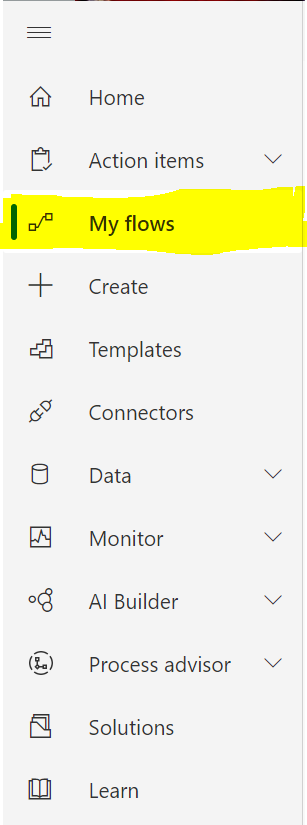
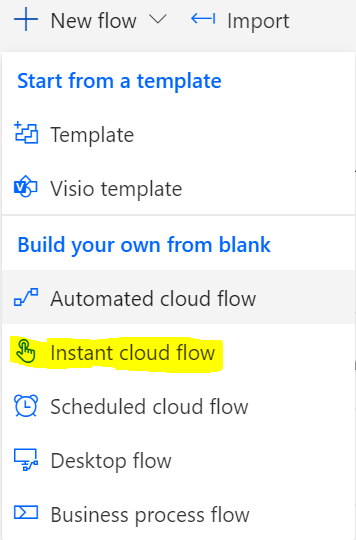
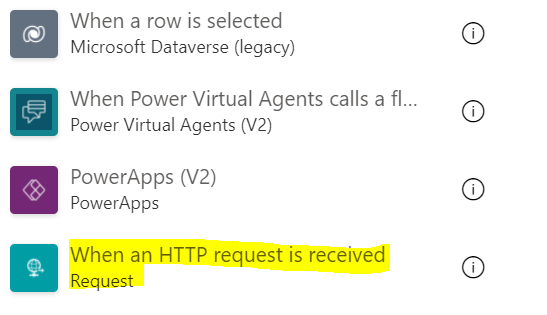
1. Open the Power Automate home page from Office 365.
2. Select My Flows.



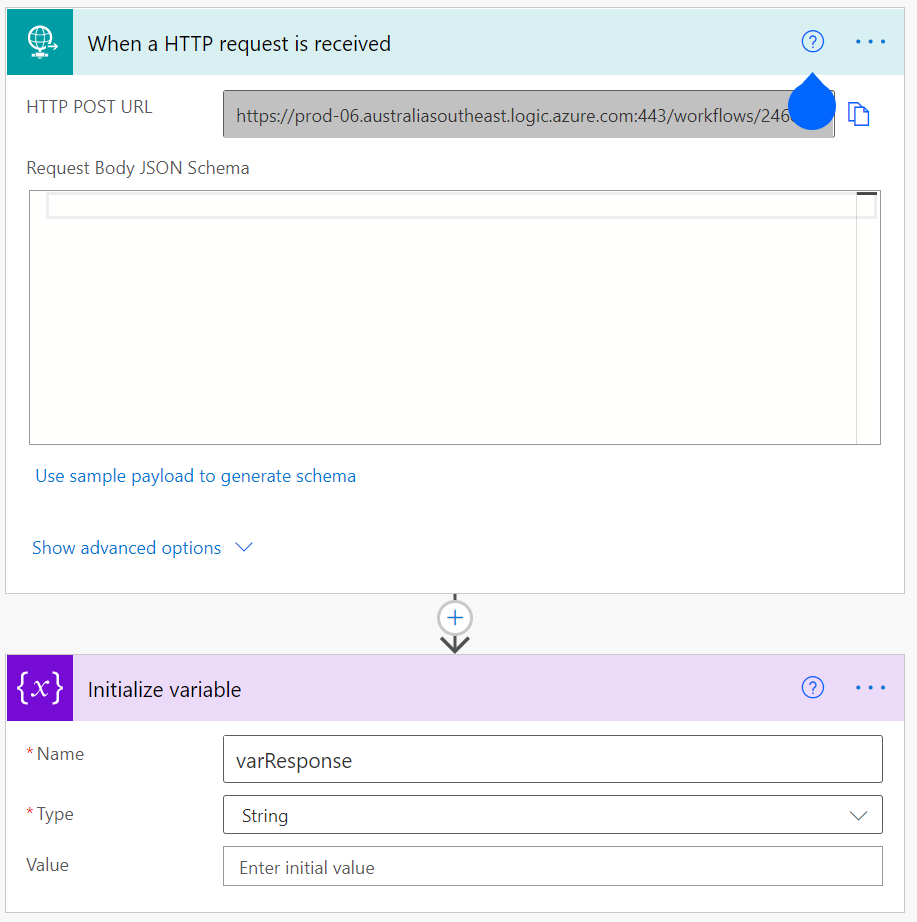
1. Create a new flow with ‘Instant cloud flow’ as the trigger type.



1. Use ‘When an HTTP request is received’ as the trigger.

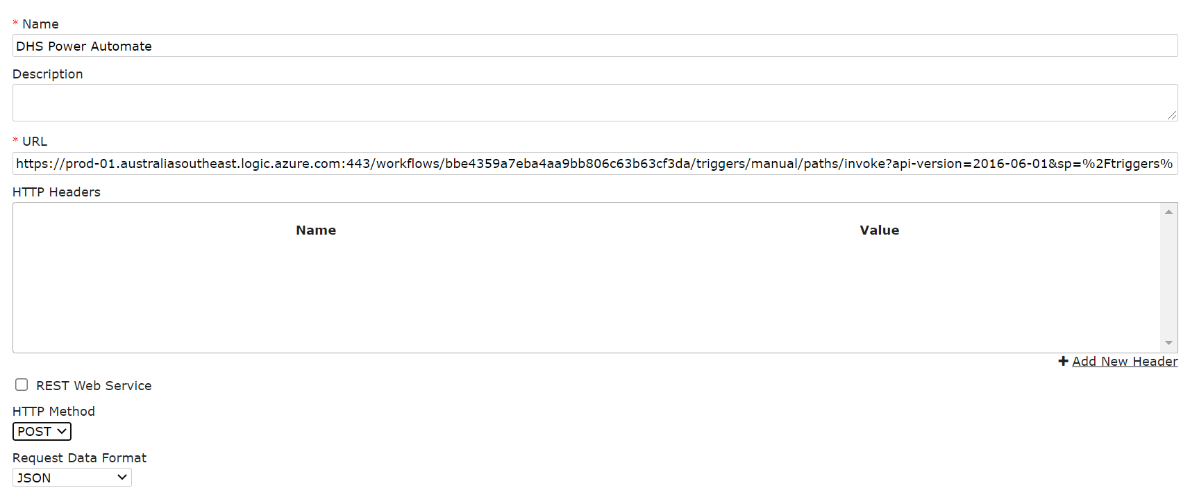


1. Insert a new step into the workflow and save to generate the URL for the trigger. The new step can be anything, the flow just needs one action before it will save. Initialize variable or Delay are the easiest because they don’t need a dynamic value and you’ll either delete this step or use it later anyway.



1. After the flow has saved, copy the URL that is generated from the HTTP request trigger. Use this end point as the URL for the web service in iApply. Make sure the Method is POST and the Data Format is JSON.

Access web services by clicking settings in iApply. To create a new one scroll down and click “create new web service”



1. Add a WebHook action to the iApply form workflow using the Web Service you configured in the previous step. To add a webhook action click on the form in iApply, click the workflow tab. Right click on the submit action, select ‘Add Action’ and then webhook.

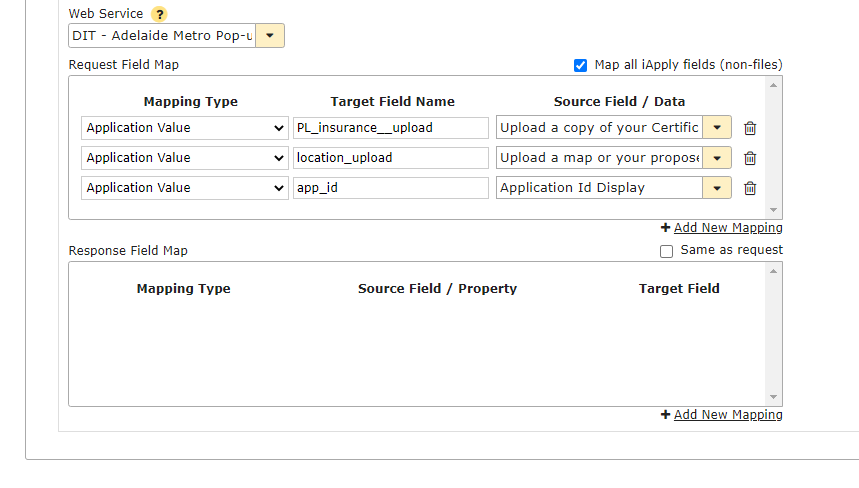
Make sure “Map all iApply fields” is checked.

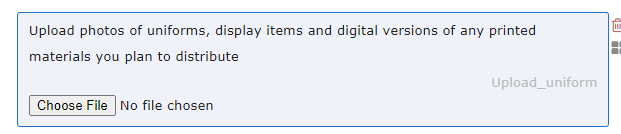
Note:

If some of the variables do not map correctly or you do not get the expected answer, you may need to map manually.

If in SharePoint your data is random characters instead of a string, map manually and select the description option instead of the value option in the source field selection.

1. For forms that require attachments to be added to SharePoint, add a new mapping

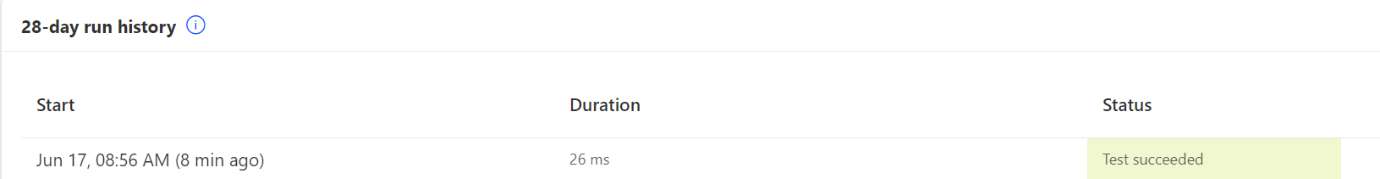


Give it a target field name that is different from the name associated with the question where the file is uploaded in iApply (example question below). When selecting the source field make sure it is the base64 option.

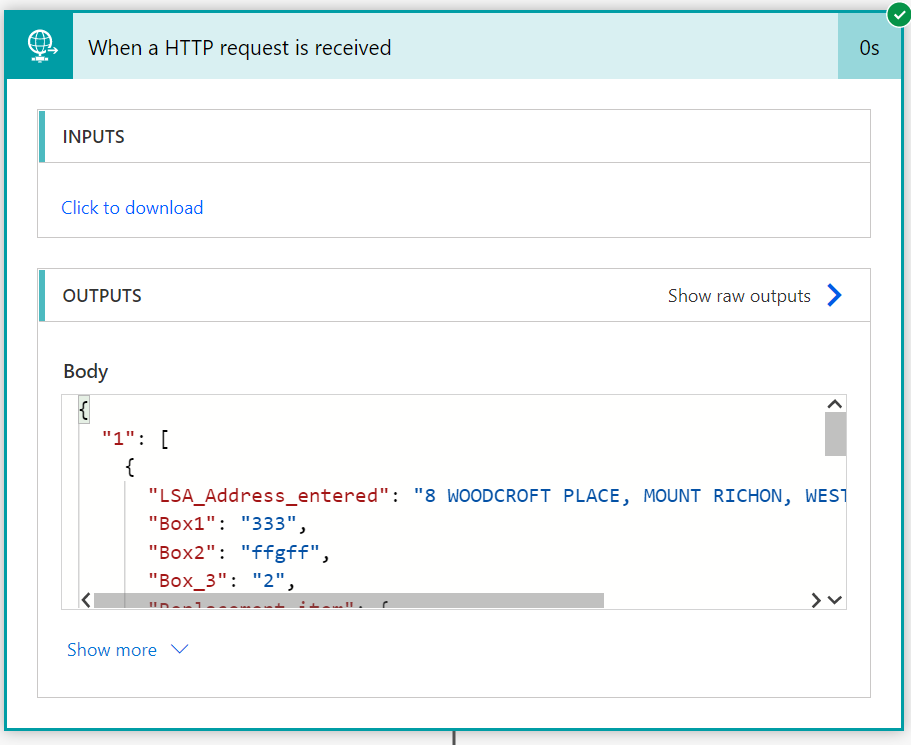
In this example the target field name should not be “Upload\_uniform”

Repeat this step if you would like the application ID to be sent to SharePoint. (Make sure ‘Application Id Display’ is selected)

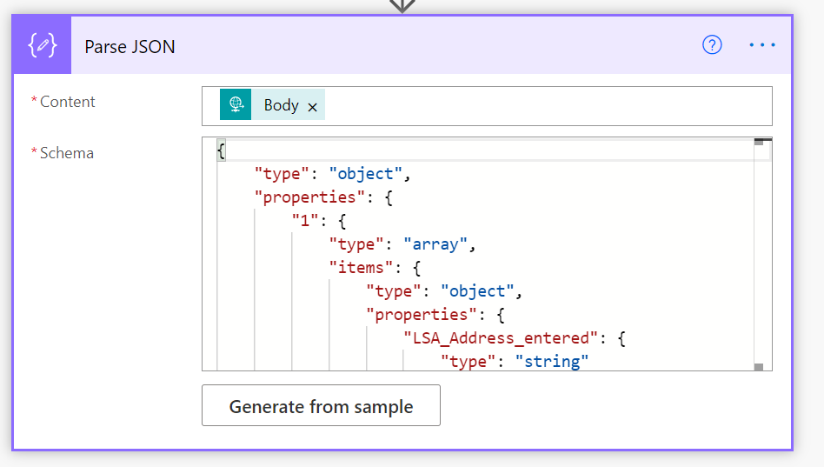
1. Submit the iApply form to trigger a test run of the workflow. This is required to generate the JSON schema in the Power Automate workflow, much easier than writing it yourself. Remember you do not need to submit a new form for every test, power automate has a resubmit button (Providing all your JSON is correct).
2. Go back to the Power Automate workflow and open the run you have just generated to view the response details. This can take a couple of minutes to come through from iApply.

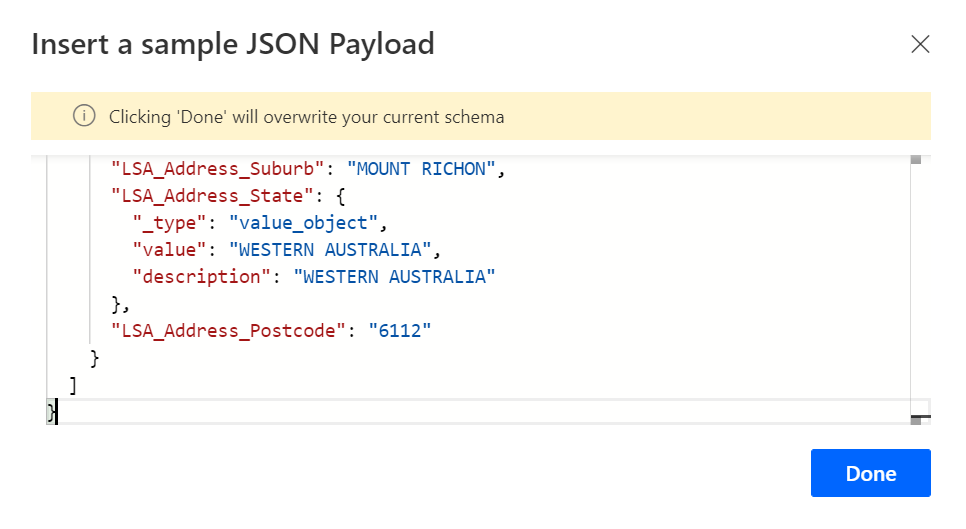


1. Copy all text in the Body field of the HTTP request trigger.



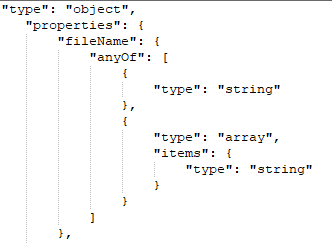
1. Delete the placeholder action you created in step 5 if it is not needed and add a new action ‘Parse JSON’ to the workflow.



1. Add a schema by clicking ‘Generate from Sample’ and pasting the body response from the trigger you copied in step 11, then click Done.

This will generate a schema. You will want to make sure each of the fields you need are in the JSON schema before moving on. If not you may have to enter them manually into the schema by referring to the iApply form’s questions. This may be due to form logic and not every answer being filled out.

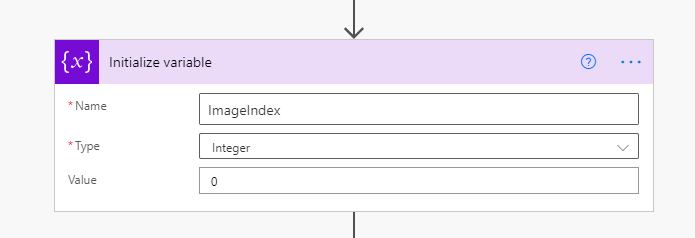
For multiple file uploads you may want to change the JSON to include strings and arrays



1. If you have a file to be attached along with the data, add a new step and select initialise variable. Otherwise skip this step.

Give the variable a name and set it to zero.

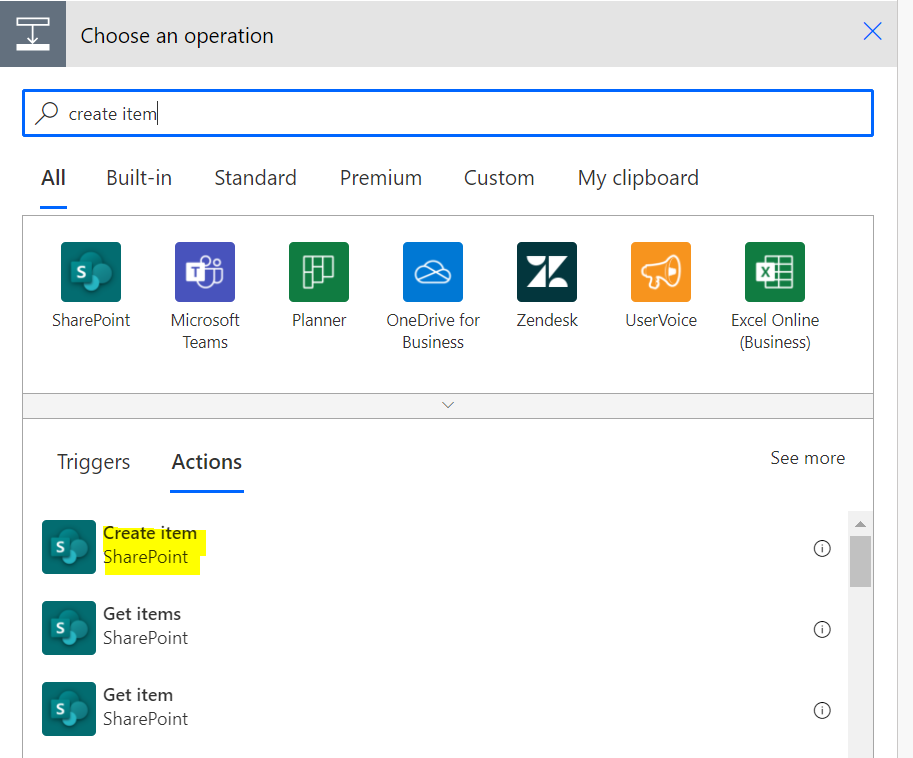
Now is also a good time to set up your SharePoint list if you have not already.



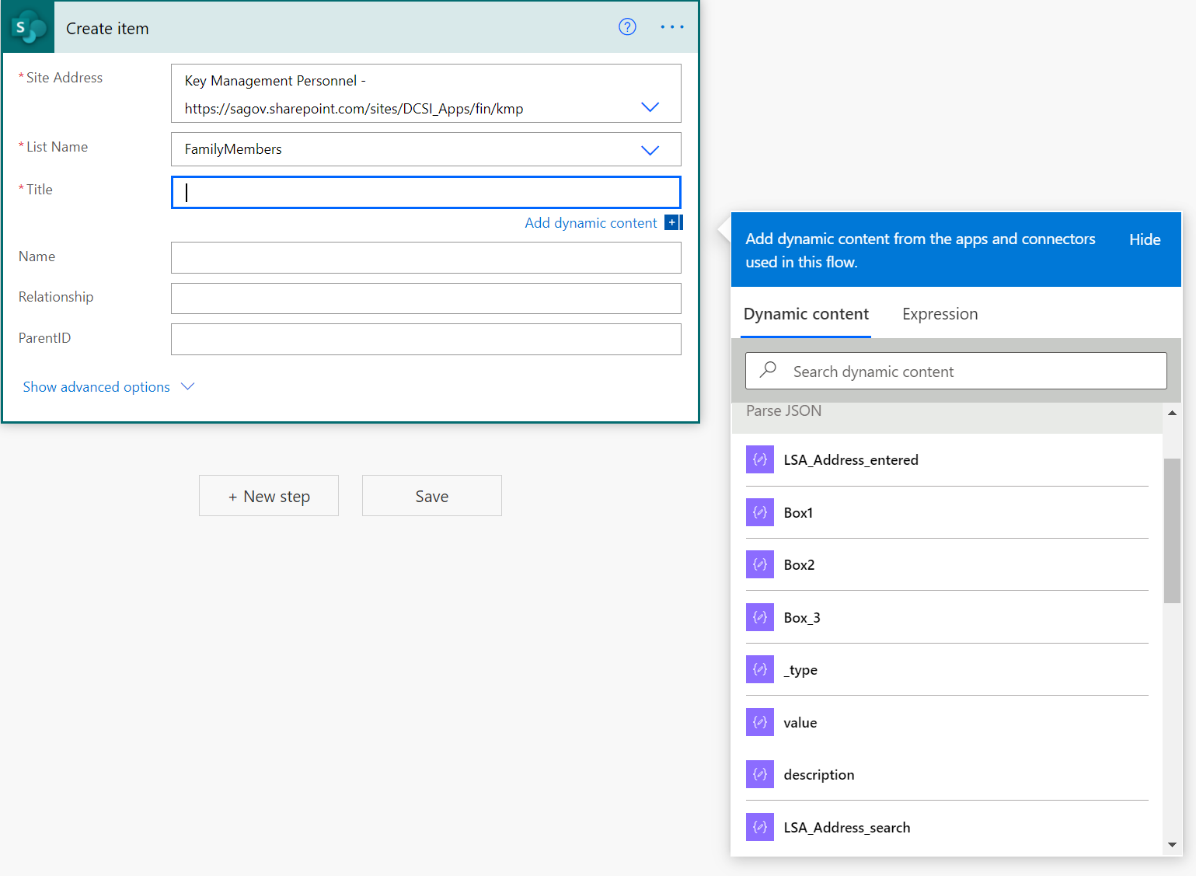
1. (Optional) you may need to add an ‘Apply to each’ step here if you have a date repeater.

If your form has an option for multiple days/date events go to step 25, otherwise continue.

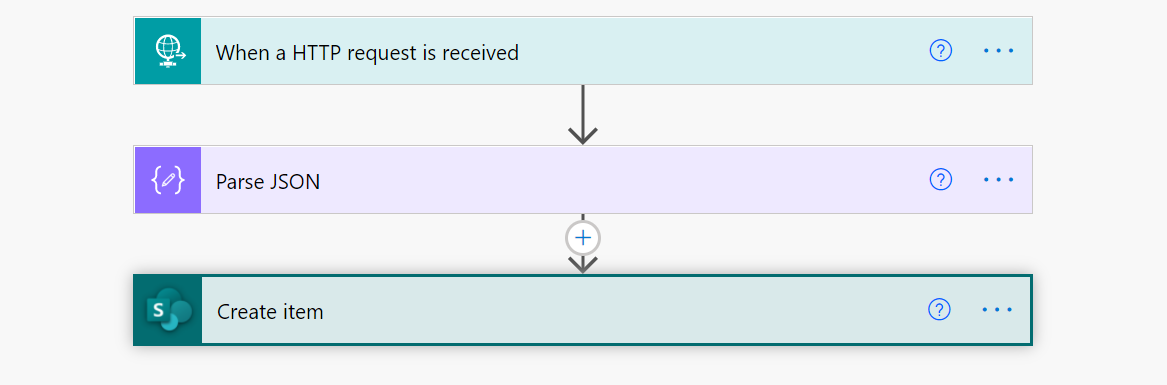
1. Add a new step using the SharePoint ‘Create Item’ action.



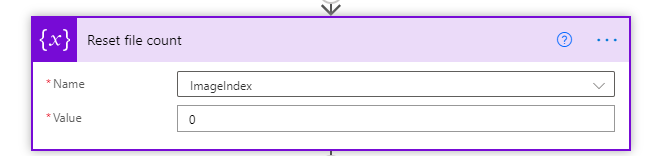
1. In the Create Item configuration screen, map the SharePoint list fields with the iApply data by selecting Add dynamic content and selecting the appropriate value from the Parse JSON action results.



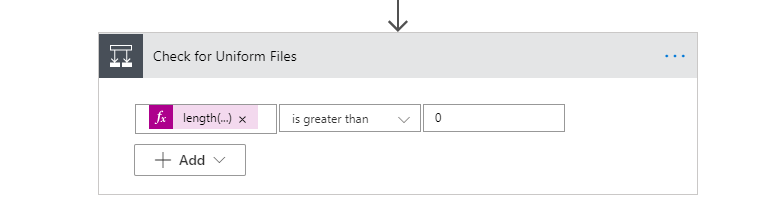
If you have no files to attach your finished workflow should look like this. Otherwise go to the next step.



1. Add a new step that resets the file count. This is a ‘set variable’ step



1. Add a new step to check for the file. This new step is a ‘control condition’ step



Enter in the left box the text below. Remember to add this in the ‘Expression’ section not the dynamic content section.

length(body('Parse\_JSON')?['TARGET FIELD NAME'])

e.g: length(body('Parse\_JSON')?['file\_uniform\_display'])

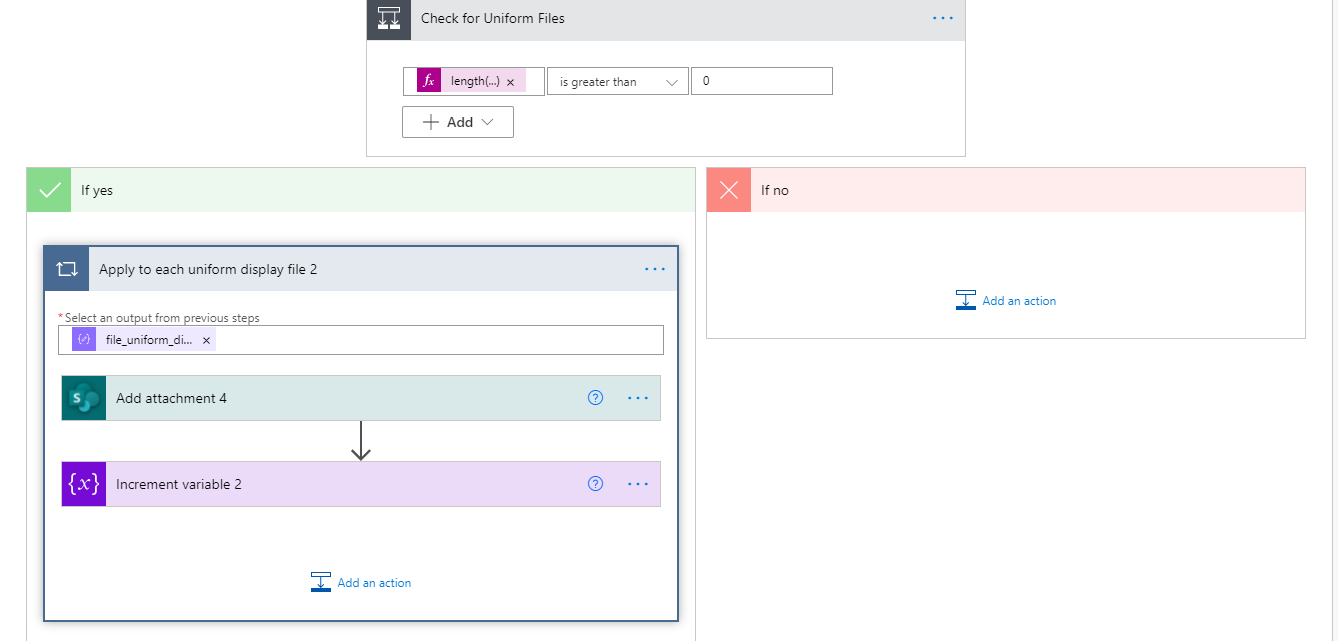
Replace TARGET FIELD NAME with the name you entered in your web service in step 8. Click OK or update to confirm the change.

Select ‘is greater than in the middle box’ and enter ‘0’ in the right box.

1. In the yes condition add an ‘Apply to each’ step

In the dynamic content box enter in the JSON target field name

In this case it is “file\_uniform\_display”



1. Add a new step to add an attachment. This step is called ‘add attachment’.

In ‘site address’ select your SharePoint site.

In ‘list name’ select your SharePoint list.

If you are unsure what these are refer to your SharePoint list’s name and where it is located.

In ‘Id’ enter and then select ID from dynamic content.

In ‘file Name’ enter in a prefix for your file name and then add split(body('Parse\_JSON')?['Upload\_uniform'], ',')[variables('ImageIndex')]

Into the expression box.

Note:

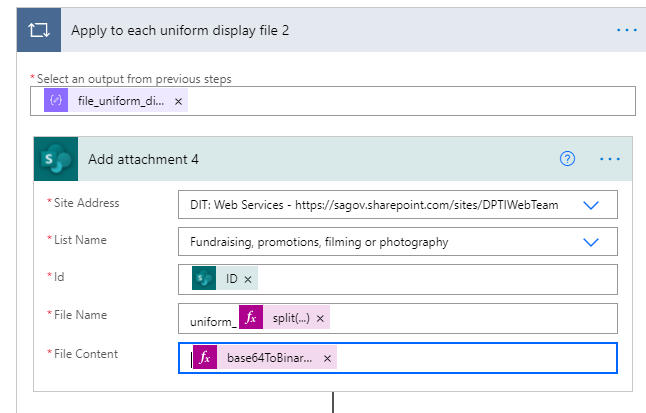
‘ImageIndex’ is the name of the variable you created.

‘Upload\_uniform’ is the name given to the question in iApply where the file is uploaded.

These will be different for your form and variable.

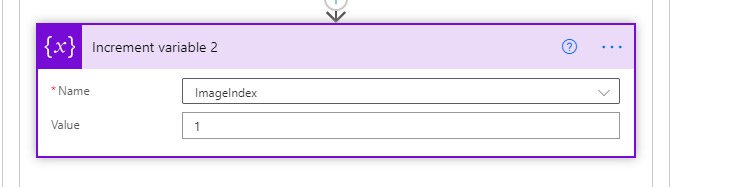
In ‘file content’ enter in the expression box: base64ToBinary(items('Apply\_to\_each'))

Note: ‘Apply to each’ is the name of the ‘apply to each’ section, so if you have renamed it you will have to type the name into this section and not “Apply\_to\_each”.



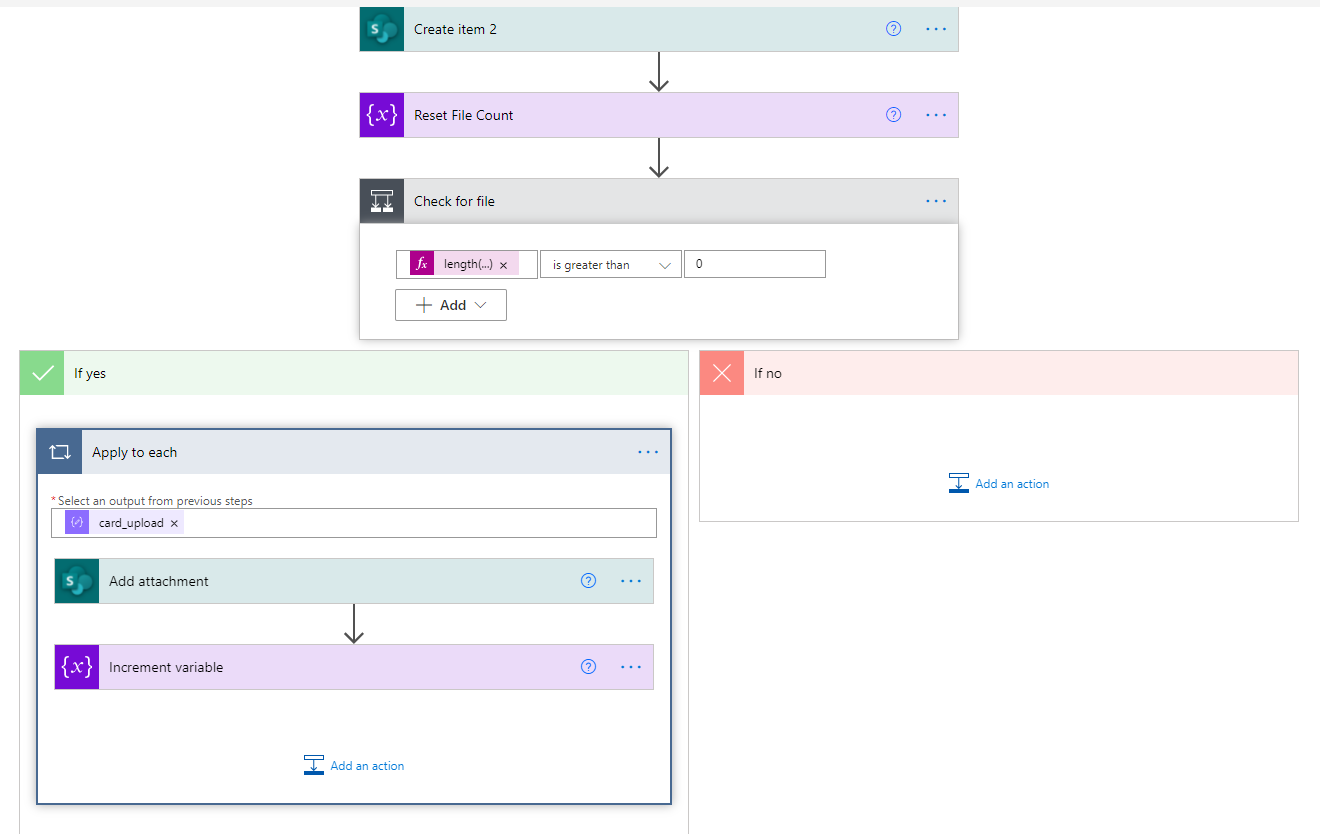
This converts the file in JSON from base64 to binary so it can be attached.

1. Add a new step to increment the variable. This step is called “increment variable”



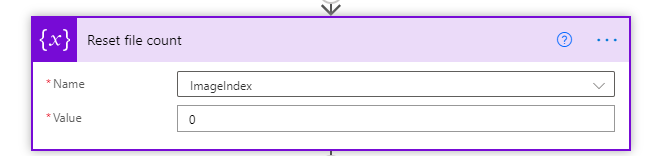
Enter your variable name into “Name” and 1 into the “value” to increment it by 1

If your form only has one file your finished flow should look like



If you have multiple files go to the next step.

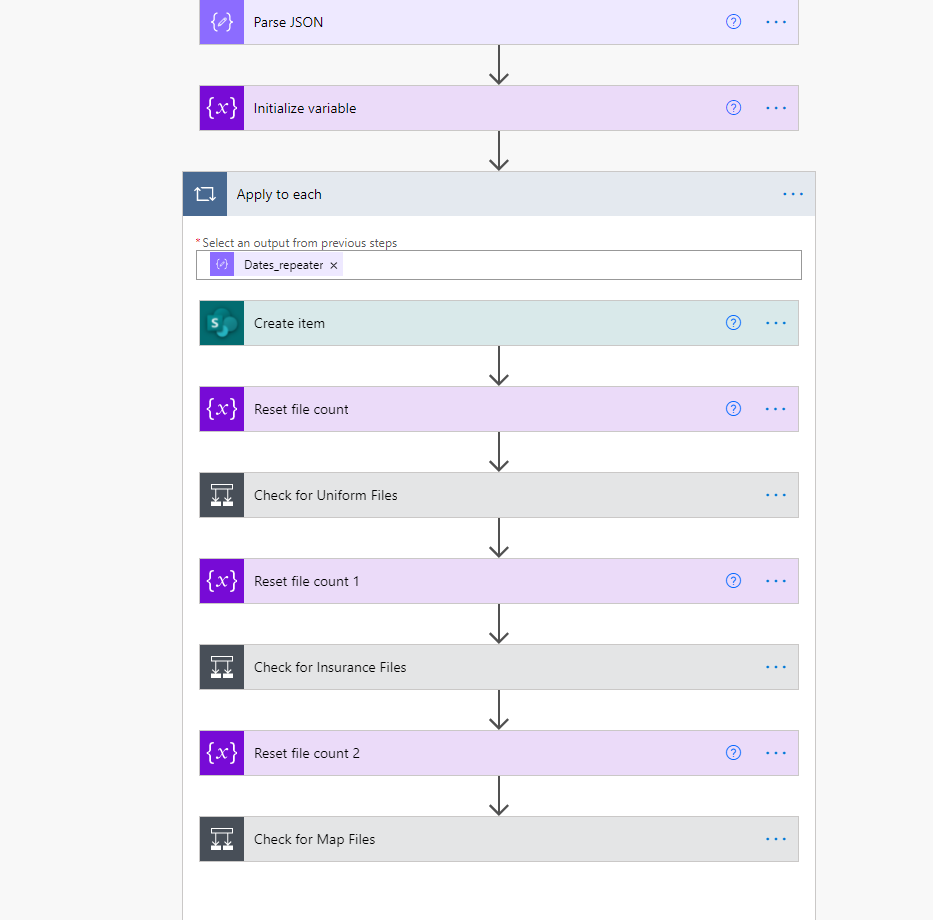
1. Add in a new step to reset your file count with a “set variable” step



Enter your variable name into “Name” and 0 into “value”

1. Repeat steps 18 to 22 as many times as necessary (one per file).

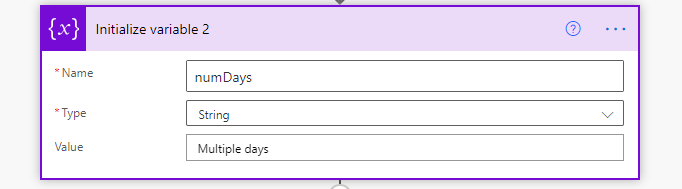
Your final product should look like



However this may depend on the steps you have taken.

IF YOUR FORM HAS MULTIPLE DAY EVENTS

1. Create a new variable with an “initialise variable” step



Enter in a Name.

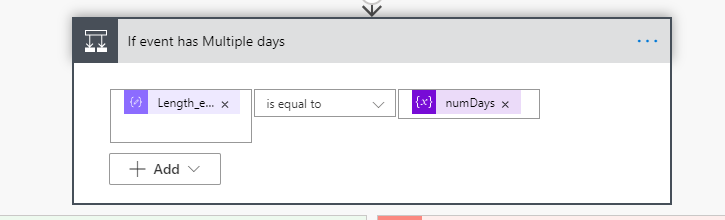
Your value and type will depend on what your form returns, in some cases it may just return ‘Yes’ so you make your Type “String” and your Value “yes”

For some forms it may be a number that is returned. For numbers make the Type an “Integer” and the Value “1”

1. Add a new conditional control step

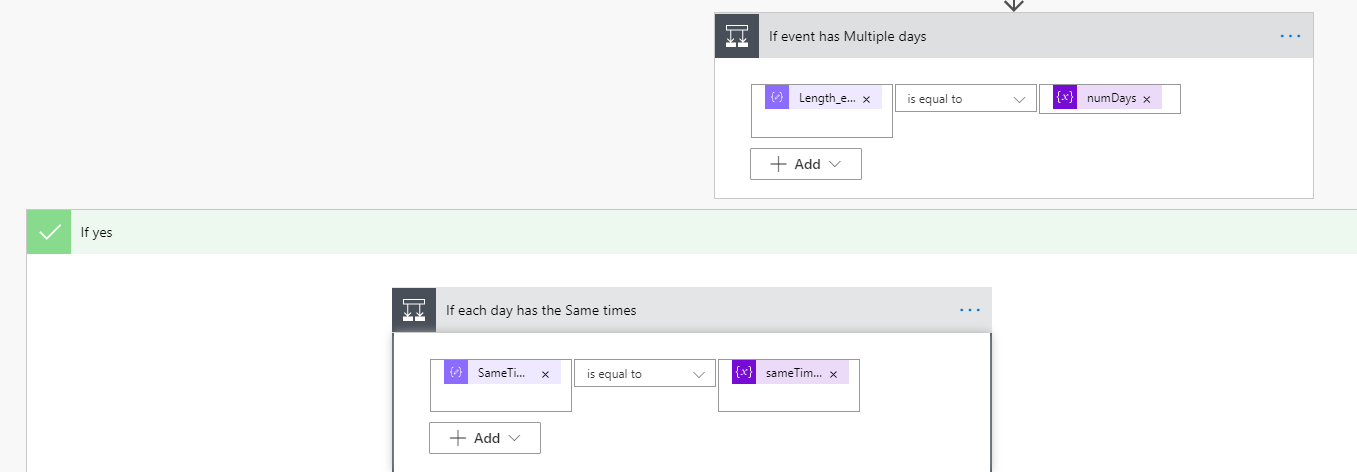
The dynamic content (LHS) will be what the form has returned to say there are multiple day events

The RHS is your newly created variable.



You may need to change the middle box based upon your variable’s type and the dynamic content.

1. If your form has the option for each date to start at the same times, repeat steps to 25-26 with the variable and dynamic content for “same times”. If not go back to step 16 to fill out the ‘yes section’ and then come back to step 28 to fill out the “No” Section. Remember to change the dynamic content in create item accordingly.



1. For same times in the yes section create a new step called ‘apply to each’ with the array that stores the dates and times in your JSON entered into the select an output box.

If your form has the same times condition repeat this for the no section as well. Note the dynamic content may be different

Once done continue on with step 16. Remember to change the dynamic content in create item accordingly.

If you have same times completed or if it is not applicable go to the “no section” of multiple days and go back to step 16. Remember to change the dynamic content in create item accordingly.

For same times and multiple days your final flow should look like the following two images

