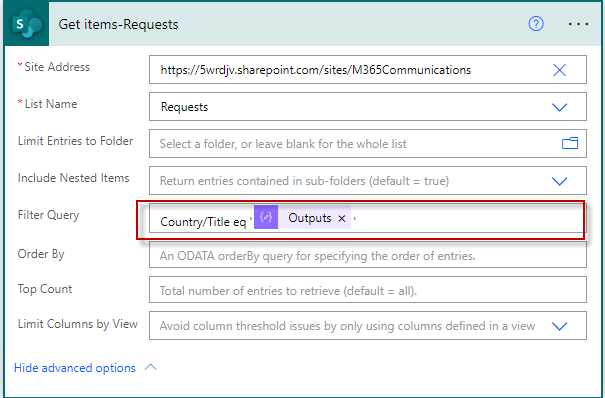
Here we can see the display name of the loop. But the flow query respects only internal name. In the internal names, the spaces will be replaced with underscore(\_). In this case the display name of the loop is ‘Apply to each’. The internal name will be ‘Apply\_to\_each’. Please note that it is required to input the



The final configuration should look like below.



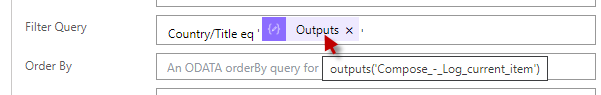
Step 6: Now add the action ‘Get Items’ inside the ‘Apply to each’. This time, we are using OData filter query to look for the items in ‘Requests’ list that matches



Since in the ‘Requests’ list ‘Country’ is of type lookup column, it is required to mention the query in the below format

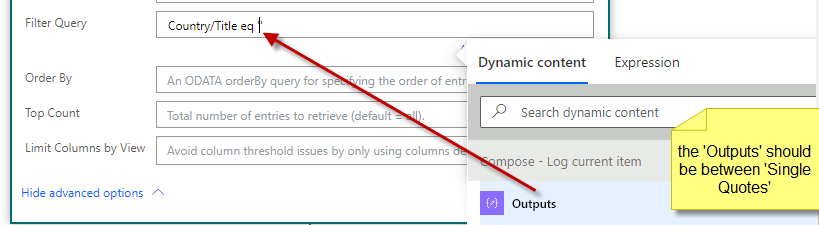
{LookUpColumnName}/{ReferencePropertyNameInLookUpReferenceList}

In our case, the filter query is as below screen shot

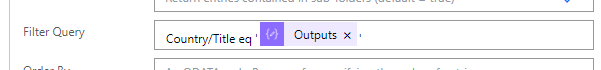


Again the ‘Outputs’ is dynamic function which is readily available. When mentioning the query you have to mention the following in filter query and with in quotes insert the dynamic value from the dynamic properties.

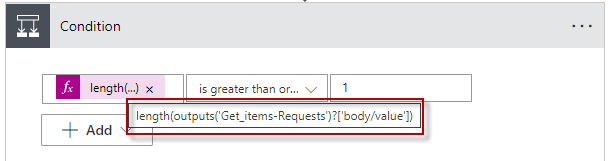
Country/Title eq ‘[InsertFromDynamicPropertySection]’



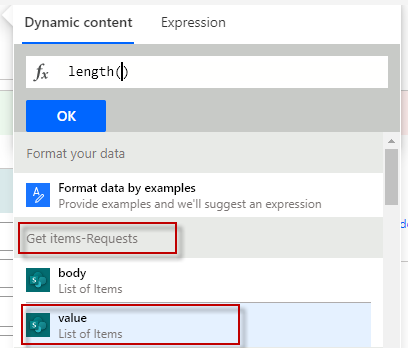
The filter query should look like below.



Step 7: Now add condition and configure it as shown below

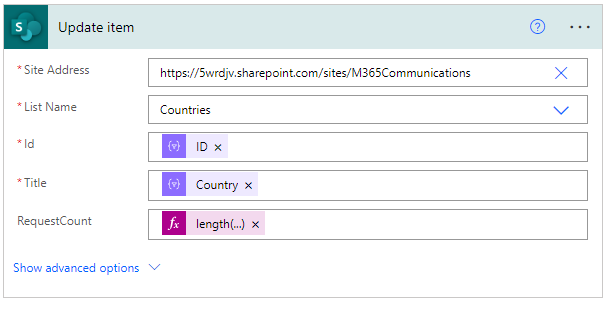


This is expression again. So use length() and with in the brackets, inset the dynamic property of outputs from ‘Get items-Requests’ action.



And then comparing if the collection returned from ‘Get items-Countries’ is greater than or equal to 1, which means if this collection contains one or more items, then update the request count in the ‘Countries’ list.

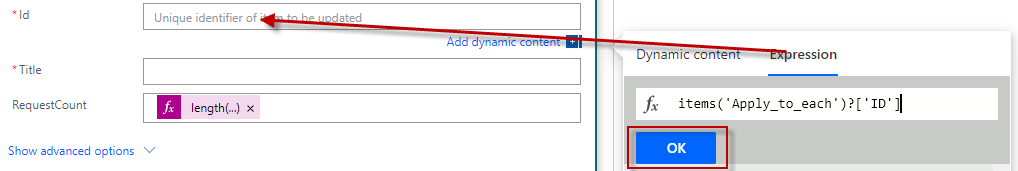
Step 8: Under the Yes block, Add ‘Update item’ action for SharePoint, and then configure the update action as per the below screen capture.



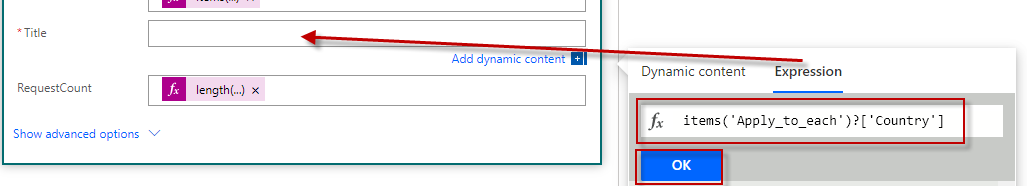
Site Address: Your site address where the values needs to be updated.

List Name: List name in the site

Id: Item ID for which the value needs to be updated. This is expression we need to write. Since we are getting Item ID, and Country Name in the ‘Apply to each’ loop, the expression for Id is *items(Apply\_to\_each)?[‘ID’]*



Title: item Title, for which the values need to be updated. The expression for Title is *items(Apply\_to\_each)?[‘Country’]*



RequestCount: The length of the collection. Follow the same steps in step 7 to insert the expression for the value of total count.

Step 9: Finally save and test the flow. You should see the request count getting updated on each item creation.