



# POWER APP IMAGE TO SPO

Frustrated, struggling with Power Apps

## Abstract

Building a Power App image to SharePoint app isn't easy. In fact, it's frustrating. Until you know how. This paper will show step by step the how. Minimal text, just the right number of images to remove ambiguities all designed to get building your Power App in the minimum about of time.

Christopher Bird  
[info@cloudhound.uk](mailto:info@cloudhound.uk)



## Contents

Power Apps Binary to SharePoint.....	1
Exercise 1: Building your app .....	1
Create Power App Image capture.....	2
Exercise 2: Build the Flow .....	3
Postman, test your Flow .....	8
Review your Flow.....	13
Move image to SharePoint .....	14
SharePoint.....	20
Exercise 3: Wire up Flow to Power App.....	21
Flow Custom Connector.....	26
Wire Custom Connector to Power Apps .....	30
<b>Summary.....</b>	<b>35</b>
References: .....	36

## Power Apps Binary to SharePoint.

---

**Lab Time:** 60 minutes

**Lab Overview:** This lab covers how to get up and running with PowerApps by creating a new simple Power App to capture a picture and submit the image to SharePoint. This training shows you how to build an image Power App. How to construct a Flow. How to customise a template Swagger file. How to build a custom connector. Then how to wire up Power App, Flow custom connector to SharePoint.

---

### Exercise 1: Building your app

Power apps has two image controls.

1. Camera
2. Add picture

**Option one**, uses a device built in camera. However, the image will be of a lower quality. The reason the image data utilises URL data format.

**Option two**, enables you to select a picture stored on your device alternatively the devices camera. This option enables you to utilise binary data format. What this means the camera device high quality choices are available to you.

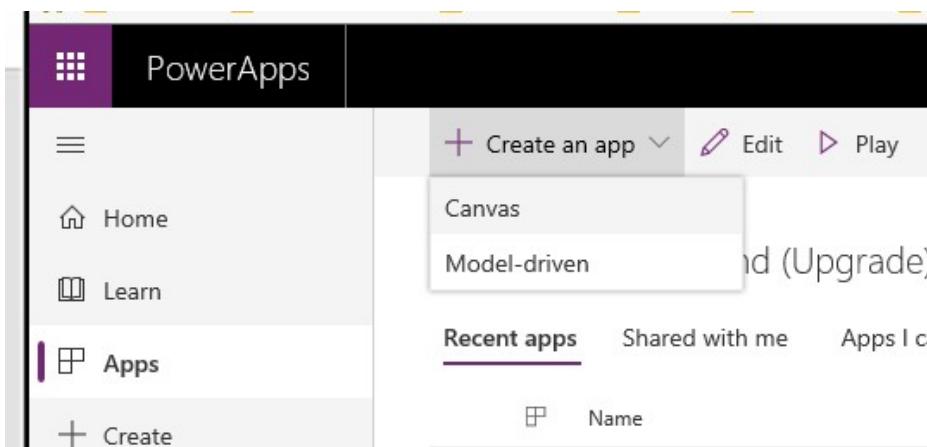


Option two is the most popular choice. Choosing 'add picture' requires Flow. The reason Power Apps doesn't have any methods for transforming one data format into another. Flow enables data collection transformation and the passing on of data.

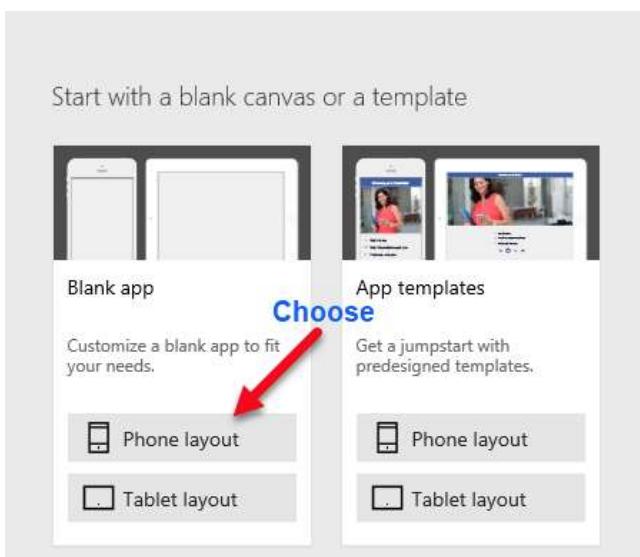
Create Power App Image capture.

Open Microsoft Power App click [here...](#)

Create a Canvas Power App see *image below*.

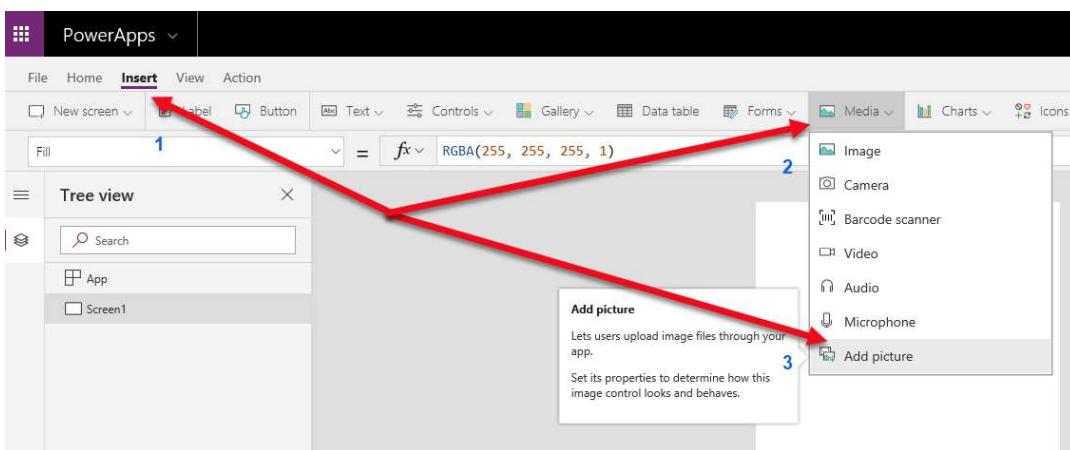


Choose phone layout.

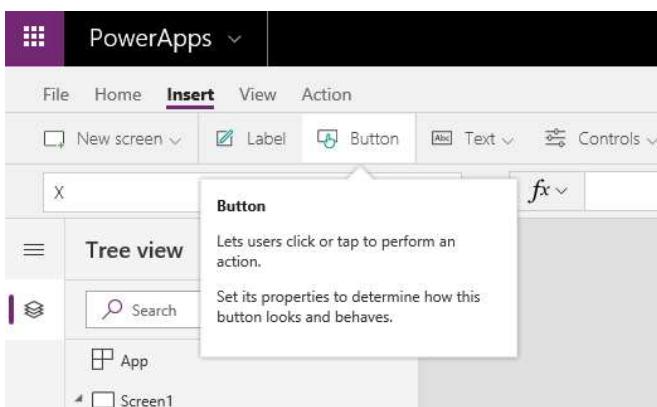


Select **Insert>Media>Add Picture**

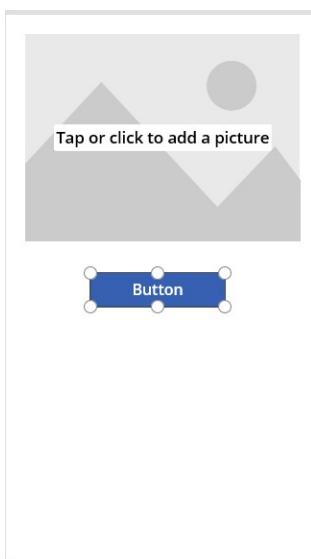
*See image below.*



Next select and add a button *see image below.*

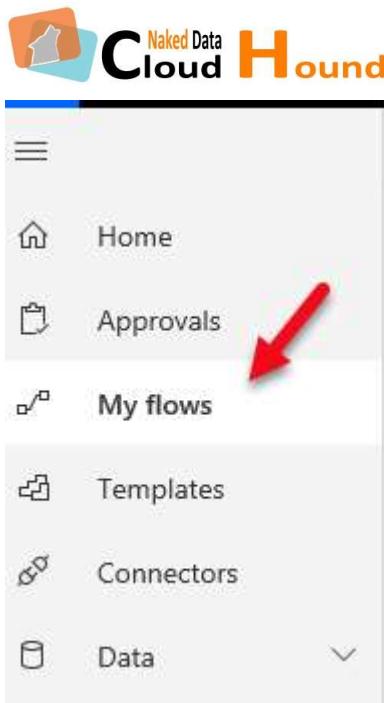


Reposition the button as shown in the *image below.*

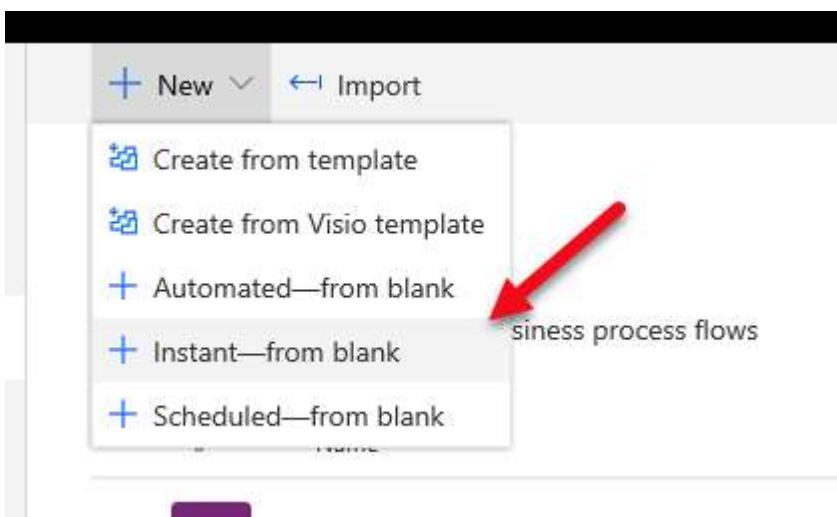


## Exercise 2: Build the Flow

Open Flow click [here...](#) and choose 'My Flows'



Under the 'New' select 'Instant—from blank'



Wizard pop-up appears.

1. Name your flow.
2. Select **Flow**.

Click '**Create**' button, see *image below*.



Build an instant flow

Flow name 1 APPFlowSPO

Choose how to trigger this flow \*

From Microsoft Flow Microsoft Flow (i)

From PowerApps PowerApp (i)

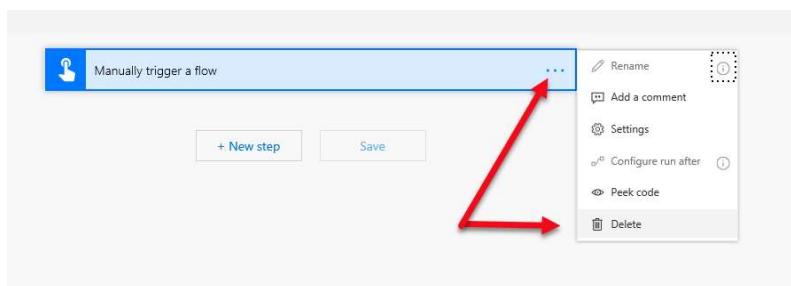
Triggered manually from any device, easy-to-share instant flows automate tasks so you don't have to repeat yourself.

Examples:

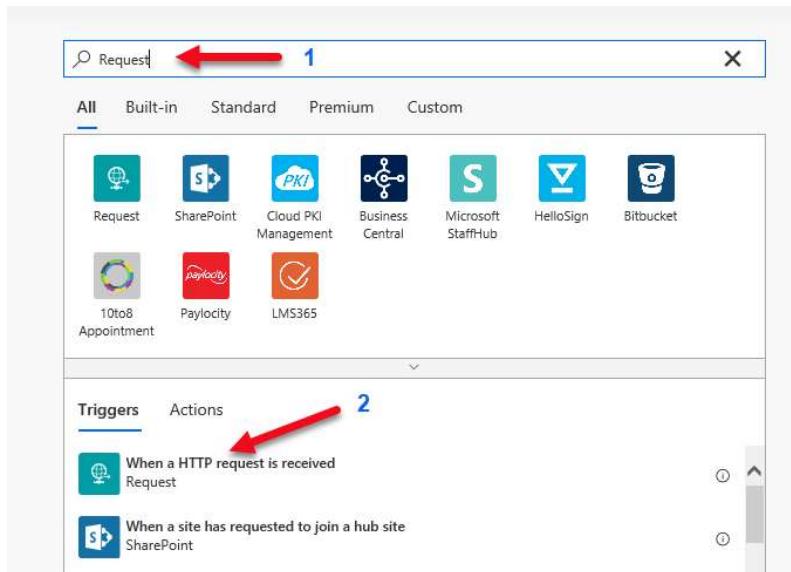
- Get an automatic mobile alert whenever a VIP client emails you
- Save all your email attachments to a folder automatically

Skip Create Cancel

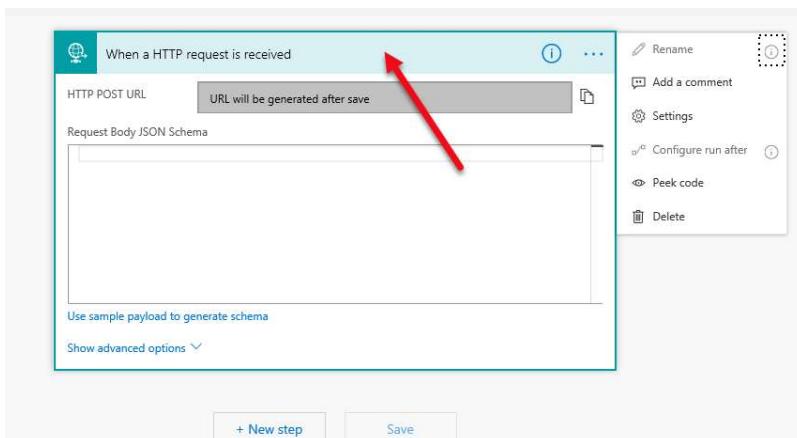
Click the three dots (...) and from sub menu select delete.



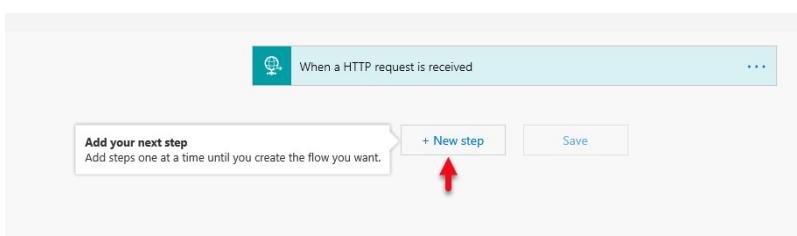
A new window option opens. Type 'request' and select 'When a HTTP request is received.'



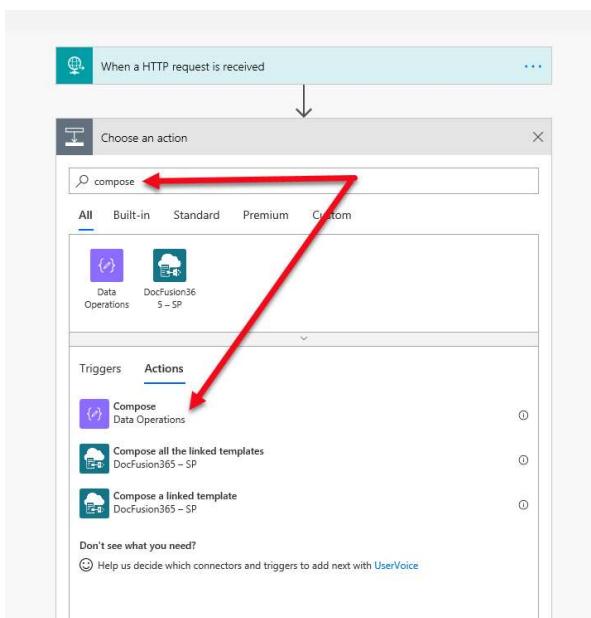
Click the green bar. This collapses HTTP request see image below.



Click the '**+ New step**' button.



Type '**compose**' and in the options below click '**Compose Data Operations.**'



To forward an image from Power Apps we need to send binary data to SharePoint. To achieve this, we use a '**triggerMultipartBody**' expression.

1. Place your mouse icon in the '**Inputs**' box.
2. Select Expression from the sub menu options.
3. In the **Fx** (function) box type trigger.
4. Select '**triggerMultipartBody**'.

*See image below.*



The screenshot shows a Microsoft Flow Designer interface. It starts with a 'When a HTTP request is received' trigger. Below it is a 'Compose' step. In the 'Compose' step, there is an input field containing 'triggerMultipartBody(0)'. A red arrow points from this input field to a 'Dynamic content' pane. The 'Dynamic content' pane has tabs for 'Expression' and 'OK'. The 'Expression' tab is selected, showing the expression 'triggerMultipartBody(0)'. A red arrow points from this tab to the 'triggerMultipartBody' function in the list of functions. Another red arrow points from the 'triggerMultipartBody' function to its parameter '(0)'. The 'OK' button is also highlighted with a red arrow.

**'triggerMultipartBody'** requires an index<sup>1</sup>. Type an open bracket and type zero and close bracket  
**'triggerMultipartBody(0)'** then OK see *image below*.

This screenshot shows a 'Dynamic content' dialog box with the 'Expression' tab selected. The input field contains the expression 'triggerMultipartBody(0)'. The 'OK' button at the bottom is highlighted with a red arrow.

You should *see the below*.

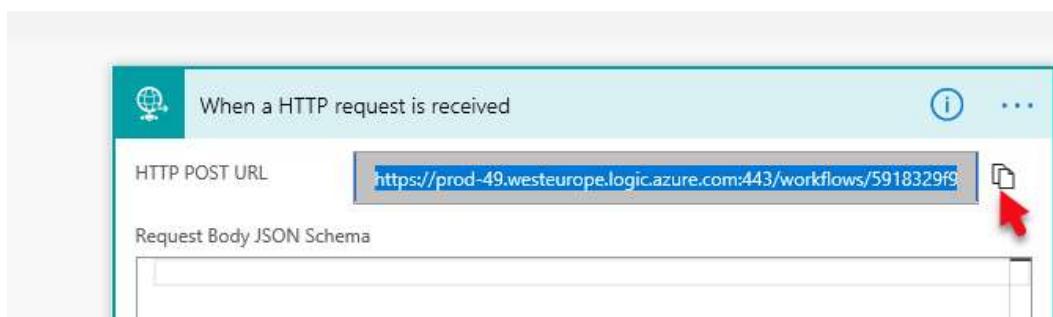
This screenshot shows the Microsoft Flow Designer with the 'Compose' step completed. The input field now contains the expression 'triggerMultipartBody(0)'. To the right of the main canvas, there is a sidebar titled 'Add dynamic content from the apps and connectors used in this flow.' It lists 'Body', 'Headers', and 'Path Parameters'.

Save your Flow.

This screenshot shows the Microsoft Flow Designer toolbar. A large red arrow points to the 'Save' button, which is highlighted with a red arrow. Other buttons in the toolbar include 'Flow Checker' and 'Test'.

Click the green bar to expand '**Http Request**' and copy the URL see *image below*.

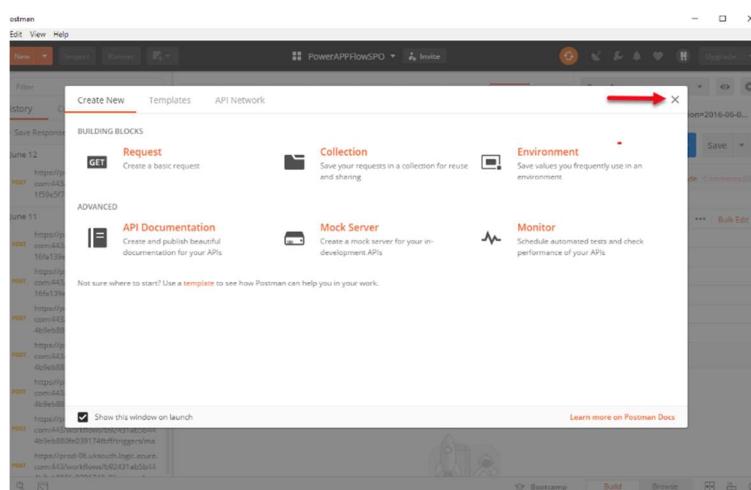
<sup>1</sup> In this lab exercise, a single image, is uploaded to SharePoint. If a gallery (multiple images) were to be uploaded then index identifies each image for example 0, 1, 2,3,4 identifies five images.



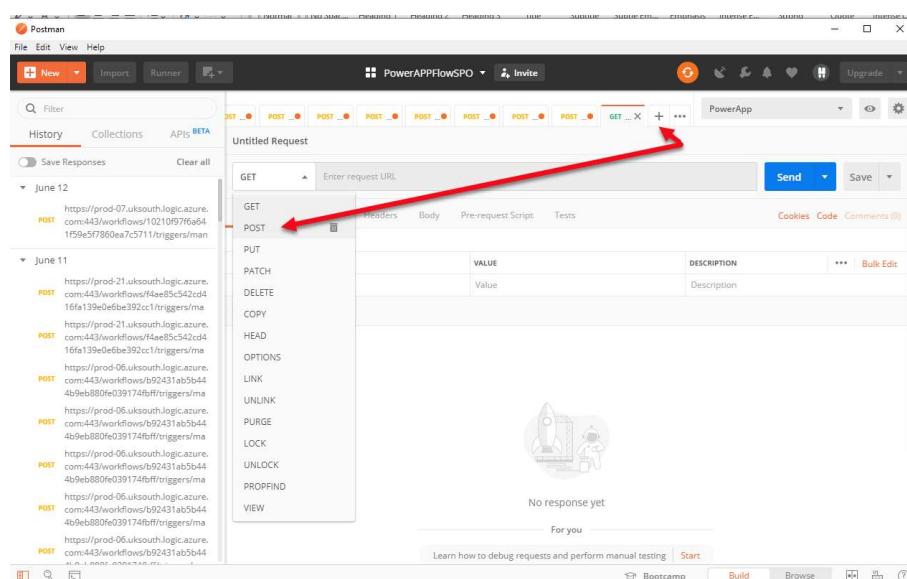
Next you need to test your flow returns the body. To achieve this you require, an application called **Postman**. It's a developer's application. If you don't have this already installed you can download it from [here...](#)

## Postman, test your Flow

When 'Postman' opens close the splash menu.



If you've used '**Postman**' before, select a new tab. In the '**Get**' drop down, select '**Post**'.



1. Check you have '**Post**' selected.
2. Past your **Flow Request URL**.



3. Select 'Body.'
4. Choose 'form-data.'
5. In the 'Key' row click the drop down and select File.

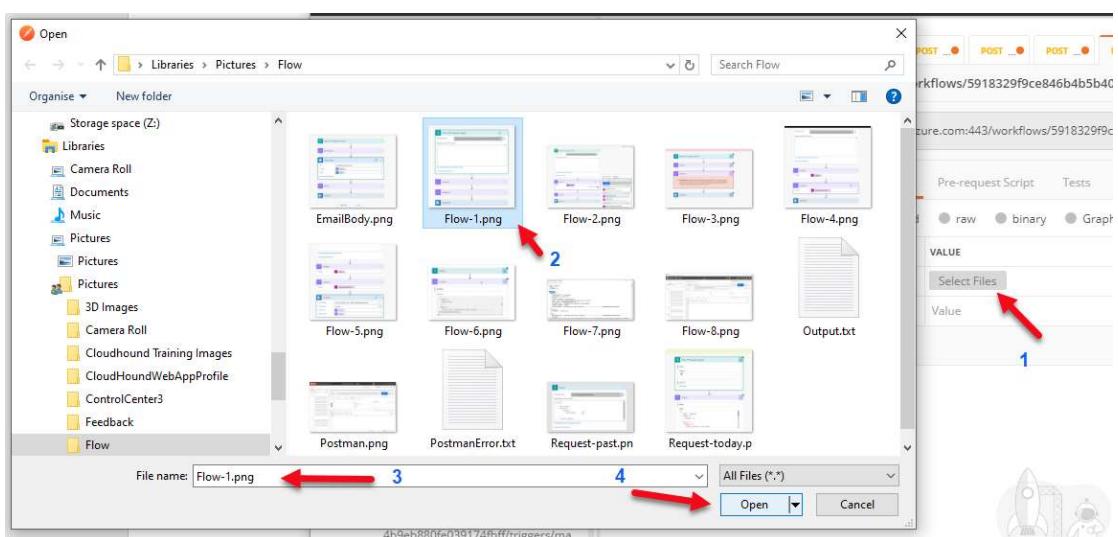
See image below.

In the adjoining 'key' column row select column 'Value' a grey box appears. Click to select. Next navigate to your image folder and choose a picture. Select a small image file<sup>2</sup>. If you choose a large image size the test will work but it will create a long string, and this can make debugging problematical. At this stage of your learning we want simplicity and to establish your success.

1. Select files

2. Choose an image
3. Check chosen image is selected
4. Click 'Open' see image below.

<sup>2</sup> If you select a large image file debugging may prove to be somewhat problematical.



Confirm '**Post**', Flow URL, '**Body**', '**Form-data**', key value 'file', are set. Check value '**Select files**' has your chosen image.

KEY	VALUE
Flow-1.png	Flow-1.png
Key	Value

1. Click send button (*see image below*).
2. You should see **202 Accepted**. (*if you see 202 Accepted you've been successful*).



The screenshot shows the Postman interface with a history of API calls. A specific POST request to a Logic App trigger is selected. The 'Body' tab shows a file named 'Flow-1.png' attached. The status bar at the bottom right shows 'Status: 202 Accepted'. Red arrows point to both the 'Send' button and the status bar.

Return to **Flow** by clicking the blue arrow see *image below*.

The screenshot shows the Microsoft Flow designer with a trigger step for 'When a HTTP request is received' from an 'HTTP POST URL'. The 'Request Body JSON Schema' field is visible. A red arrow points to the back arrow in the top left corner of the designer interface.

Now you should view **Flow run history**. You should see '**Start**' and '**Duration**' time and '**Succeeded**'. See *image below*.

Congratulations.



Details		Edit
Flow	APPFlowSPO	Status
Owner	CHRISTOPHER BIRD	On
		Created
		Jun 28 03:52 PM
		Modified
		Jun 28 03:52 PM
		Type
		Automated
Runs		All runs
Start	Duration	Status
Jun 28 05:03 PM (2 min ago)	00:00:00	Succeeded

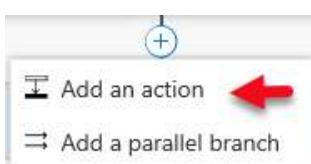
Click 'Start' to review success run.

Runs	All runs	
Start	Duration	Status
Jun 28 05:03 PM (17 h ago)	00:00:00	Succeeded

Expand the HTTP request and in ‘**Outputs**’ section you should see “**body**” this confirms image data capture see *image below*.

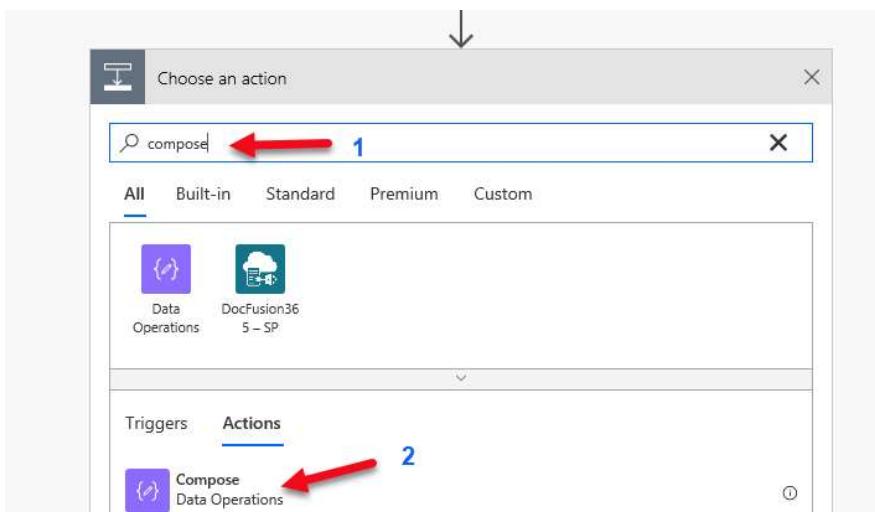
Expand the **Compose**. Basically, the image is wrapped in a Json object.

Next add another ‘Compose.’ Click the blue plus sign (+).



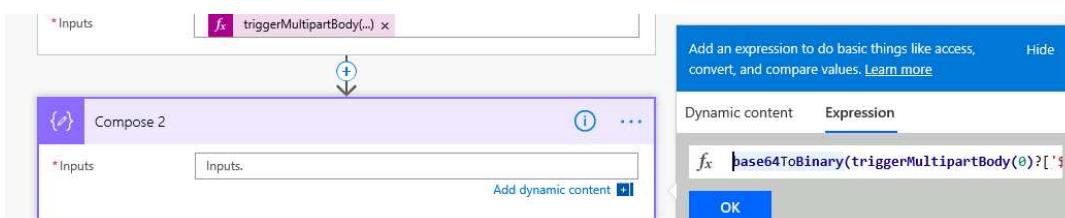
In the 'Choose and action' box

1. Type compose.
2. Click 'Compose Data Operations.'



Place your mouse cursor in the 'Inputs' box. Select the 'Expression' tab and type **base64ToBinary(triggerMultipartBody(0)?['content'])<sup>3</sup>**

Click 'OK.'



Review your Flow.

Your Flow should see the same as shown in the *image below*.

---

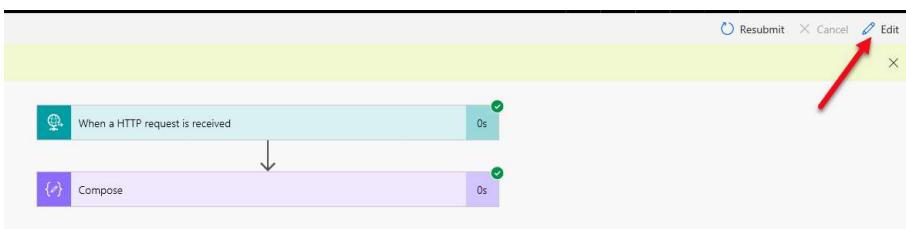
<sup>3</sup> **Don't copy and paste the code.** If you do, then paste into '**NotePad**' then copy and paste that. Doing so minimises transcription errors. Characters, quote marks, can vary between applications. This may cause you considerable frustration if you need to debug. I recommend you type the code. Doing so will help to build keyboard finger memory and this will help you when you writer your own '**Expressions**'.



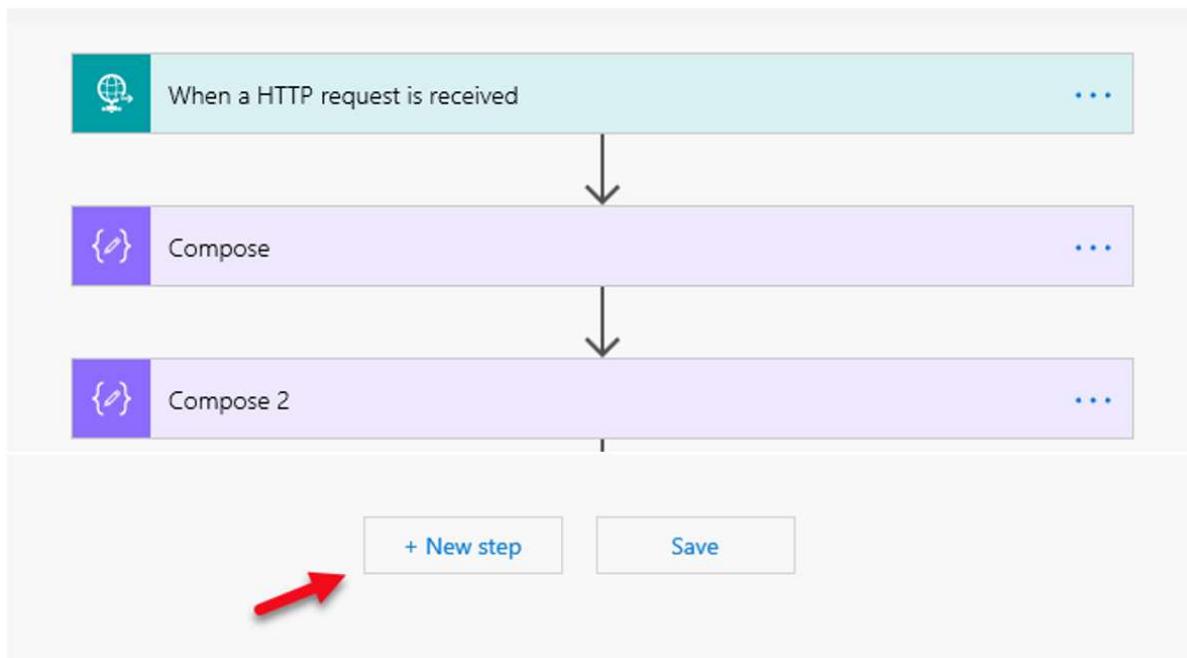
The second '**compose**' step converts the first '**compose**' to base 64 binaries. This step enables your image to be stored in SharePoint.

### Move image to SharePoint

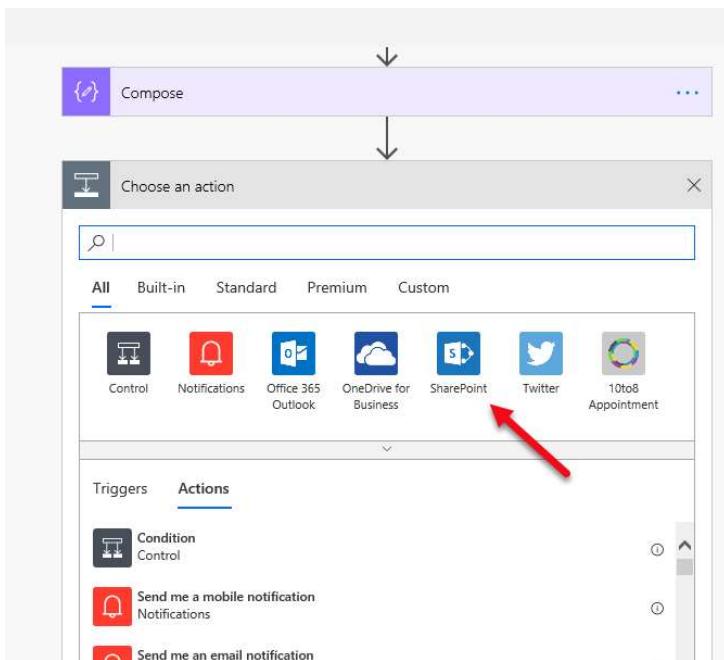
Click '**Edit**' see image below.



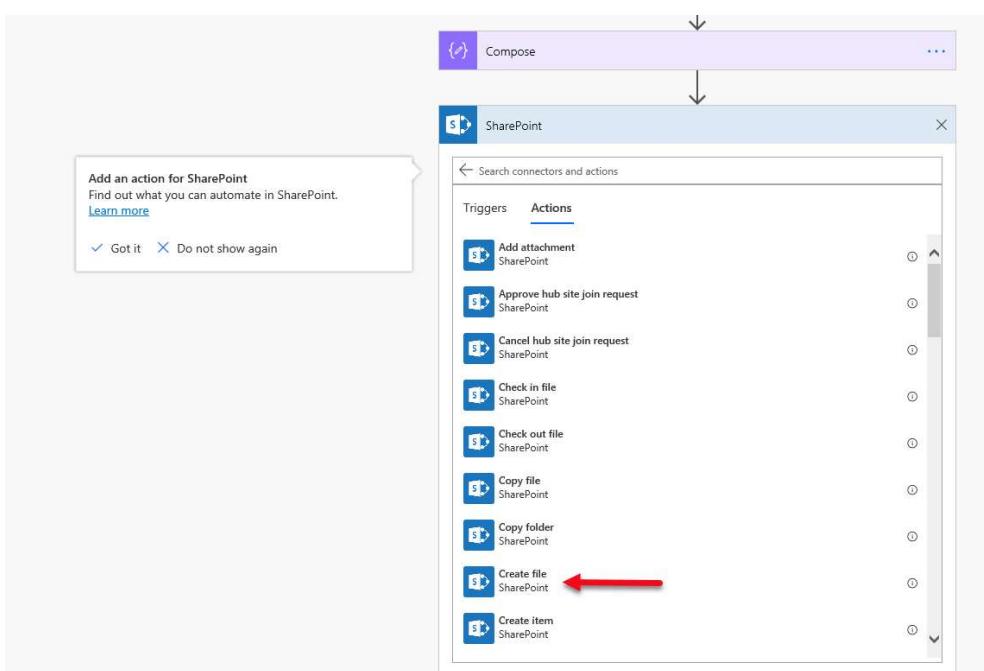
Click '**New Step**'



Select '**SharePoint**' see image below.

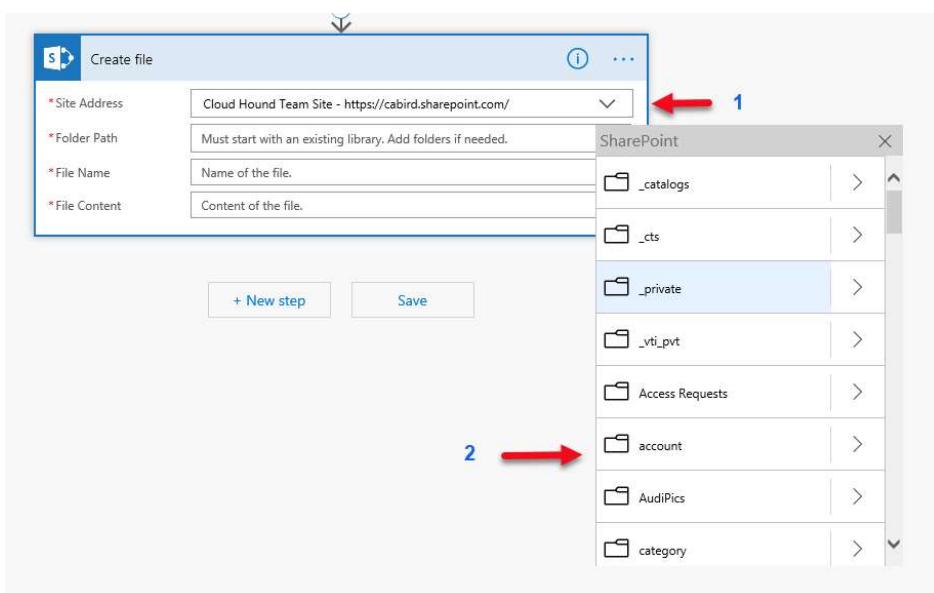


Select '**Create a file**' see image below.



1. '**Site Address**' click the drop down and select your SharePoint Online URL.
2. '**Folder Path**' click (*the folder icon – right hand side*) to reveal submenu and scroll down to select you're document library.

*See image below.*

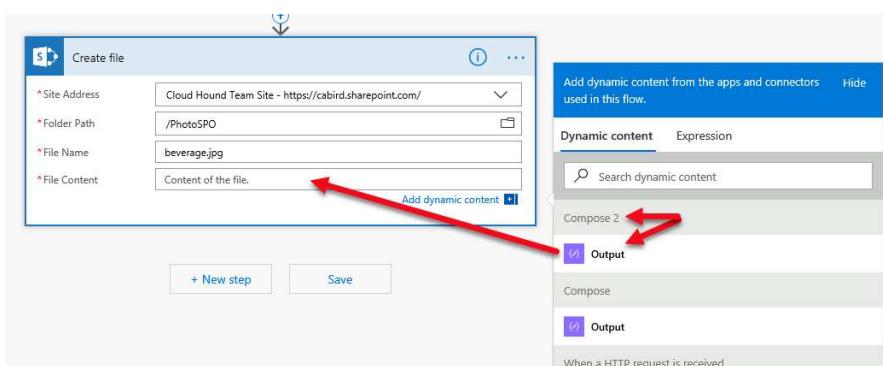


Your SharePoint library destination doesn't require anything special for this exercise. **Name** and defaults are acceptable *see image below*.

### PhotoSPO

	Name	Type Of Animal	Picture Size	File Size	Modified	+ Add column
	jump.jpg		170 x 189	8.83 KB	Yesterday at 09:42	

For '**File Content**' select '**Compose 2**'<sup>4</sup> '**Output.**'



In the next image below note '**Compose 2**' I've added a comment. Useful step in debugging and understanding the flow process *see image below*.

<sup>4</sup> Best practice for a production Flow – is to name all your Flow '**Steps**' to something more meaningful. If you or someone else return to the Flow at a later date a descriptive title may help you or someone else to better understand the Flow logic process.

Compose 2

```
base64ToBinary(triggerMultipartBody(0)?['$content'])
```

\* Inputs  base64ToBinary(...)

Create file

\* Site Address: Cloud Hound Team Site - https://cabird.sharepoint.com/

\* Folder Path: /PhotoSPO

\* File Name: beverage.jpg

\* File Content:

+ New step

Now let's test this.

**SAVE** your Flow then return to Postman and click '**Send**' button *see image below*.

Postman

History Collections APIs BETA

POST https://prod-49.westeurope.logic.azure.com:443/workflows/5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke?api-version=2016-08-01&sp=RunFlow

Params Authorization Headers (9) Body Pre-request Script Tests

Body (form-data)

KEY	VALUE	DESCRIPTION
beverage.jpg		

Send Save

Assuming all goes well you should see '**Status 202 Accepted**' *see image below*.

Postman

History Collections APIs BETA

POST https://prod-49.westeurope.logic.azure.com:443/workflows/5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke?api-version=2016-08-01&sp=RunFlow

Params Authorization Headers (9) Body Pre-request Script Tests

Body (form-data)

KEY	VALUE	DESCRIPTION
beverage.jpg		

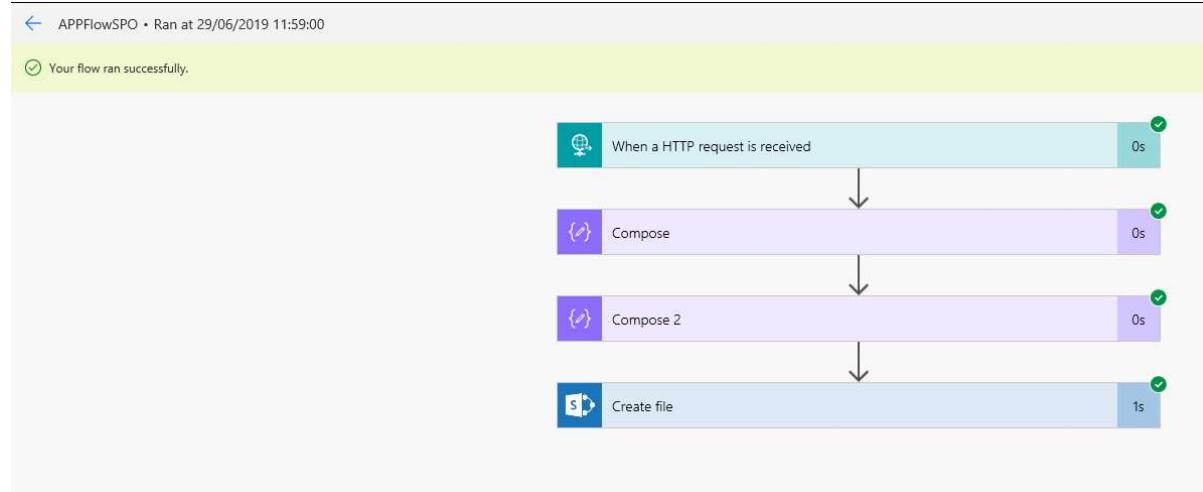
Status: 202 Accepted Time: 997 ms Size: 1.08 KB

Return to your **Flow** and view **Flow Runs** and click the '**Runs**' just run (Jun 29 11:59 AM (55 sec ago)) *see image below*.



Runs			All runs
Start	Duration	Status	
Jun 29 11:59 AM (55 sec ago)	00:00:01	Succeeded	
Jun 29 11:55 AM (4 min ago)	00:00:00	Succeeded	
Jun 29 11:02 AM (57 min ago)	00:00:00	Succeeded	
Jun 28 05:03 PM (18 h ago)	00:00:00	Succeeded	

You should see the image below.



Expand your Flow steps and view your marvellous achievement.

1. ‘HTTP’ request image received from Power Apps (Green).
2. First ‘Compose’ image Json string format (Green).
3. ‘Compose2’ converts to Binary (Green).
4. ‘Create file’ image sent to SharePoint (Green).

See expanded images below.

When a HTTP request is received

0s

INPUTS

Schema

{} (empty)

OUTPUTS

Body

```
{
  "$content-type": "multipart/form-data; boundary=-----",
  "$content": "LS0tLS0tLS0tLS0tLS0tLS0tLS0tLTE3NjEyNzIxM",
  "$multipart": [
    {
      "headers": {
        "Content-Disposition": "form-data; name=\"\"; filename=\"\"",
        "Content-Type": "image/png"
      }
    }
  ]
}
```

Show more ▾



Compose 2

Os

**INPUTS**

Inputs

```
{"$content-type": "application/octet-stream",  
 "$content": "/9j/4RddRXhpZgAAUT0AKgAAAAGABwESAAMAAAABAAEAAAEEaA/
```

**OUTPUTS**

Outputs

```
{"$content-type": "application/octet-stream",  
 "$content": "/9j/4RddRXhpZgAAUT0AKgAAAAGABwESAAMAAAABAAEAAAEEaA/
```



**Naked Data**  
**Cloud Hound**

Create file 1s

**INPUTS**

Site Address  
https://cabird.sharepoint.com/

Folder Path  
/PhotoSPO

File Name  
beverage.jpg

File Content

```
  "$content-type": "application/octet-stream",
  "$content": "/9j/4RddRXhpZgAATU0AKgAAAAgABwESAAMAAAABAAEAAAEEaAA
```

**OUTPUTS**

ItemId  
6

Id  
%252fPhotoSPO%252fbeverage.jpg

Name  
beverage.jpg

DisplayName  
beverage.jpg

Path  
/PhotoSPO/beverage.jpg

LastModified  
2019-06-29T10:59:02Z

Size  
64156

MediaType  
image/jpeg

IsFolder  
false

ETag  
"72342802-E9C3-4448-A28A-9F4F3D77F9B9",1"

FileLocator  
atasset=aHR0cHM6Ly9jYWJpcmQuC2hhcmVwb2ludC5jb20=,id=JTI1MmZQaG90b

SharePoint

Open your SharePoint library.

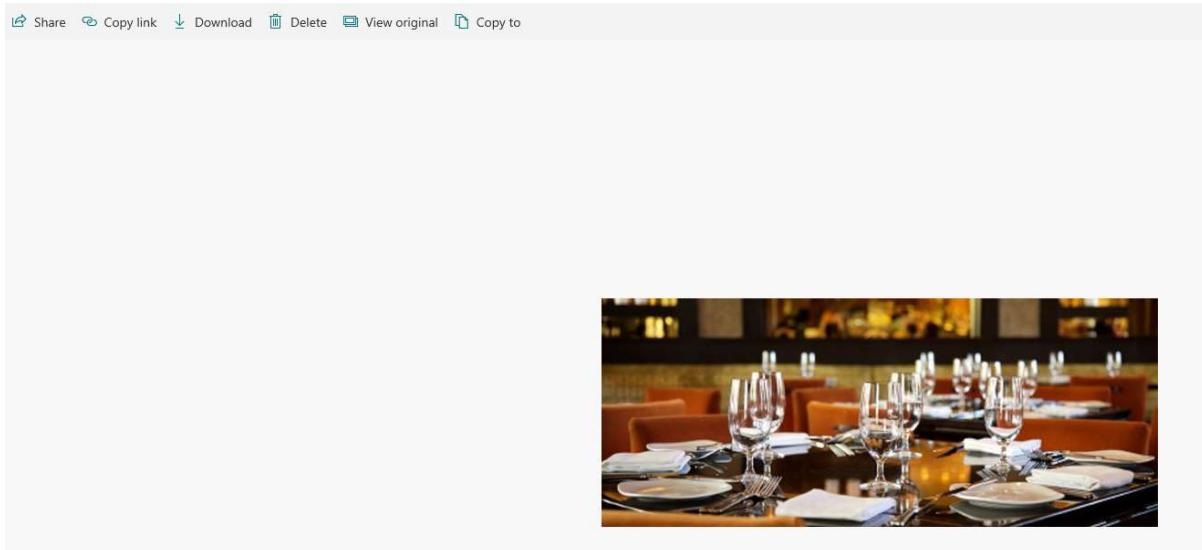
Select your image *see image below.*



## PhotoSPO

Name	Type Of Animal	Picture Size	File Size	Modified	+ Add column
beverage.jpg		610 x 250	62.7 KB	A few seconds ago	
jump.jpg		170 x 189	8.83 KB	Yesterday at 09:42	

Your image opens for you to view your uploaded image *see image below*.



### Exercise 3: Wire up Flow to Power App.

To enable Power Apps to utilise the **Flow** we need a Swagger file *see image below*<sup>5</sup>.

```

{
  "swagger": "2.0",
  "info": {
    "description": "Upload a Photo to SharePoint",
    "version": "1.0.0",
    "title": "photohandler"
  },
  "host": "prod-a9.ulcsouth.logic.azure.com",
  "basePath": "/workflows",
  "schemes": [
    "https"
  ],
  "consumes": [],
  "produces": [],
  "paths": {
    "/d9fb9990c34dd58abdc58b7be539f6/triggers/manual/paths/invoke": {
      "post": {
        "summary": "Upload a Photo to SharePoint",
        "description": "Upload a Photo to SharePoint",
        "operationId": "UploadPhoto",
        "consumes": [
          "multipart/form-data"
        ],
        "parameters": [
          {
            "name": "api-version",
            "in": "query",
            "default": "2016-06-01",
            "required": true,
            "x-ms-visibility": "internal",
            "type": "string"
          },
          {
            "name": "sp",
            "in": "query",
            "default": "triggers/manual/run",
            "required": true,
            "x-ms-visibility": "internal",
            "type": "string"
          },
          {
            "name": "sr",
            "in": "query",
            "default": "1.0",
            "required": true,
            "x-ms-visibility": "internal",
            "type": "string"
          }
        ]
      }
    }
  }
}

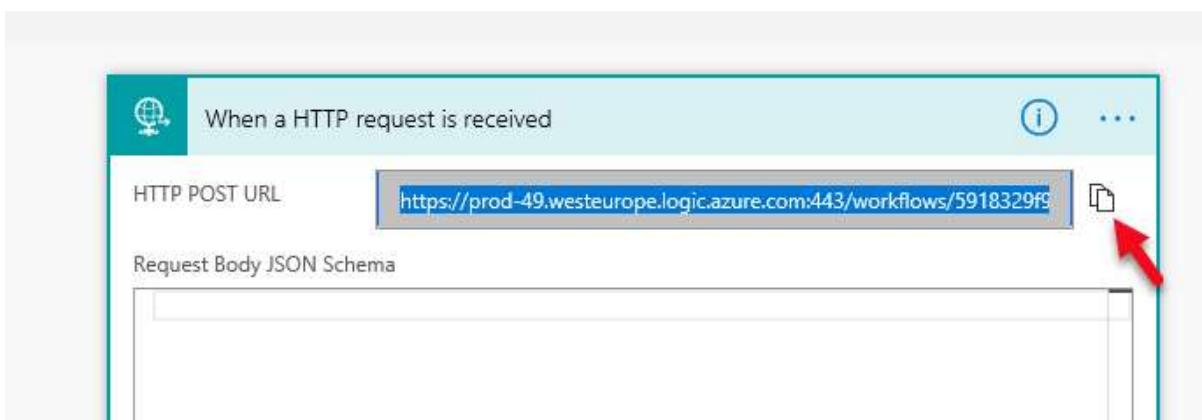
```

<sup>5</sup> Template swagger file click [here...](#) To make the template swagger file work for you will need to use your Flow URL to change some entries.



Using **Microsoft Visual Studio Code** click [here...](#) to download free. You could use **Notepad++** click [here...](#) or any other developer code application for example **Microsoft Code Writer** click [here...](#)

Take a look at the *image below*. Copy YOUR ‘Flow URL’ see *image below*.



Using your copied ‘Flow URL’ paste into your code editor of choice as shown below. This is for referencing and copying purposes and **MUST** be deleted prior to deploying your swagger file.

1. Check “[https](#)”
2. The “[host](#)”: is changed to “[prod-49.westeurope.logic.azure.com](#)” **Flow URL**.
3. Having implemented the above two steps delete [https://prod-49.westeurope.logic.azure.com:443](#) part of the **URL**.

*See image below.*



Your URL code should only have the following part of the **URL** left *see image below*.

/workflows/5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=fAq0xXsHOI1spJnjt7dBIUezmk8voAXq2X\_fvUXIGqE

Next check “[basePath](#)”: “[/workflows](#)”, see *image below*.



```

1  {
2    "swagger": "2.0",
3    "info": {
4      "description": "Upload a Photo to SharePoint",
5      "version": "1.0.0",
6      "title": "PhotoHandler"
7    },
8    //workflows/5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke?api-version=2016-06-01&sp=5918329f9ce846b4b5b40d3c36c9025d
9    "host": "prod-49.westeurope.logic.azure.com",
10   "basePath": "/workflows", ←
11   "schemes": [
12     "https"
13   ],
14   "consumes": [],
15   "produces": []

```

Copy 5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke and paste to “`paths`”: {  
see image below.

```

1  {
2    "swagger": "2.0",
3    "info": {
4      "description": "Upload a Photo to SharePoint",
5      "version": "1.0.0",
6      "title": "PhotoHandler"
7    },
8    //5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke?api-version=2016-06-01&sp=5918329f9ce846b4b5b40d3c36c9025d
9    "host": "prod-49.westeurope.logic.azure.com",
10   "basePath": "/workflows",
11   "schemes": [
12     "https"
13   ],
14   "consumes": [],
15   "produces": [],
16   "paths": {
17     "/f88fba9900c34dd58abdc58b7be539f6/triggers/manual/paths/invoke": {
18       "post": {
19         "summary": "Upload a Photo to SharePoint".

```

Remove `/workflows/5918329f9ce846b4b5b40d3c36c9025d/triggers/manual/paths/invoke` from the URL.

Next check the following:

1. “`name`” “`api-version`” matches
2. “`default`”: matched **URL** value in this example “`2016-06-01`”
3. “`name`”: “`sp`”
4. “`default`”: path matches “`/trigger/manual/run`”

See image below.



```

1  {
2    "swagger": "2.0",
3    "info": {
4      "description": "Upload a Photo to SharePoint",
5      "version": "1.0.0",
6      "title": "PhotoHandler"
7    },
8    //?api-version=2016-06-01&sp=%2ftriggers%2fmanual%2frun&sv=1.0&sig=fAq0xXsH0l1spJnjt7dBlUezmk8voAXq2X_fvUX1GqE
9    "host": "prod-49.west-europe.logic.azure.com",
10   "basePath": "/workflow",
11   "schemes": [
12     "https"
13   ],
14   "consumes": [],
15   "produces": []
16   "paths": {
17     "/f88fba9900c4dd58abd58b7be539f6/triggers/manual/paths/_invoke": {
18       "post": {
19         "summary": "Upload a Photo to SharePoint",
20         "description": "Upload a Photo to SharePoint",
21         "operationId": "UploadPhoto",
22         "consumes": [
23           "multipart/form-data"
24         ],
25         "parameters": [
26           {
27             "name": "api-version",
28             "in": "query",
29             "default": "2016-06-01",
30             "required": true,
31             "x-ms-visibility": "internal",
32             "type": "string"
33           },
34           {
35             "name": "sp",
36             "in": "query",
37             "default": "/triggers/manual/run",
38             "required": true,
39             "x-ms-visibility": "internal",
40             "type": "string"
41           },
42           {
43             "name": "sv",
44             "in": "query",
45             "default": "1.0",
46             "required": true,
47             "x-ms-visibility": "internal",
48             "type": "string"
49           }
50         ]
51       }
52     }
53   }
54 }
```

Next check “name”: “sv” matches and “default”: matches “1.0”

Edit the **Flow URL** to remove the above entries. You should be left with the following *see below image*.

```
//&sig=fAq0xXsH0l1spJnjt7dBlUezmk8voAXq2X_fvUX1GqE
```

Now we need to “name”: “sig” and “default”: “fAq0xXsH0l1spJnjt7dBlUezmk8voAXq2X\_fvUX1GqE”<sup>6</sup> *see image below.*

---

<sup>6</sup> fAq0xXsH0l1spJnjt7dBlUezmk8voAXq2X\_fvUX1GqE is the signature.



```

},
//&sig=fq0xXsH0l1spJnjt7dBlUezmk8voAXq2X_fvUXlggE
"host": "prod-49.westeurope.logic.azure.com",
"basePath": "/workflows",
"schemes": [
  "https"
],
"consumes": [],
"produces": [],
"paths": {
  "/f88 ba9900c34dd58abdc58b7be539f6/triggers/manual/paths/invoke": {
    "post": {
      "summary": "Upload a Photo to SharePoint",
      "description": "Upload a Photo to SharePoint",
      "operationId": "UploadPhoto",
      "consumes": [
        "multipart/form-data"
      ],
      "parameters": [
        {
          "name": "api-version",
          "in": "query",
          "default": "2016-06-01",
          "required": true,
          "x-ms-visibility": "internal",
          "type": "string"
        },
        {
          "name": "sp",
          "in": "query",
          "default": "/triggers/manual/run",
          "required": true,
          "x-ms-visibility": "internal",
          "type": "string"
        },
        {
          "name": "sv",
          "in": "query",
          "default": "1.0",
          "required": true,
          "x-ms-visibility": "internal",
          "type": "string"
        },
        {
          "name": "sig",
          "in": "query",
          "default": "fq0xXsH0l1spJnjt7dBlUezmk8voAXq2X_fvUXlggE",
          "required": true
        }
      ]
    }
  }
}

```

Now delete what's left of the comment in the Swagger file see *image below*.

```

1  {
2    "swagger": "2.0",
3    "info": {
4      "description": "Upload a Photo to SharePoint",
5      "version": "1.0.0",
6      "title": "ImageFlowSP"
7    },
8    "host": "prod-49.westeurope.logic.azure.com",
9    "basePath": "/workflows",
10   "schemes": [
11     "https"
12   ],
13   "consumes": [],
14   "produces": [],
15   "paths": {
16     "/f88 ba9900c34dd58abdc58b7be539f6/triggers/manual/paths/invoke": {
17       "post": {
18         "summary": "Upload a Photo to SharePoint",
19         "description": "Upload a Photo to SharePoint",
20         "operationId": "UploadPhoto",
21         "consumes": [
22           "multipart/form-data"
23         ],
24         "parameters": [
25           {
26             "name": "api-version",
27             "in": "query",
28             "default": "2016-06-01",
29             "required": true,
30             "x-ms-visibility": "internal",
31             "type": "string"
32           },
33           {
34             "name": "sp",
35             "in": "query",
36             "default": "/triggers/manual/run",
37             "required": true,
38             "x-ms-visibility": "internal",
39             "type": "string"
40           },
41           {
42             "name": "sv",
43             "in": "query",
44             "default": "1.0",
45             "required": true,
46             "x-ms-visibility": "internal",
47             "type": "string"
48           }
49         ]
50       }
51     }
52   }
53 }

```

Take your time over these amendments better to get it right.



## Save the file.

**Note:** You will need the json file to create a custom connector so save the file where you can easily retrieve the json file.

## Flow Custom Connector

Return to **Flow** and click the **gear** icon and in the drop-down submenu click '**Custom Connectors**' see image below.

Click '**Create custom connector**' in the drop menu select '**Import and Open API file.**'

A wizard window opens.

1. Name the **custom connector**
2. Import your **swagger file** you created *see the below images.*

Click '**Continue**' button.



You should see entries from your **swagger** file see *image below*.

**General information**

Add an icon and short description to your custom connector. Your host and base URL will be automatically generated from the swagger file.

Icon background color:

Description: Upload a Photo to SharePoint

Connect via on-premises data gateway [Learn more](#)

Scheme \*:  HTTPS  HTTP

Host \*: prod-49.westeurope.logic.azure.com

Base URL: /workflows

Click '**security**' at the bottom.

**General** > **2. Security** > **3. Definition** > **4. Test**

to your custom connector. Your host and base URL will be automatically generated from the swagger file.

Icon background color:

Description: Upload a Photo to SharePoint

Connect via on-premises data gateway [Learn more](#)

Scheme \*:  HTTPS  HTTP

Host \*: prod-49.westeurope.logic.azure.com

Base URL: /workflows

Security →

Leave **security** with defaults and click '**Definition**'.

**CONNECTOR NAME**: ImageFlowSPO

**1. General** > **2. Security** > **3. Definition** > **4. Test**

Create connector  Can

**Security**

Choose the authentication type and fill in the required fields to set the security for your custom connector. [Learn more](#)

**Authentication type**

Choose what authentication is implemented by your API \*

No authentication

[Edit](#)

[General](#) Definition →

The '**General**' can be left with the defaults taken from the **swagger** file.

The request section you need to confirm verb is '**Post**', **api-version**, **sp**, **sv**, **sig** all have a red asterisk **\***.

*See images below.*



1. General > 2. Security > 3. Definition > 4. Test

**Actions (1)**  
Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

- UploadPhoto**
- New action**

**Triggers (0)**  
Triggers read data in from your connector. A trigger focuses on a particular event that happens, say a new Contact or Order being created and provides the relevant data so that users can take action on that event.

- New trigger**

**References (0)**  
References are reusable parameters used by both actions and triggers.

**Policies (0)**

**Create connector**

**Request**

Verb \*  
The verb describes the operations available on a single path.  
**POST** ←

URL \*  
This is the request URL.  
<https://prod-49.westeurope.logic.azure.com/workflows/f88fba990c34dd58abdc58b7be539f6/t>

Path  
Path is used together with Path Templating, where the parameter value is actually part of the operation's URL.

Query  
Query parameters are appended to the URL. For example, in /items?id=####, the query parameter is id.

\* api-version ... \* sp ... \* sv ... \* sig ...

Headers  
These are custom headers that are part of the request.

Body  
The body is the payload that's appended to the HTTP request. There can only be one body parameter.

If the \* is missing, click the three dots and click edit. Even if \* is present you should select edit to check all four query strings.

For query strings **api-version**, **sp**, **sv** and **sig** check the following.

**Note:** You shouldn't have to make any amendments as these were configured in your swagger file nevertheless it's worth checking.

Check '**Is required?**' radio button is set to '**Yes**' and '**Visibility**' radio button is set to '**Internal**' see images below. Setting the query string to '**internal**' ensures the default value will be used but **Power Apps** and **Flow** will not show these parameters.

**Query**

Query parameters are appended to the URL. For example, in /items?id=####, the query pa

\* api-version ... \* sp ... \* sv ... \* sig ...

**Edit** ←

**Delete**



1. General > 2. Security > **3. Definition** > 4. Test

Create connector

**Parameter**

Name \*

Description [Learn more](#)

Summary [Learn more](#)

Default value

Is required?  Yes  No

Visibility [Learn more](#)  none  advanced  internal  important

Location \*  Path  Query  Header  Body

Type  Format

**Actions (1)**  
Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

**Triggers (0)**  
Triggers read data in from your connector. A trigger focuses on a particular event that happens, say a new Contact or Order being created and provides the relevant data so that users can take action on that event.

**References (0)**  
References are reusable parameters used by both actions and triggers.

**Policies (0)**

To return to edit and or check the next query string click the **back** arrow see *image below*.

**Back**

**Parameter**

Name \*

Description [Learn more](#)

Summary [Learn more](#)

Default value

Is required?  Yes  No

Visibility [Learn more](#)  none  advanced  internal  important

**Actions (1)**  
Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

**Triggers (0)**  
Triggers read data in from your connector. A trigger focuses on a particular event that happens, say a new Contact or Order being created and provides the relevant data so that users can take action on that event.

**References (0)**  
References are reusable

Having confirmed settings are correct click '**Create Connector.**'

**Connector Name** ImageFlowSPO

1. General > 2. Security > **3. Definition** > 4. Test

Create connector

**Actions (1)**  
Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

**General**

Summary [Learn more](#)

You should see the below once '**custom connector**' has been created.



## Response

It defines the shape of response returned by the underlying connector when making the request.

200 successful operation

[+ Add default response](#)

## Validation

This helps you identify potential issues with this action.

## Validation

Validation succeeded.

[← Security](#)

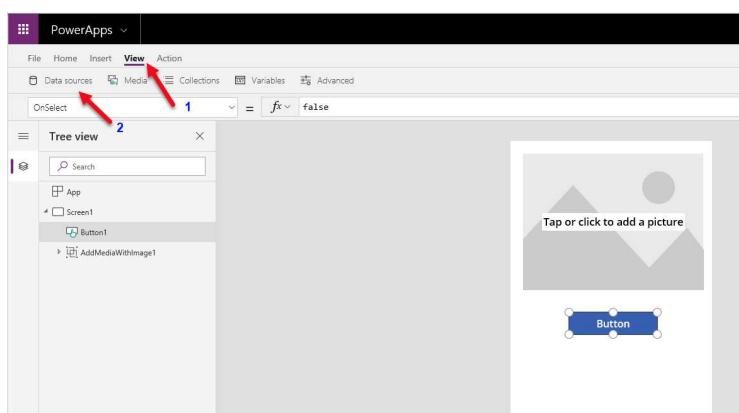
[Test →](#)

## Wire Custom Connector to Power Apps

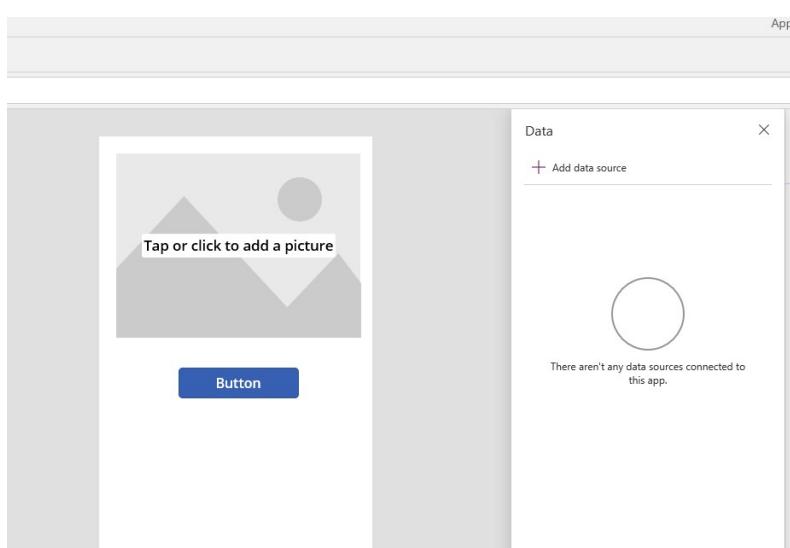
[Return to Power Apps.](#)

We need to add the data source.

1. Click 'View'
2. Below in the ribbon click 'Data Sources'

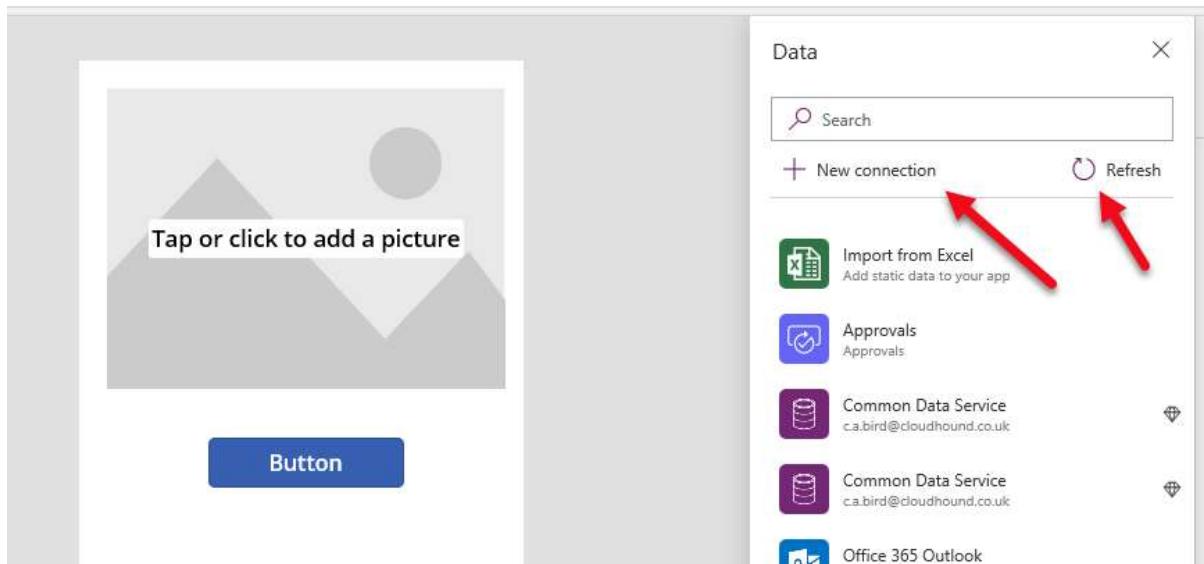


Next click 'Add data source.'

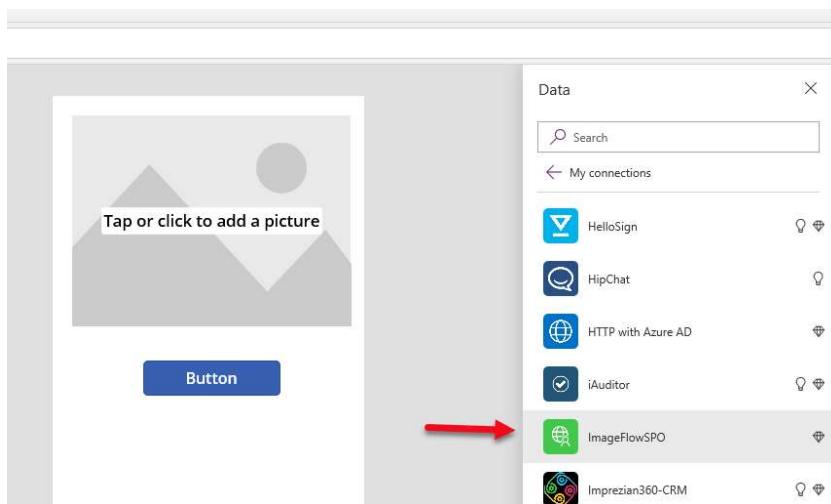




Assuming you have previous connections, you will see these listed. If you don't see all the connections, you made click refresh. Almost certainly you will need to click the '**New Connection**' option to add your newly built connection see image below.

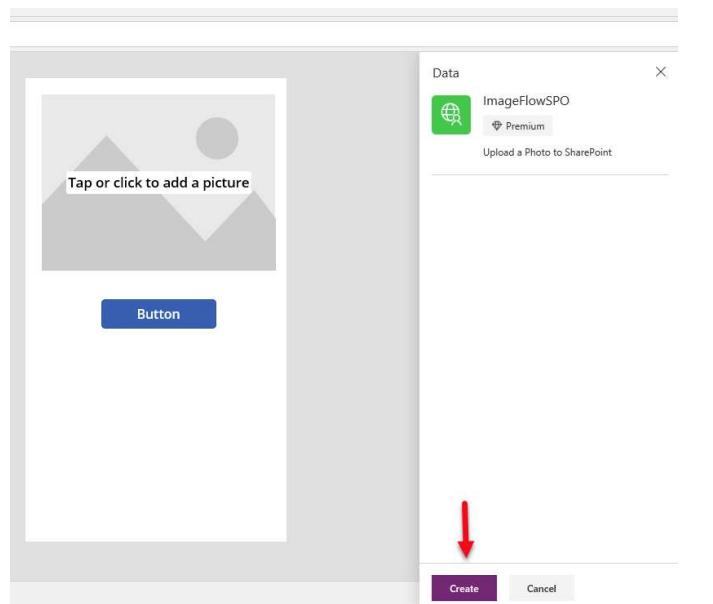


Scroll down to locate your '**custom connector**' I named mine '**ImageFlowSPO**' see image below.

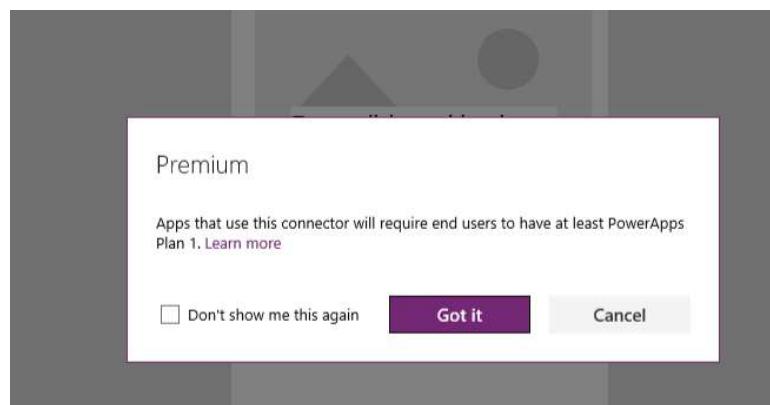


Click to select.

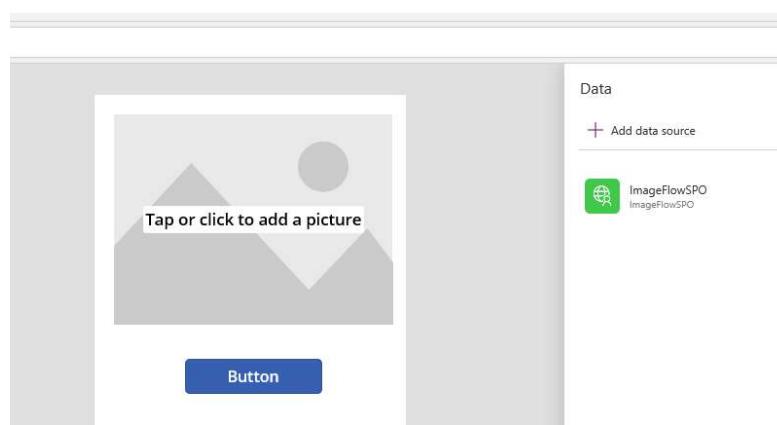
Click the '**Create**' button at the bottom see *image below*.



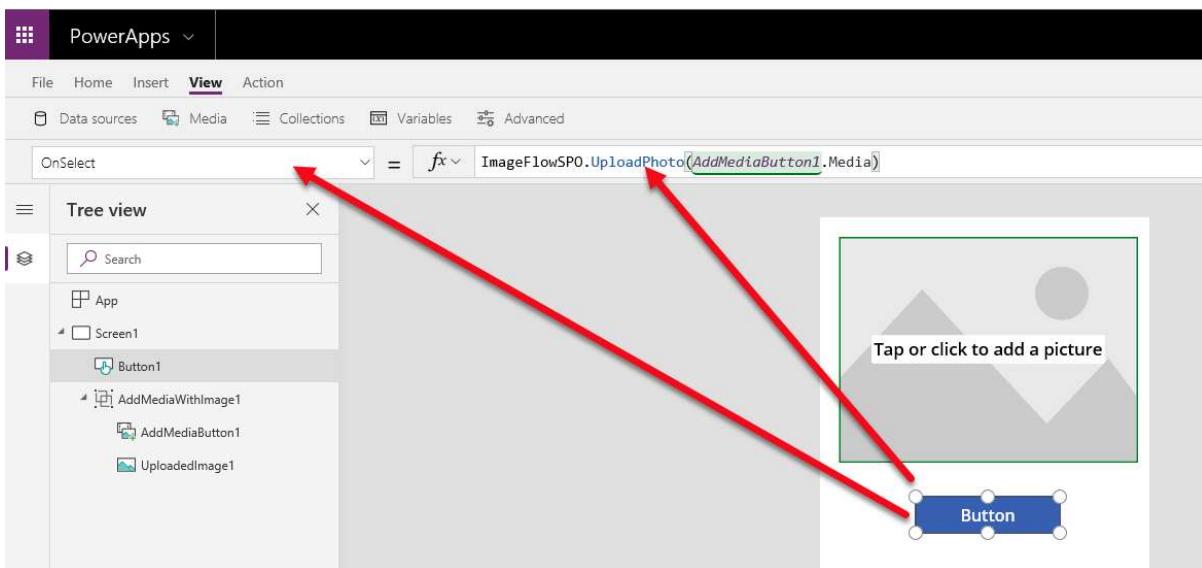
You will probably see the message window below. Assuming you have the right Plan click '**'Got it'** button *see image below.*



You should see your app has access to the data source *see image below.*



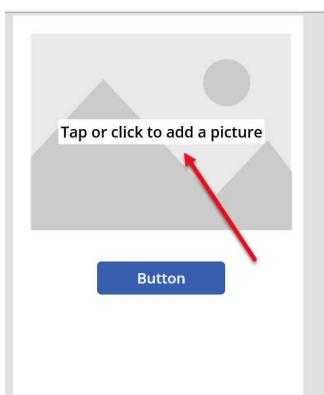
Select your Power App button. Check '**'On Select'** is showing. In the **'fx'** type '**'ImageFlowSPO.UploadPhoto(AddMediaButton1.Media)**' *see image below.*

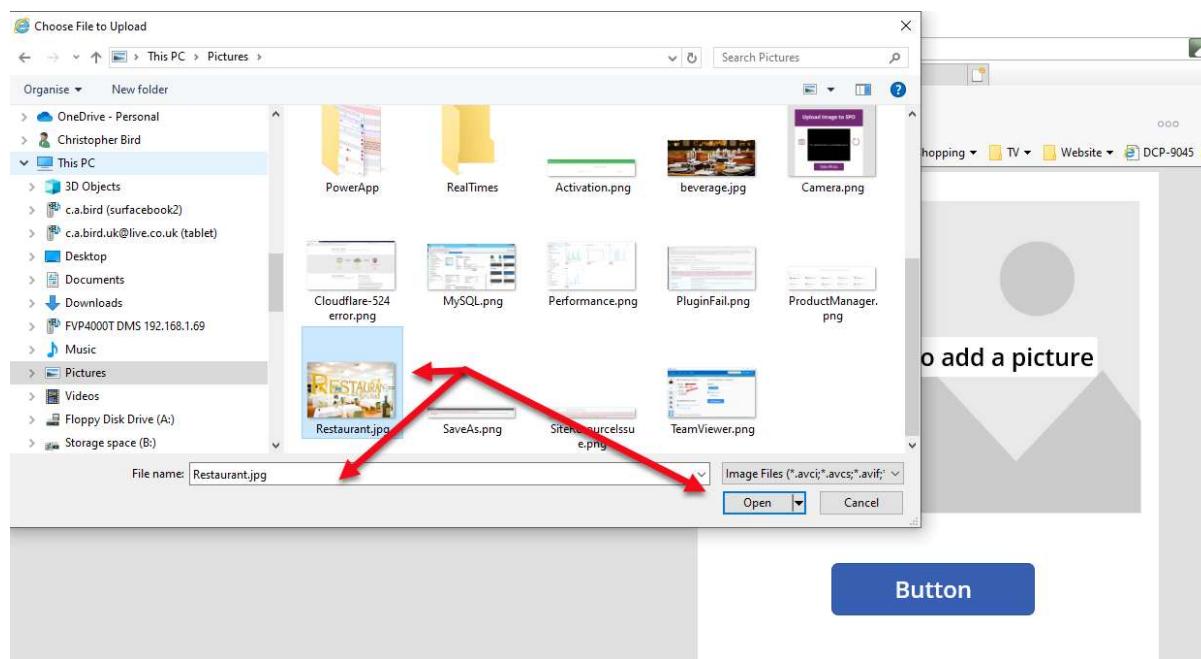


Now let's test this app. Click the 'Play' icon top right of the screen see image below.

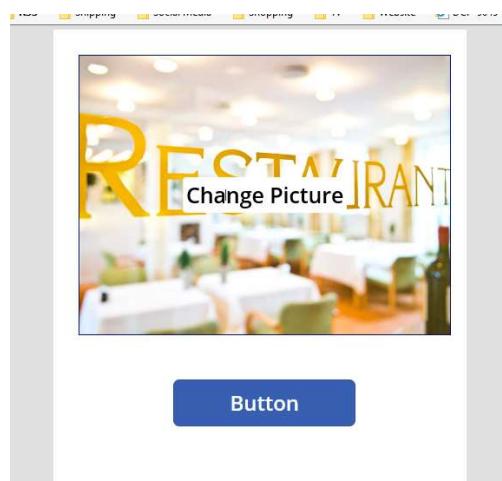


Click the 'Tap or click to add a picture.' Navigate to an image you want to store in SharePoint then select and click 'open' see image below.





Click the 'Button' see image below.



Watch for the dots running across the top of the screen.

Next navigate to your SharePoint site and view the image see *images below*.

PhotoSPO					
Name	Type Of Animal	Picture Size	File Size	Modified	+ Add column
beverage.jpg		610 x 250	62.7 KB	Yesterday at 12:20	
jump.jpg		610 x 250	62.7 KB	2 days ago	
Restaurant.jpg		1000 x 630	65.2 KB	A few seconds ago	



[Share](#) [Copy link](#) [Download](#) [Delete](#) [View original](#) [Copy to](#)



**CONGRATULATIONS. YOU HAVE NOW COMPLETED THE FIRST LAB IN THIS COURSE  
AND GONE THROUGH THE EXPERIENCE OF CREATING A SIMPLE APP.**

*If there is demand for further presentations, I will expand upon this presentation adding additional Power App capabilities for example automated file naming, uploading multiple images, description and audit meta data.*

## Summary

In this lab you learnt how to create a simple MS Power App using 'add picture control' and button to send the image to Flow. You learnt how to use Flow, http response, to acquire file content and using Compose data operations to convert image to Json object then convert json to binary 64 thus enabling the data to be passed to SharePoint Online.

Using a swagger file template; you then learnt how to use Flow http response URL to create a customised swagger file for your Power App. A swagger file is used to create a custom connector the connector enables Power Apps to pass data to Flow. Flow processes your image converts the data then uploads your image onto a SharePoint library.

**This concludes the lab exercise.**



## References:

I highly recommend gurus John Liu and Paul Culmsee. Their blog articles are brilliant.

**John Liu** blog: <http://johnliu.net/blog>

**Paul Culmsee** blog: <http://www.cleverworkarounds.com/>

**Swagger template**<sup>7</sup>: <https://github.com/C-A-BIRD/PowerAppFlowSPO>

```
{
  "swagger": "2.0",
  "info": {
    "description": "Upload a Photo to SharePoint",
    "version": "1.0.0",
    "title": "ImageFlowSPO"
  },
  "host": "prod-49.westeurope.logic.azure.com",
  "basePath": "/workflows",
  "schemes": [
    "https"
  ],
  "consumes": [],
  "produces": [],
  "paths": {
    "/f88fbg9900c34dd58abdc58b7be589f6/triggers/manual/paths/invoke": {
      "post": {
        "summary": "Upload a Photo to SharePoint",
        "description": "Upload a Photo to SharePoint",
        "operationId": "UploadPhoto",
        "consumes": [
          "multipart/form-data"
        ],
        "parameters": [
          {
            "name": "api-version",
            "in": "query",
            "default": "2016-06-01",
            "required": true,
            "x-ms-visibility": "internal",
            "type": "string"
          },
          {
            "name": "sp",
            "in": "query",
            "default": "/triggers/manual/run",
            "required": true,
            "x-ms-visibility": "internal",
            "type": "string"
          }
        ]
      }
    }
  }
}
```

---

<sup>7</sup> Copying and pasting code may result in errors. Recommendation download from GitHub click [here...](#) alternatively copy and paste into **Notepad** or **Notepad ++** be aware quotes may need to removed and retyped.



```

    "type": "string"
},
{
  "name": "sv",
  "in": "query",
  "default": "1.0",
  "required": true,
  "x-ms-visibility": "internal",
  "type": "string"
},
{
  "name": "sig",
  "in": "query",
  "default": "fAq1xXzH0l1spJnjt7dB1Uezmk8uoAXq2X_fvUX1GqE",
  "required": true,
  "x-ms-visibility": "internal",
  "type": "string"
},
{
  "name": "File",
  "in": "formData",
  "description": "file to upload",
  "required": true,
  "type": "file"
}
],
"responses": {
  "200": {
    "description": "successful operation"
  }
}
}
},
"definitions": {},
"parameters": {},
"responses": {},
"securityDefinitions": {},
"security": [],
"tags": []
}
}

```