

Lab 14. Application Lifecycle Management Light

Learning objective: a good practice in programming is to be able to reuse code. Creating reusable and generic flow is a great way to reuse code and to make your code more robust. Child flow are part of the standard Power Platform license and does not require a premium license.

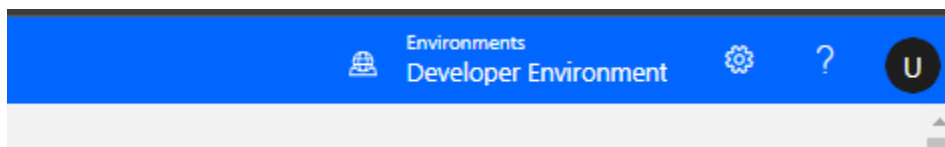
Duration: 30 minutes

Scenario: we will create a flow that will retrieve information from a SharePoint list and from an excel file. In order to make this flow easier to be deployed across several environment we will dynamically provide the list and file url. We will create a child flow that will retrieve this information from environment variables. Child flow must be created from a Solution, so you will create a solution as well.

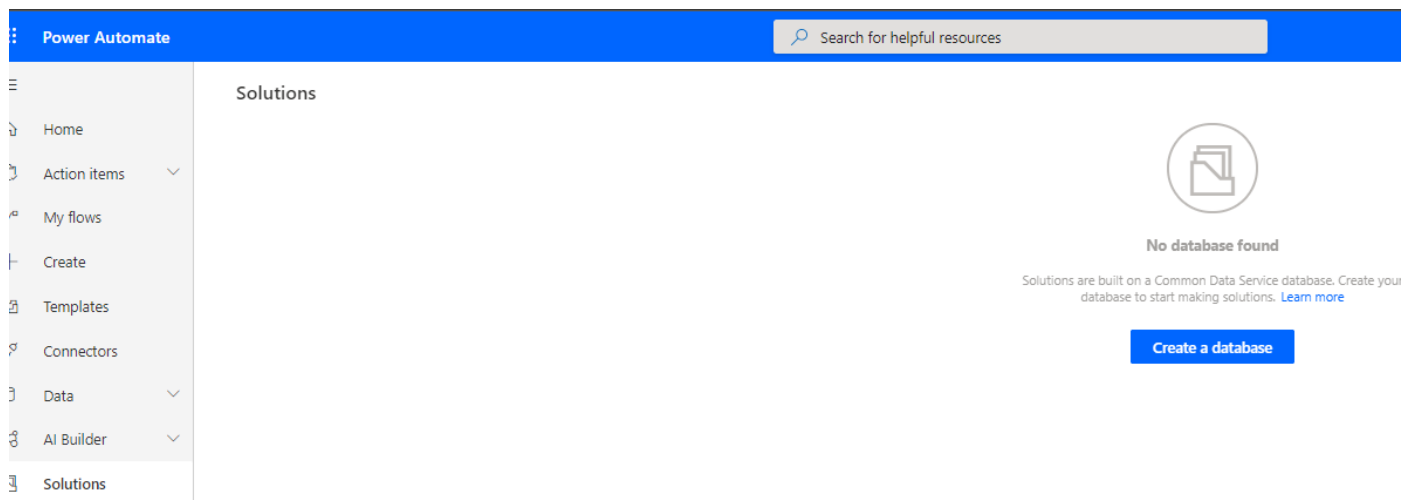
PART 1. Define the parameters

Tasks:

1. Make sur you have a premium license; this will be necessary to retrieve Environment variable from a flow 'see lab0).
2. Switch to your Developer environment



3. Go to Solutions and create a new solution from scratch; if requested click **Create a database**.



4. Fill in the database properties and click **create my database**:

New database



Choose the currency and language your data should use. ⓘ

Currency ⓘ

EUR



Language ⓘ

English



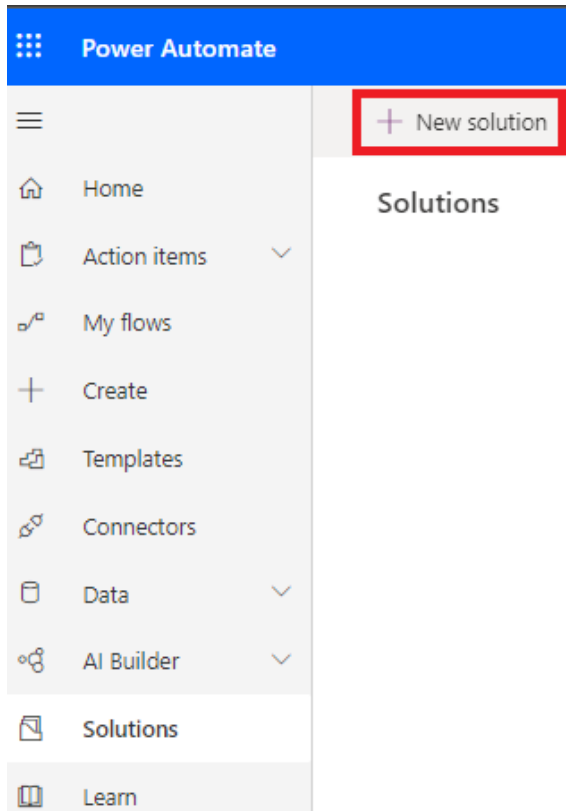
☐ Include sample apps and data

ⓘ By choosing Create my database, you agree Microsoft can use entity and field names that you create (but not content in the database tables) to help improve our common data model. These names may be stored in our diagnostic systems and copied across regions. [Learn more](#)

Cancel

Create my database

Wait a couple of minutes for the database to be created.
Click New Solution



Fill-in the solution properties and under Publisher click + Publisher:

New solution ×

Display name *

serge Solution

Name *

sergeSolution

Publisher *

+ Publisher

CDS Default Add new publisher

Contoso

Default Publisher for org969b872b

Microsoft First Party

In the next windows, replace the existing publisher with a new one by clicking on **+ Publisher**:

Solution settings ×

Display name *

Serge Solution

Name *

SergeSolution

Publisher *

+ Publisher

CDS Default Publisher Add new publisher

Default Publisher for org969b872b

Microsoft First Party

1. A new window will pop up where you can select a new Display Name like **Contoso** and a prefix like **contoso**:

https://org969b872b.crm4.dynamics.com/

File Save and Close

Publisher: New Publisher

Information

General

Display Name *

Description

Set the prefix name for custom entities and fields

Prefix *

Select **Contoso** as the new publisher and click Update:

Solution settings

Display name *

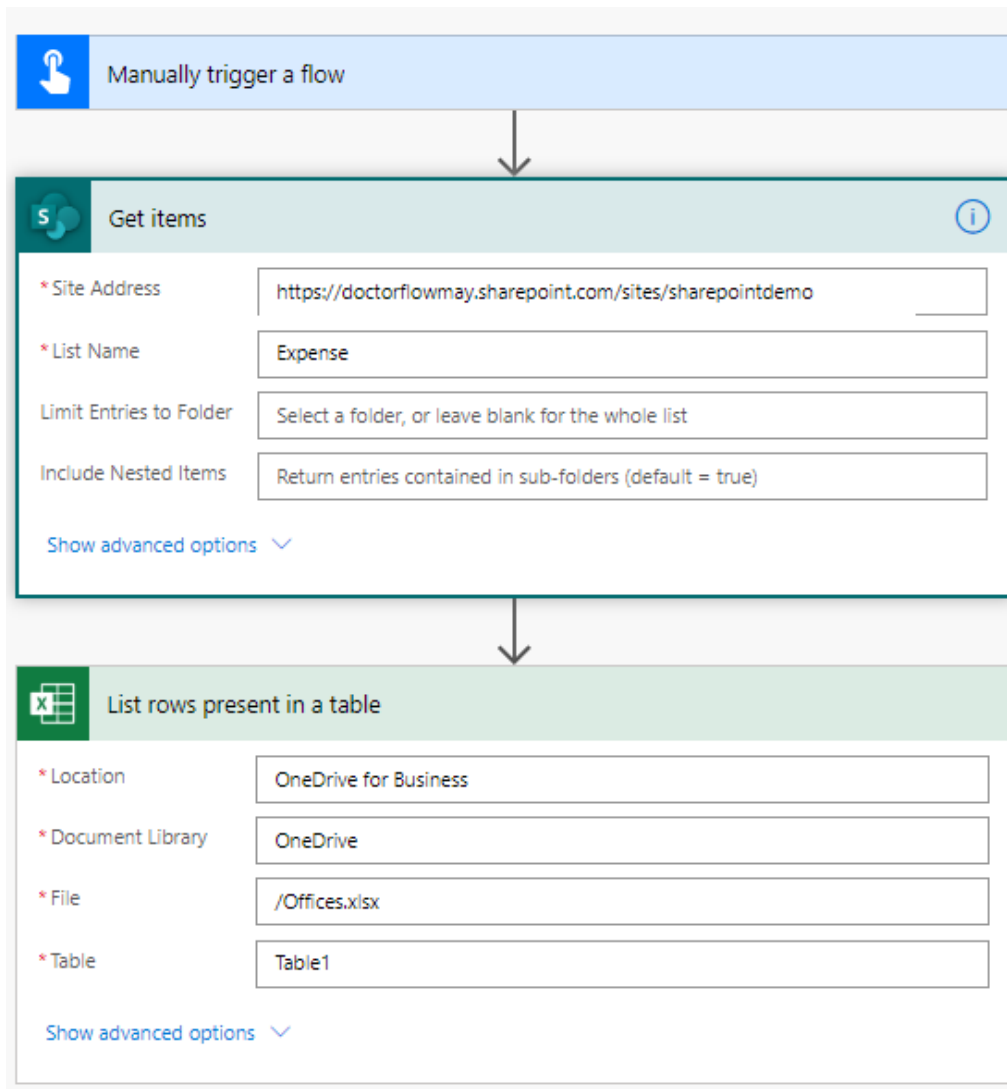
Name *

Publisher *

Version *

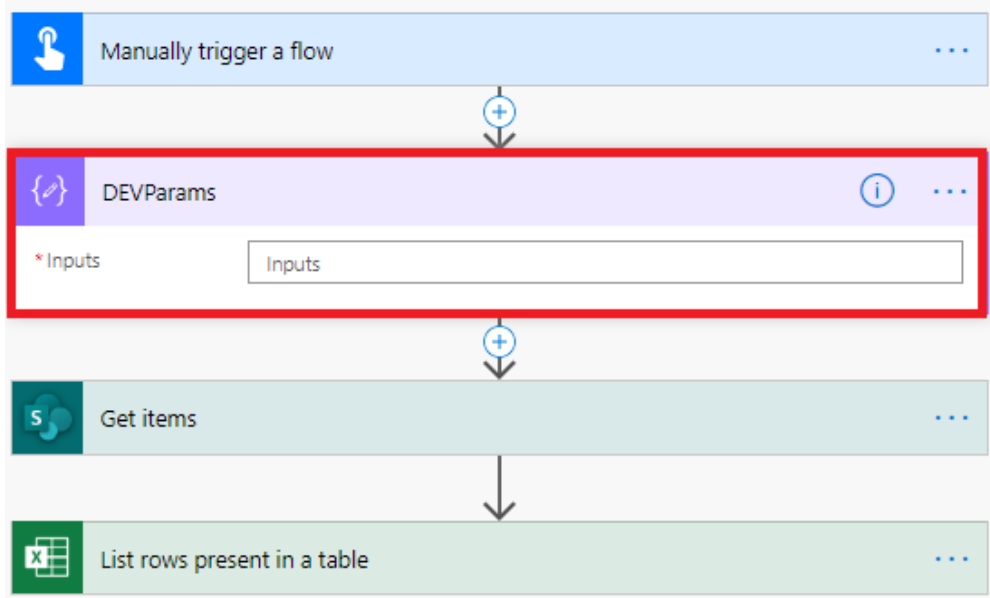
2. Click Create:

Click your solution and create a Flow that is started manually and that connects to a SharePoint list and to an Excel document with a table:

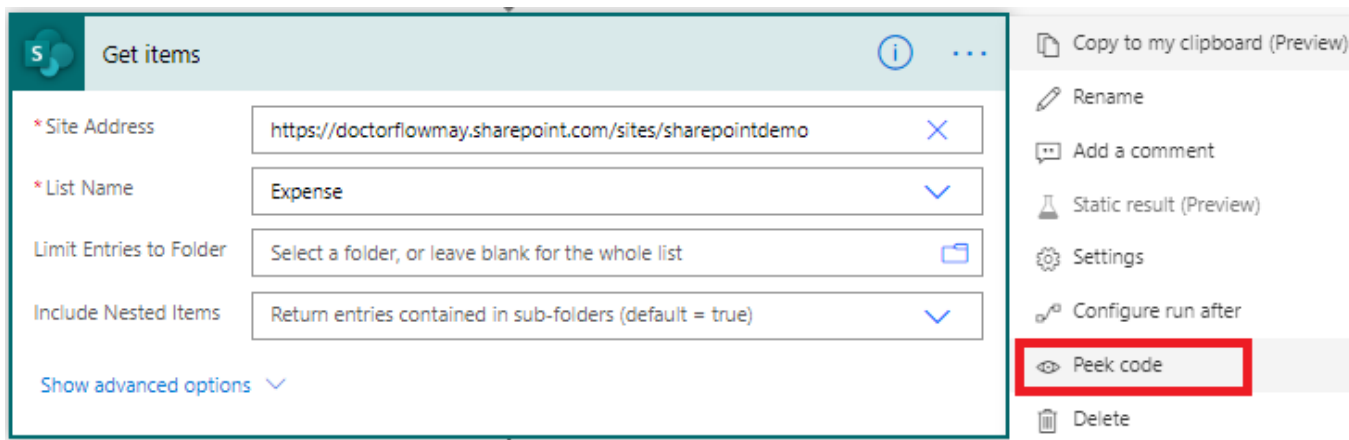


Enterprise flows can have many actions and deploying these flows from different environments (DV, TEST, PROD) can quickly become a nightmare.

The first option is to create a new Compose called DEVParams that will contain the internal references:



To retrieve the references, you need to click the Peek code option of the action:



Manually trigger a flow

DEVParams

*Inputs

```
{
  "siteUrl": "https://doctorflowmay.sharepoint.com/sites/sharepointdemo",
  "expenseListGuid": "5a948e7a-ad8d-4fe3-9ebb-96be91ad0e8a"
}
```

Add dynamic content

'Get items' (code view)

```
1
2  "inputs": {
3    "host": {
4      "connectionName": "shared_sharepointonline",
5      "operationId": "GetItems",
6      "apiId": "/providers/Microsoft.PowerApps/apis/
shared_sharepointonline"
7    },
8    "parameters": {
9      "dataset": "https://doctorflowmay.sharepoint.com/sites/
sharepointdemo",
10     "table": "5a948e7a-ad8d-4fe3-9ebb-96be91ad0e8a"
11   },
12   "authentication": {
13     "type": "Raw",
14     "value": "@json(decodeBase64(triggerOutputs().headers
['X-MS-APIM-Tokens']))['$ConnectionKey']"
15   }
}
```

Proceed like this to retrieve the references used by the Excel action :

DEVParams

* Inputs

```
{
  "siteUrl": "https://doctorflowmay.sharepoint.com/sites/sharepointdemo",
  "expenseListGuid": "5a948e7a-ad8d-4fe3-9ebb-96be91ad0e8a",
  "exceldrive": "b!p1JExxcz7EWQCHu2ZAHIfzEs-
fD6g2ZCsTEO3FpLAsX2yOzq19oqRYhJ4Jje99Fg",
  "excelFile": "01FUEHPJSOIJYVAX6IBNDLKPAPIJSXA5Q5",
  "excelTable": "{94A1FF6D-4D3F-4F2C-920C-26CD17B7C4FA}"
}
```

↓

S Get items

* Site Address

fx

outputs()x

×

* List Name

fx

outputs()x

×

Limit Entries to Folder

Select a folder, or leave blank for the whole list

📁

Include Nested Items

Return entries contained in sub-folders (default = true)

✓

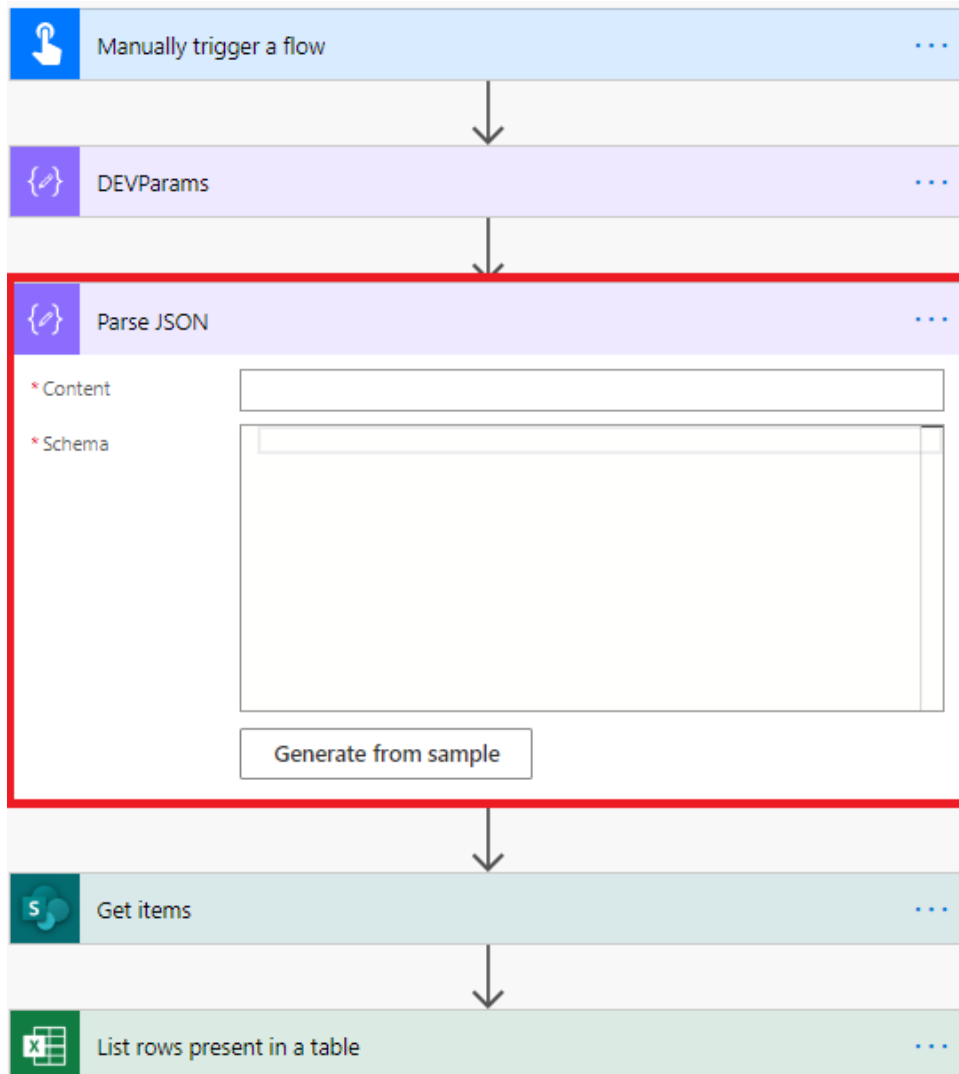
Show advanced options ▾

⬇

'List rows present in a table' (code view)

```
1
2  "inputs": {
3    "host": {
4      "connectionName": "shared_excelonlinebusiness",
5      "operationId": "GetItems",
6      "apiId": "providers/Microsoft.PowerApps/apis/
shared_excelonlinebusiness"
7    },
8    "parameters": {
9      "source": "me",
10     "drive":
"b!p1JExxcz7EWQCHu2ZAHIfzEs-fD6g2ZCsTEO3FpLAsX2yOzq19oqRYhJ4Jje99Fg",
11     "file": "01FUEHPJSOIJYVAX6IBNDLKPAPIJSXA5Q5",
12     "table": "{94A1FF6D-4D3F-4F2C-920C-26CD17B7C4FA}"
13   }
14 }
```

We need to use dynamic properties to use these references in our code. Add a Parse JSON action and rename it **Params**:



Copy the JSON information from the action DEVParams and click **Generate from sample**, paste the info in there:

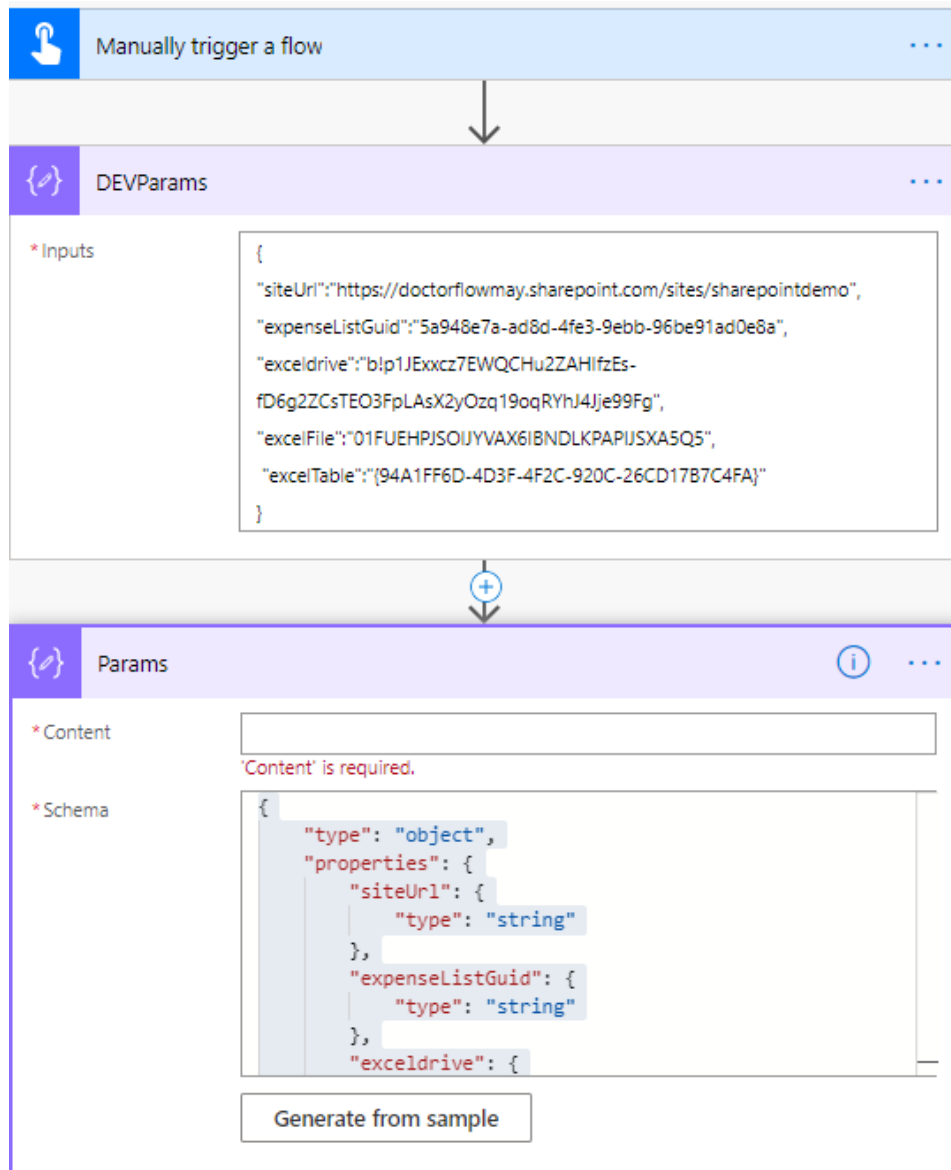
Insert a sample JSON Payload



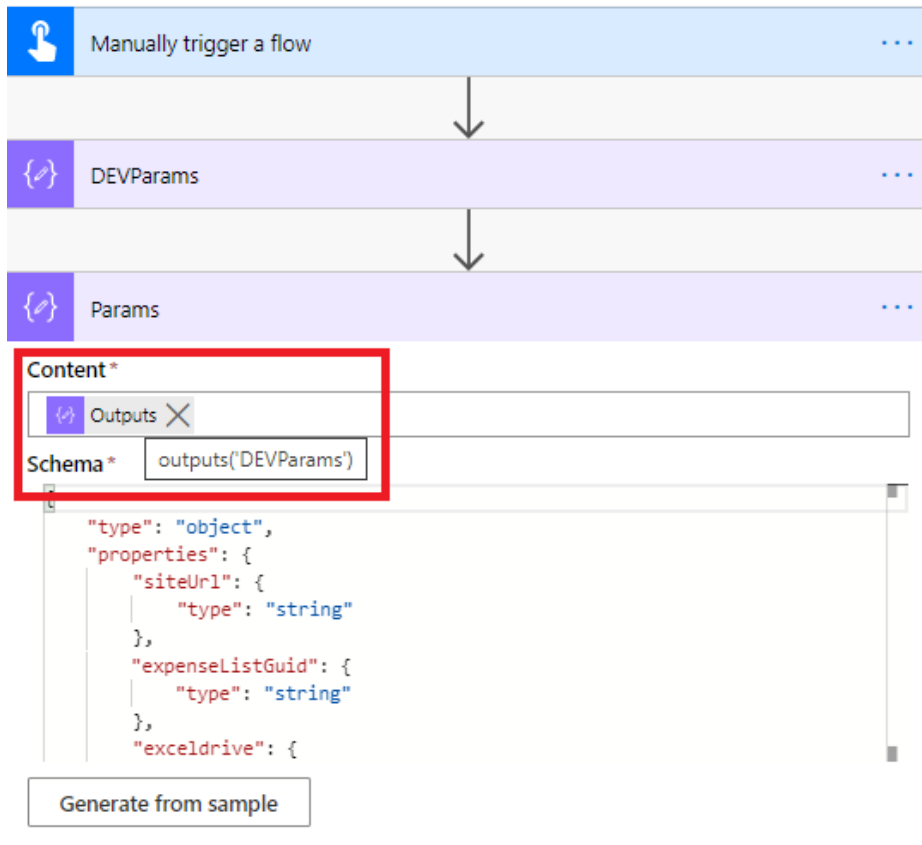
① Clicking 'Done' will overwrite your current schema

```
{  
  "siteUrl": "https://doctorflowmay.sharepoint.com/sites/sharepointdemo",  
  "expenseListGuid": "5a948e7a-ad8d-4fe3-9ebb-96be91ad0e8a",  
  "exceldrive": "b!p1jExxcz7EWQChu2ZAHIfzEs-fD6g2ZCsTE03FpLAsX2yOzq19oqRYhJ4Jje99Fg",  
  "excelTable": "{94A1FF6D-4D3F-4F2C-920C-26CD17B7C4FA}"  
}
```

Done



Use DevParams output in Params content:



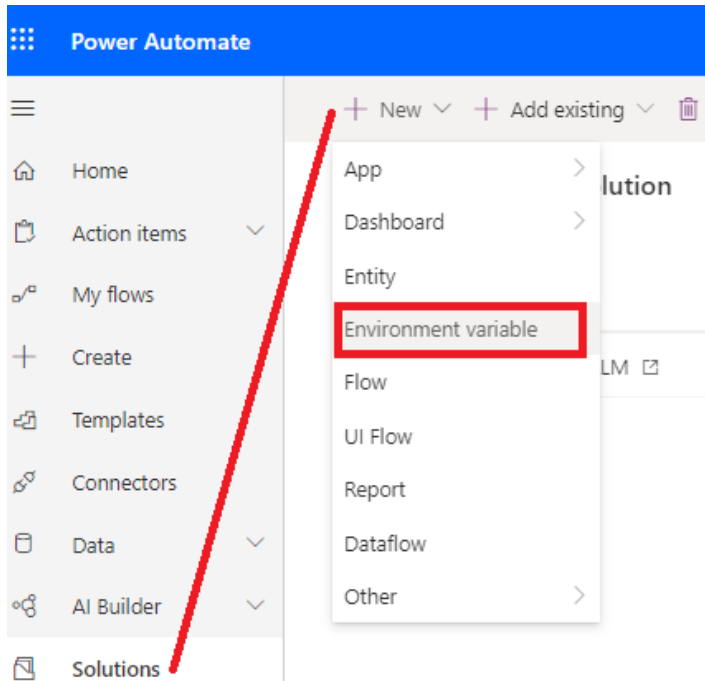
Replace the reference in the SharePoint and Excel actions with dynamic value. Now if you need to deploy your flow to another environment (like another SharePoint Test site), you can create a new Compose object named TESTParams with the reference in test. And you just bind Params with TESTParams. Test your flow, it should still be working.

PART 2. Use environment variables

This JSON information can be stored in an environment variables. In the next part we will create a child flow that will retrieve this information from environment variables.

Tasks:

1. Click your solution, select New and create an environment variable:



Name the environment variable "PARAMS" and select the Data type JSON and store the JSON data stored in the action DEVParams and click Save:

New environment variable ✕

Enter information about this variable that will help others use it when it's imported into their environments. [Learn more](#)

Display name *

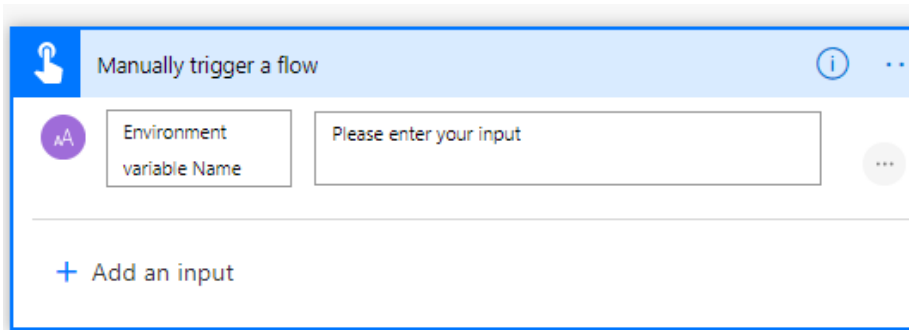
Name * ⓘ

Description

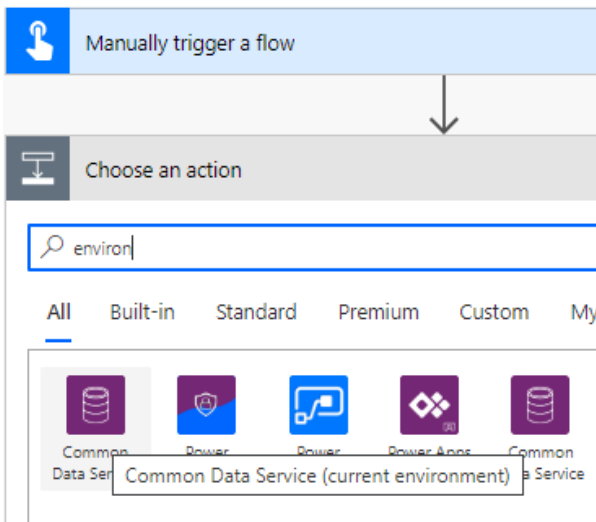
Data Type *

Default Value ⓘ

We now need to dynamically retrieve this information from our flows. From the solution create new flow and named it "Find Params". The trigger should be the "Manually trigger a flow" trigger. Add a first parameter called Environment variable:



Add an action **List records** from the Common Data Service (current environment) connector:



Rename the action as **List records - Find Environment variable definition**

In the filter **Query property**, filter on *schemaname eq '<your environment variable name>'*:

Manually trigger a flow

List records - Find Environment variable definition

Entity name *

Environment Variable Definitions

Select Query

Limit the properties returned while retrieving data.

Filter Query

schemaname eq 'Params'

Order By

An ODATA orderBy query for specifying the order of entries.

Expand Query

Related entries to include with requested entries (default = none).

Fetch Xml Query

Fetch Xml query

Top Count

Total number of entries to retrieve (default = all).

Skip token

The skip token.

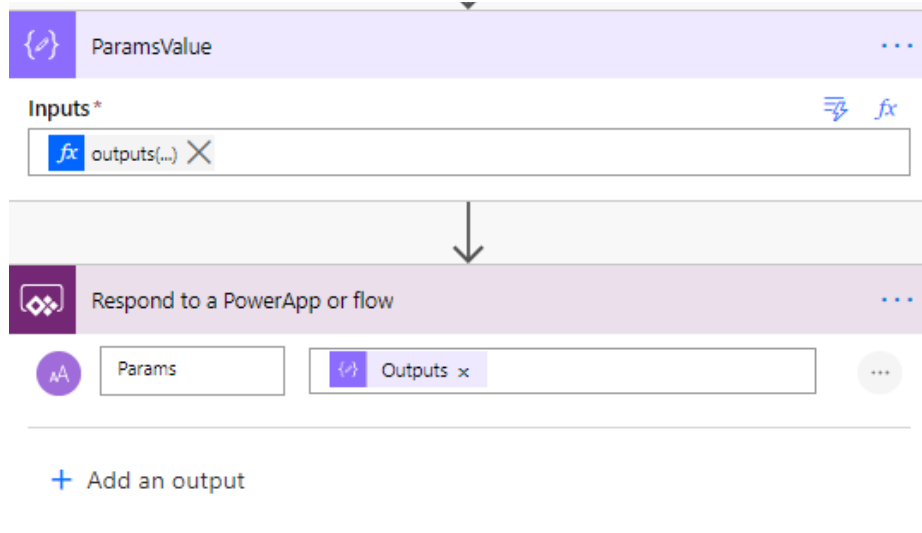
[Hide advanced options](#)

Add a **Compose** action and name it **ParamsValue** :

Set its value to

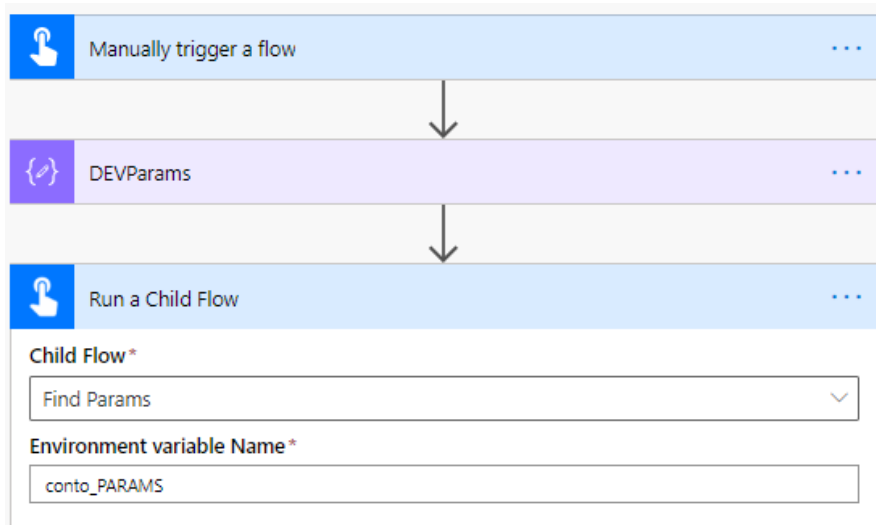
```
outputs('List_records_-_Find_Environment_variable_definition')?['body']?['value']?[0]?['defaultvalue']
```

Add a **Responds to a PowerApps or flow** action to return the **ParamsValue**:

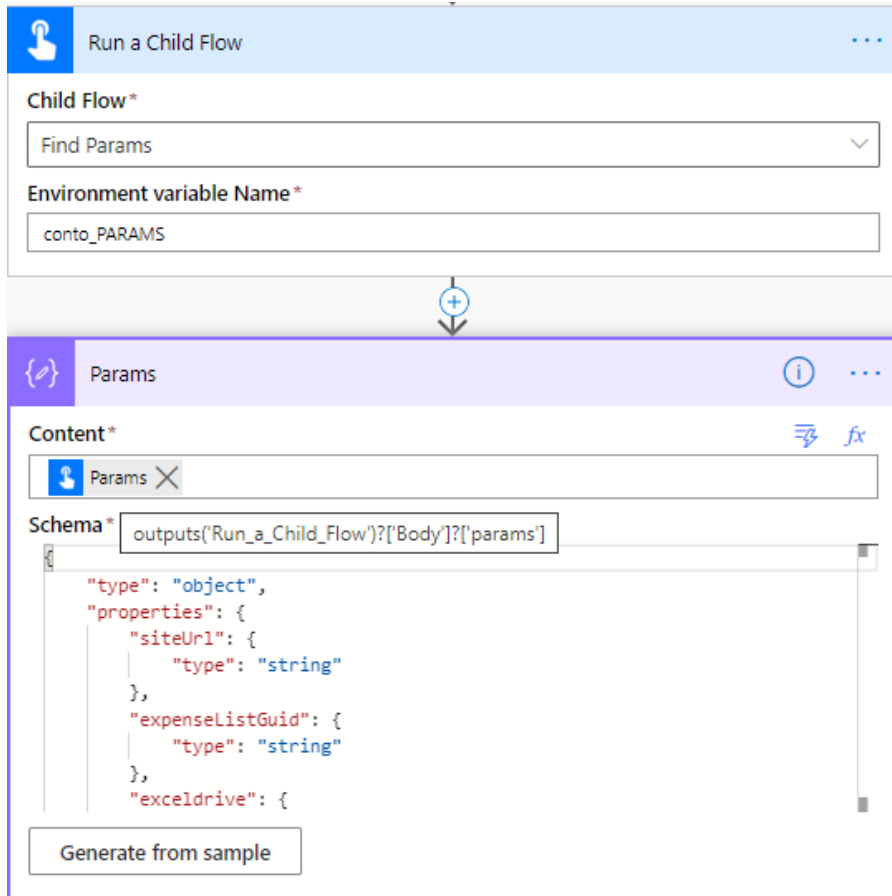


You will now call his flow from the flow you created in part 1 of this lab. Edit the flow

and add a Run a child flow action



In the Params action, grab the returned value of Run a child flow :



Run a Child Flow

Child Flow*
Find Params

Environment variable Name*
conto_PARAMS

Params

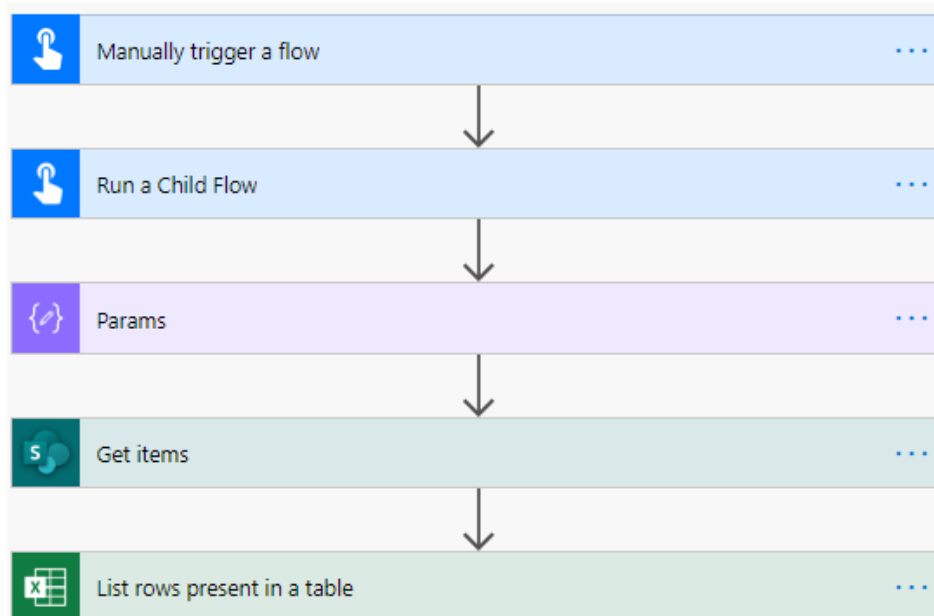
Content*
Params

Schema*
outputs('Run_a_Child_Flow')['Body']['params']


Generate from sample

Test your flow.


The Action **DEVParams** can now be removed.





As a good practice we can group Run a child flow and Params in a **scope** called **ALM**:


 Manually trigger a flow ...



 ALM ...

 Run a Child Flow ...


 Params ...

 Add an action



 Get items ...



 List rows present in a table ...

