

Creating a PowerApp with Wiise

Item Journal Lines

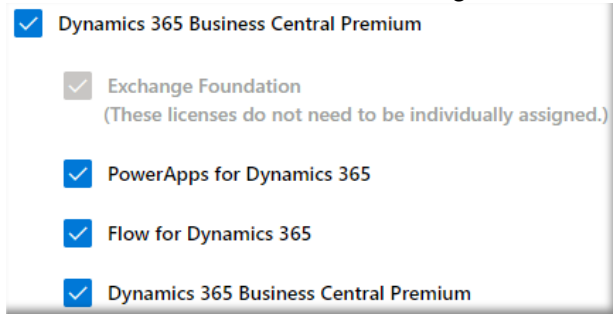
Table of Contents

- Wiise..... 3
 - User Setup 3
 - Web Services Setup..... 5
- Postman..... 6
 - Test connection 6
 - Test connection with URL variables 8
- Power Apps..... 10
 - Create Custom Connector 10
 - Create a Power App..... 17
 - Item Journal List..... 17
 - Submit Item Journal Line 21

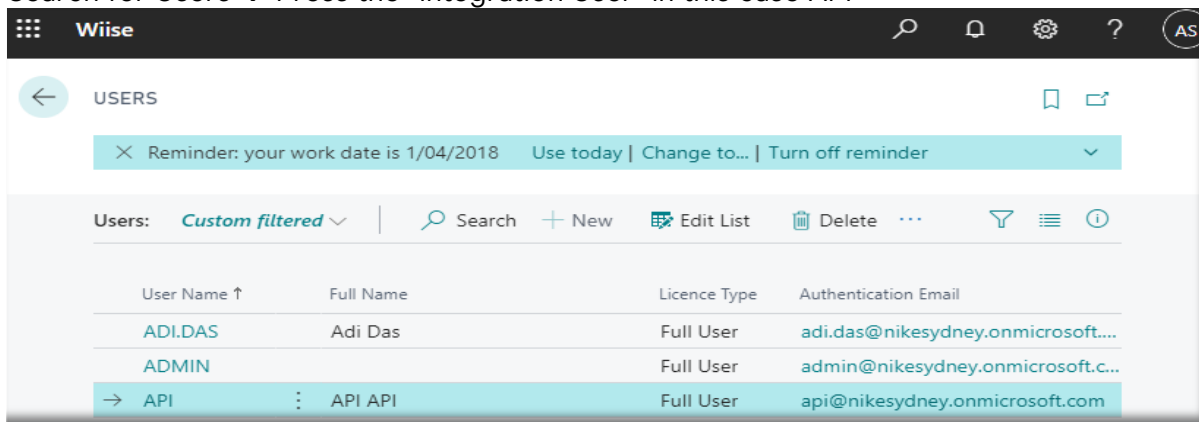
Wiise

User Setup

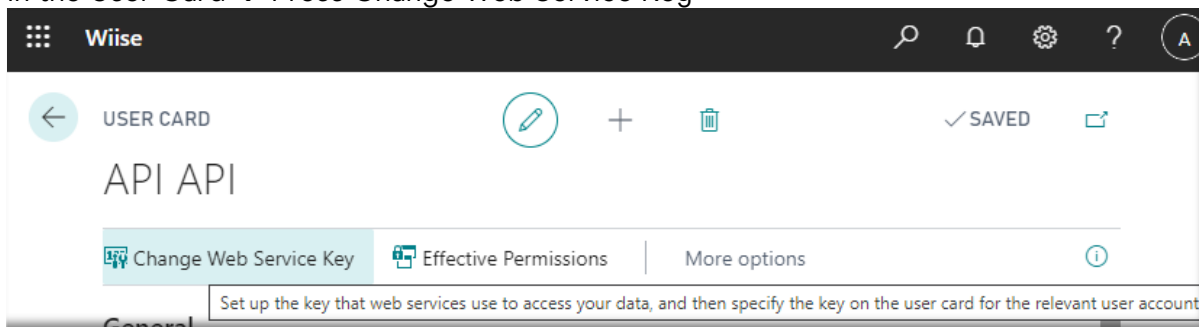
Log into wiise with the Integration User. This user will use basic authentication to integration with Wiise. The user should have the following in their Assigned license.



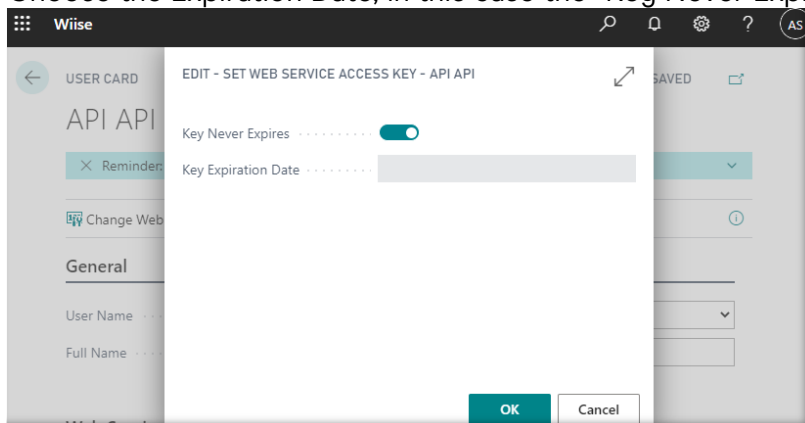
Search for Users → Press the “Integration User” in this case API



In the User Card → Press Change Web Service Key

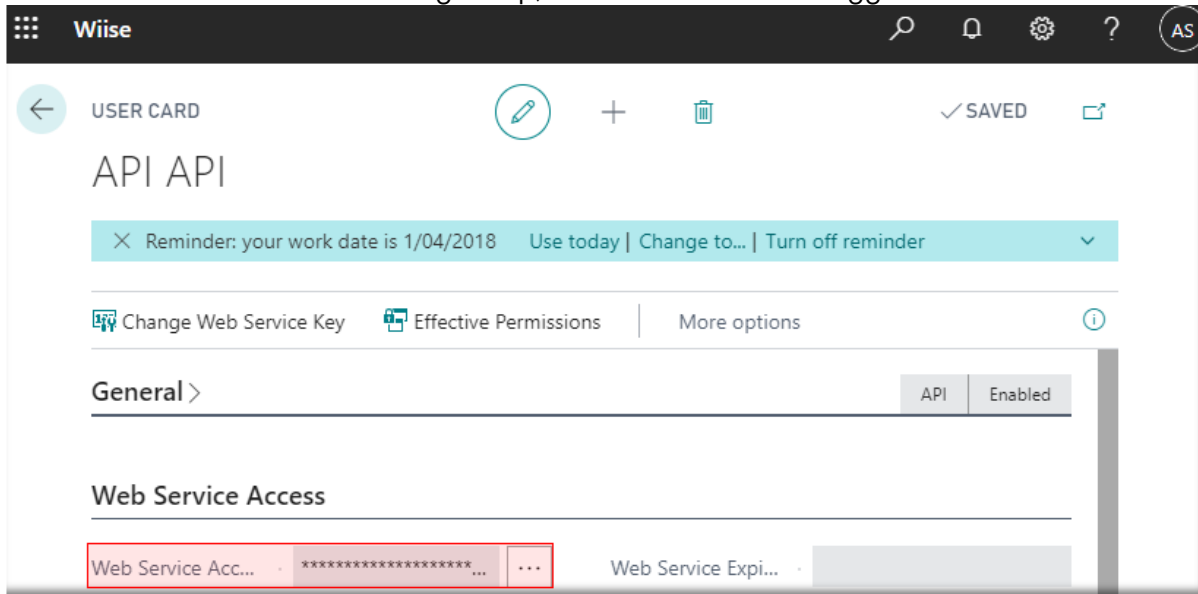


Choose the Expiration Date, in this case the “Key Never Expires”

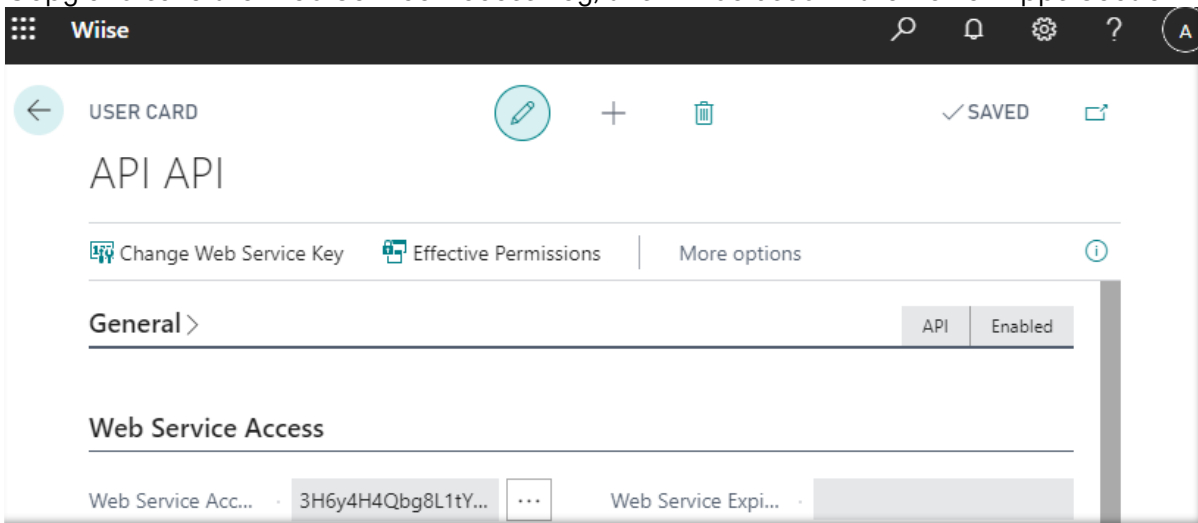


Press OK

If the User Card returns ***** in the Web Service Access Key Wiise has been accessed with a different user than the user being setup, in this case the user logged in is ADMIN and not API.



Copy and save the Web Service Access key, this will be used in the Power Apps Section



Web Services Setup (Item Journal)

Search for Web Services → Edit List → +New, Page, 40, in this case the Service Name nativeItemJournals is added.

Object Type ↑	Object ID	Object Name	Service Name ↑	All Ten...
Page	2860	nativeInvoicingUnitsOfMeasure	nativeInvoicingUnitsOfMeasure	
Page	2866	Native - GST Setup	nativeInvoicingVATSetup	
→ Page	40	Item Journals	nativeItemJournals	

Tick → Published

If the OData V4 URL link does not appear, move away from the record. It should automatically appear.

Copy the OData V4 URL link, this will be used in the Power Apps Section

[https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company\('CRONUS%20AU'\)/nativeItemJournals](https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeItemJournals)

Object Type ↑	Pub...	OData V4 URL
Page	<input checked="" type="checkbox"/>	https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeItemJournals
Page	<input checked="" type="checkbox"/>	https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeItemJournals
→ Page	<input checked="" type="checkbox"/>	https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeItemJournals

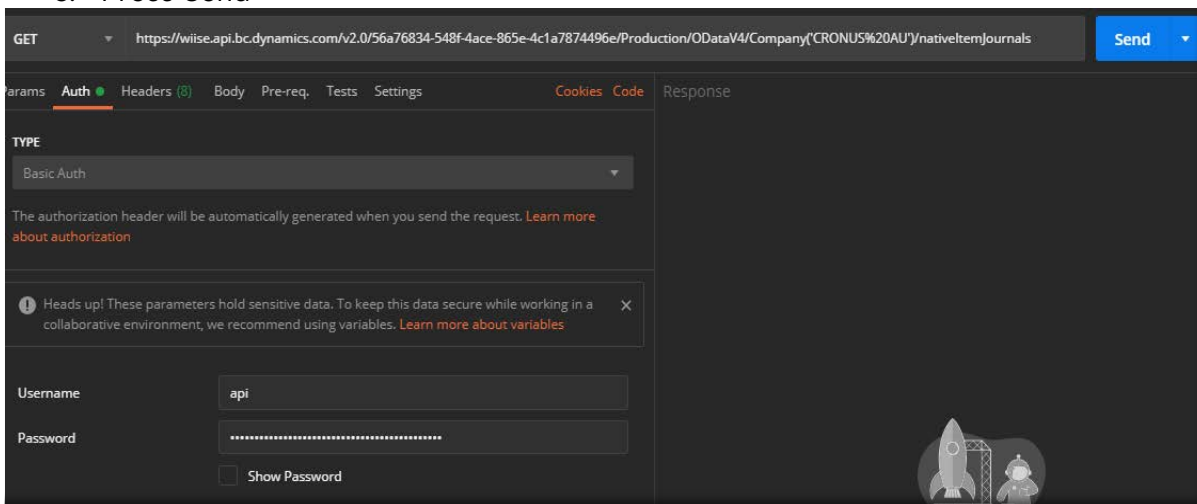
Postman

Test connection

Note: We will need the result of this test for the Power Apps Section

In Postman

1. Create new Request
2. Request Type → Get
3. Enter request URL → [https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company\('CRONUS%20AU'\)/nativeternJournals](https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeternJournals)
4. Auth (Tab) → Choose Type Basic Auth, enter Username and Password (Web Service Access Key) from previous steps
5. Press Send



If item Journal Lines exist expect a result like (Copy the result output)

```

Body ▾ 200 OK 5.98 s 2.54 KB Save Response ▾
Pretty Raw Preview Visualize JSON ↻
1 {
2   "@odata.context": "https://wiise.api.bc.dynamics.com/v2.0/
   56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/$metadata#Company
   ('CRONUS%20AU')/nativeItemJournals",
3   "value": [
4     {
5       "@odata.etag": "W/
   \JzQ0011tUkR0QjVMdm1RSU5wZXAxVE5wVWFwFw5EdVjXbjVZbU1rSvpjOFNQME09MTswMDsn\"
6       ,
7       "Journal_Template_Name": "ITEM",
8       "Journal_Batch_Name": "DEFAULT",
9       "Line_No": 10000,
10      "Posting_Date": "2020-11-14",
11      "Document_Date": "2020-11-14",
12      "Entry_Type": "Positive Adjmt.",
13      "Price_Calculation_Method": "Not defined",
14      "Document_No": "CYCLECOUNT",
15      "External_Document_No": "",
16      "Item_No": "1896-S",
17      "Variant_Code": "",
18      "Description": "ATHENS Desk",
19      "Location_Code": "",
20      "Bin_Code": "",
21      "Salespers_Purch_Code": "",
22      "Gen_Bus_Posting_Group": "",
23      "Gen_Prod_Posting_Group": "RETAIL",
24      "Quantity": 5,
25      "Unit_of_Measure_Code": "PCS",
26      "Unit_Amount": 1477,
       "Amount": 7385,
     }
   ]
}

```

Otherwise a correct result should look like.

```

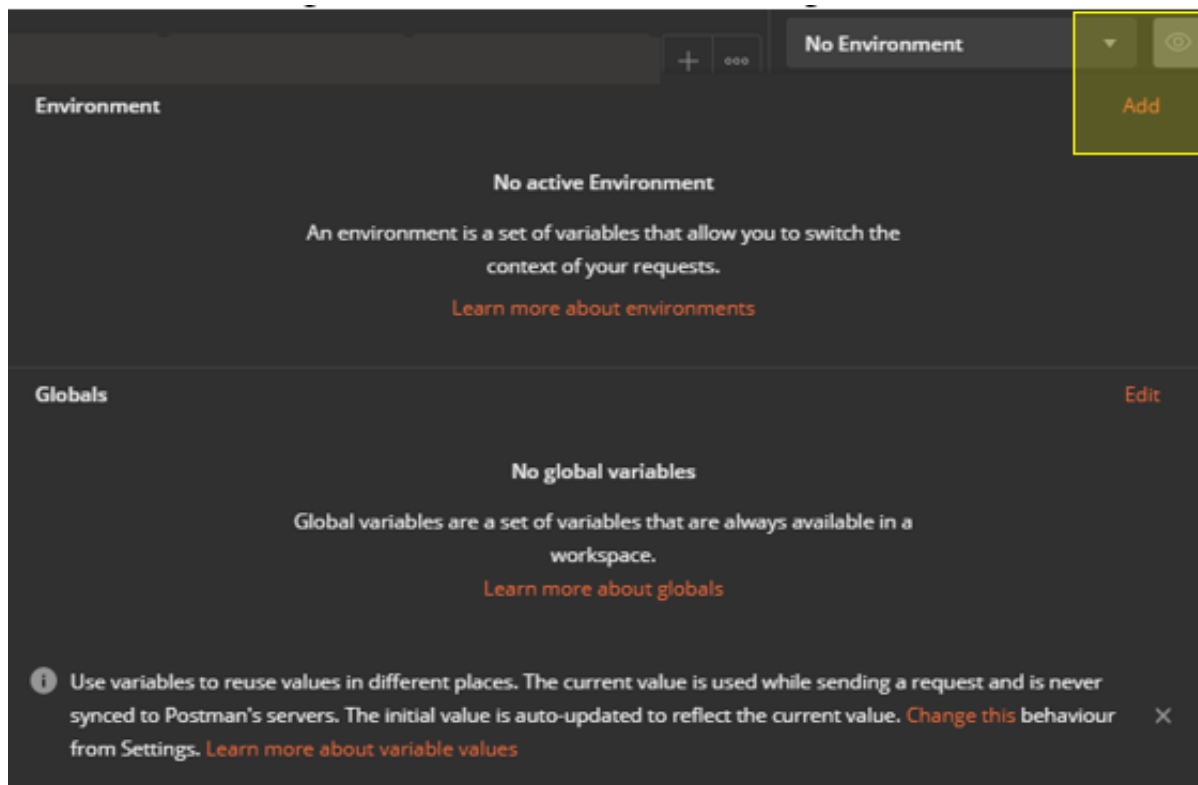
Body ▾ 200 OK 173 ms 1.11 KB Save Response ▾
Pretty Raw Preview Visualize JSON ↻
1 {
2   "@odata.context": "https://wiise.api.bc.dynamics.com/v2.0/
   56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/$metadata#Company
   ('NIKE%20SYDNEY')/nativeItemJournals",
3   "value": []
4 }

```

Test connection with URL variables

Postman can also be setup to use variables in the URL

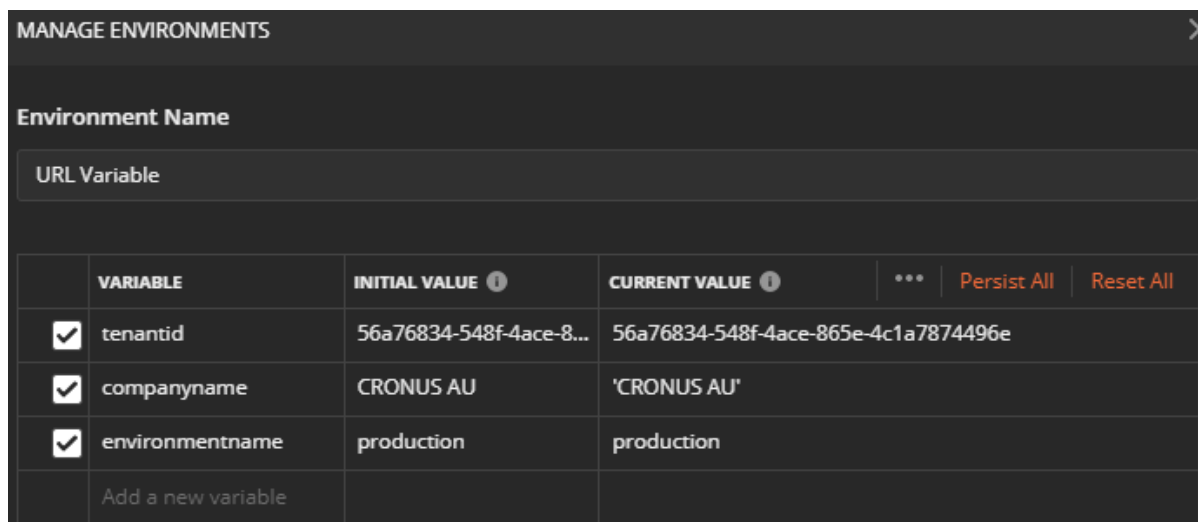
Press Environment Quick Look → Add



Define a name, in this case URL Variables

Add

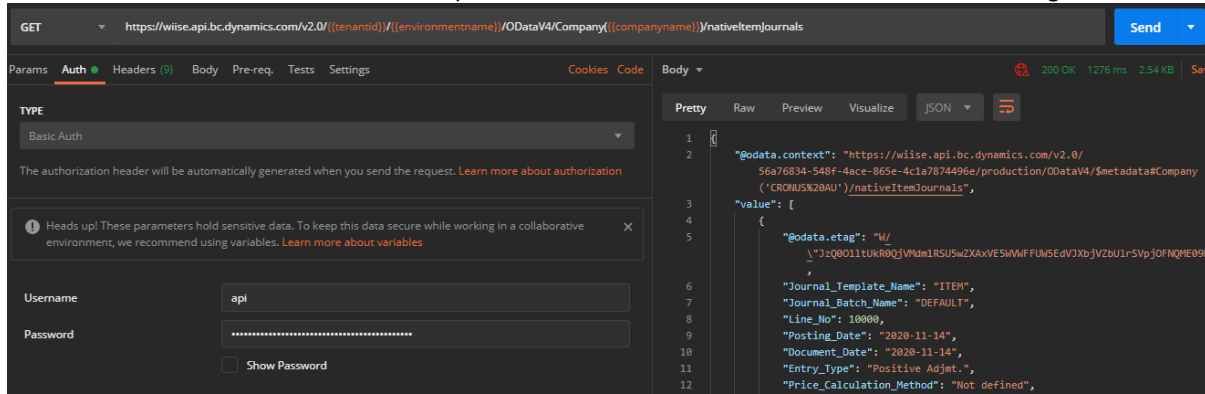
tenantid = 56a76834-548f-4ace-865e-4c1a7874496e (can also be replaced with domain name)
 environmentname = Production (can also be replaced with sandbox environment name)
 companyname = 'CRONUS AU'



Use the following URL

`https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}/ODataV4/Company({companyname})/nativeItemJournals`

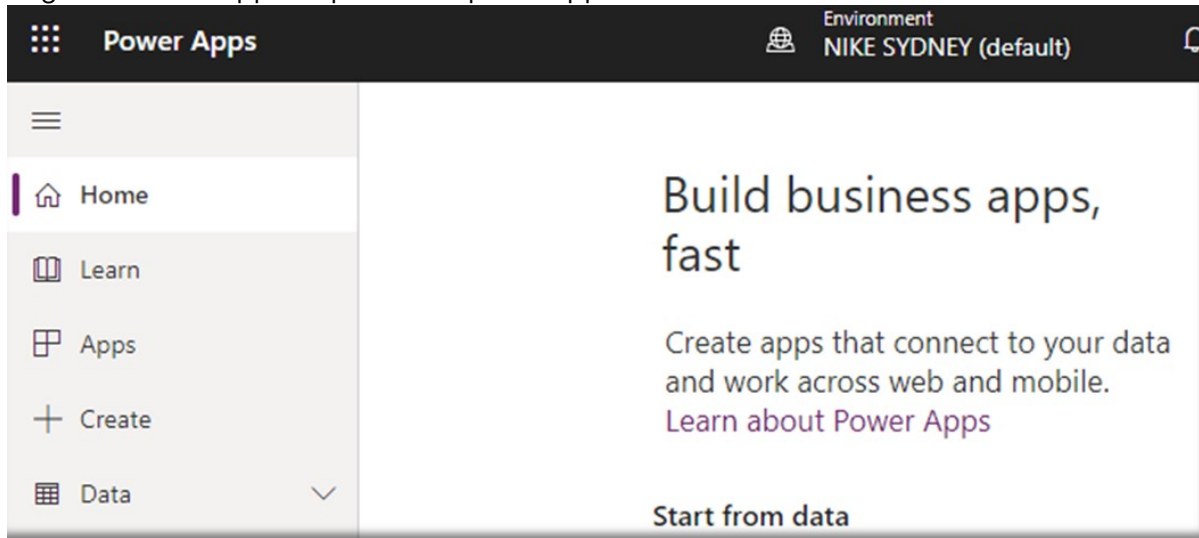
Note: if the variables are found in postman, the URL variables will show as orange.



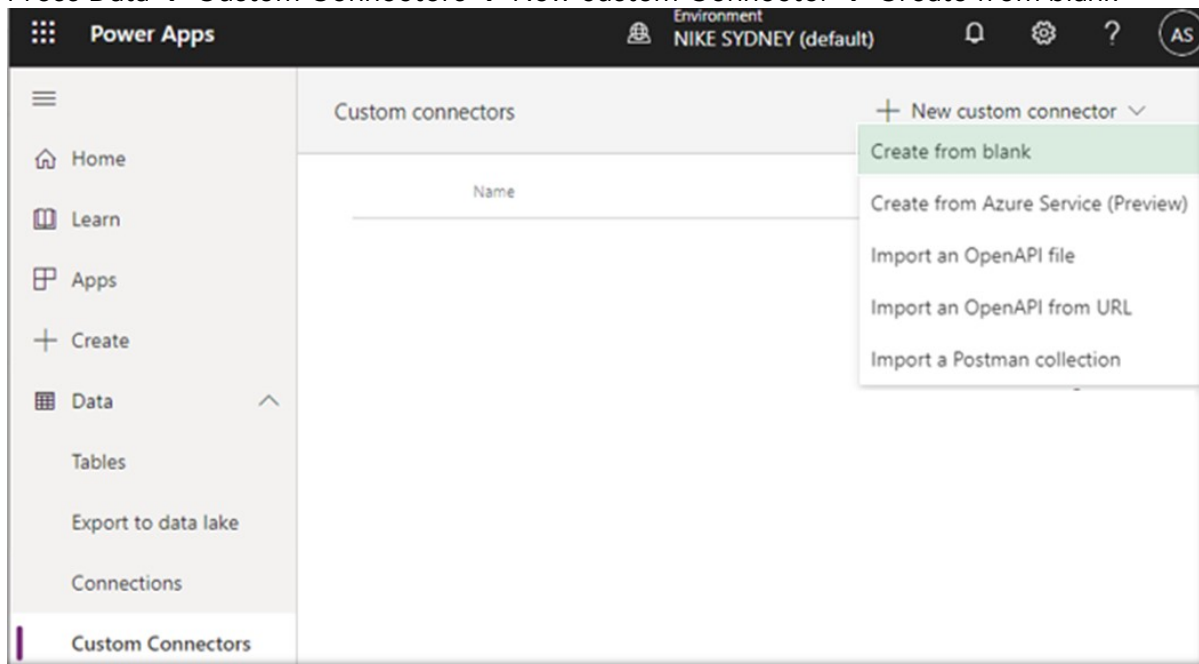
Power Apps

Create Custom Connector

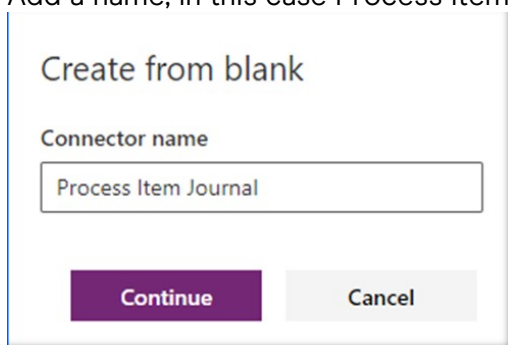
Log into Power Apps <https://make.powerapps.com/> with an admin user



Press Data → Custom Connectors → New custom Connector → Create from blank



Add a name, in this case Process Item Journal



General

Host: wiise.api.bc.dynamics.com

Base URL: /v2.0/

← Connector Name Process Item Journal

1. General > 2. Security > 3. Definition > 4. Test Swagger Editor ✓ Create connector ✗ Cancel

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.

General information

Upload connector icon

Supported file formats are PNG and JPG. (< 1MB)

↑ Upload

Icon background color

A color to show behind the icon (e.g., '#007ee5')

Description

Give your custom connector a short description

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

Scheme *

HTTPS HTTP

Host *

wiise.api.bc.dynamics.com

Base URL

/v2.0/

Security → Choose Basic authentication and add the Integration Username and Password from previous steps, in this case api

← Connector Name Process Item Journal

1. General > 2. Security > 3. Definition > 4. Test Swagger Editor ✓ Update connector ✗ Close

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 *

Basic authentication

✎ Edit

ⓘ Do NOT enter secrets here. These fields are used to configure display names for connections.

Basic authentication

Users will have to provide a valid user name and password before using this API

Parameter label *	Parameter name
api	username
asdf3MMNe:sIUew0eJnvDcw0CWttxd\$	password

Definition → + New action, in this case GetItemJournalLines

← Connector Name Process Item Journal

1. General > 2. Security > 3. Definition > 4. Test Swagger Editor ✓ Create connector ✕ Cancel

▼ 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.

! 1 GetItemJourn... ⋮

+ **New action**

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

▼ Policies (0)
Policies are used to change the behavior of actions and triggers through configuration. You can use one or more policies from a set of predefined templates.

+ **New policy**

General

Summary [Learn more](#)

Description [Learn more](#)

Operation ID *
This is the unique string used to identify the operation.

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

Request
It defines the pre-requirements needed in order to make a request. Describes a single operation parameter. A unique parameter is defined by a combination of a name and location.

+ Import from sample

Defintion → + Import from sample

Using the OData V4 URL ([https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company\('CRONUS%20AU'\)/nativeItemJournals](https://wiise.api.bc.dynamics.com/v2.0/56a76834-548f-4ace-865e-4c1a7874496e/Production/ODataV4/Company('CRONUS%20AU')/nativeItemJournals)) create URL variables

For reference:

{tenantid} = 56a76834-548f-4ace-865e-4c1a7874496e (can also be replaced with domain name)
 {environmentname} = Production (can also be replaced with sandbox environment name)
 {companyname} = 'CRONUS AU'

Verb* → Get

URL →

[https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}/ODataV4/Company\({companyname}\)/nativeItemJournals](https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}/ODataV4/Company({companyname})/nativeItemJournals)

Headers → leave blank

← Connector Name Process Item Journal

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

▼ 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.

! 1 GetItemJourn... ⋮

+ **New action**

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

General

Summary [Learn more](#)

Description [Learn more](#)

Operation ID *
This is the unique string used to identify the operation.

Visibility [Learn more](#)

✕

Import from sample

Verb *
 GET DELETE POST PUT HEAD OPTIONS
 PATCH

URL *

This is the request URL.

Headers
Headers separated by a new line, e.g.:
 Content-Type application/json
 Accept application/json
These are custom headers that are part of the request.

Import Close

Press Import

Result

Connector Name Process Item Journal

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

behavior of actions and triggers through configuration. You can use one or more policies from a set of predefined templates.

New policy

Request

It defines the pre-requirements needed in order to make a request. Describes a single operation parameter. A unique parameter is defined by a combination of a name and location.

Request + Import from sample

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

GET

URL *
This is the request URL.

`https://wise.api.bc.dynamics.com/v2.0/(tenantid)/(environmentname)/ODataV4/Company((compar`

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

* tenantid ... * environmentname ... * companyname ...

Press + Add default response

Response

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

default default

+ Add default response

Paste the Response from the Postman test in the Body, Press Import

Import from sample

Headers

Headers separated by a new line, e.g.:

```
Content-Type application/json
Accept application/json
```

These are custom headers that are part of the response.

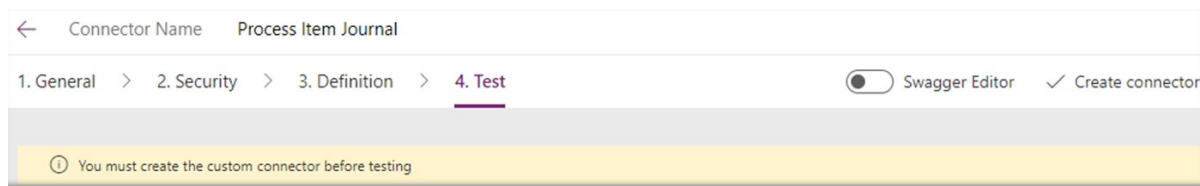
Body

```
{
  "Shortcut_Dimension_2_Code": "",
  "ShortcutDimCode3": "",
  "ShortcutDimCode4": "",
  "ShortcutDimCode5": "",
  "ShortcutDimCode6": "",
  "ShortcutDimCode7": "",
  "ShortcutDimCode8": "",
  "ItemDescription": "ATHENS Desk"
}
```

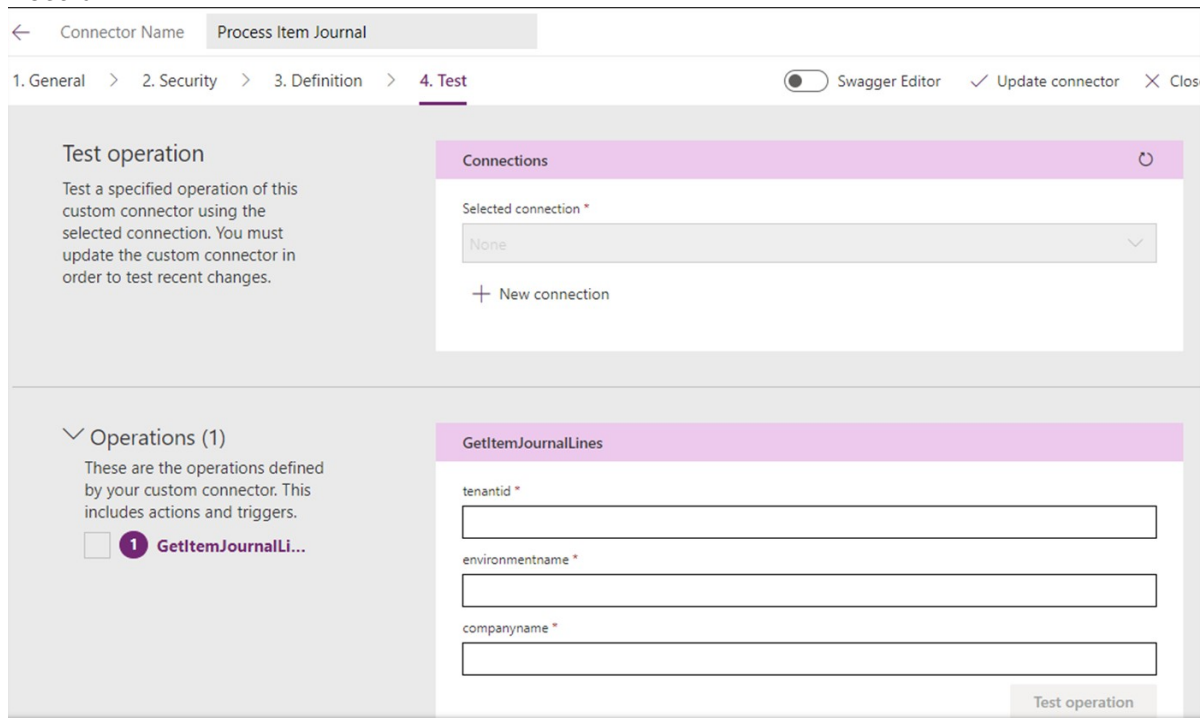
The payload that is available on the response. These are the tokens that will show up as the outputs in designer.

Import **Close**

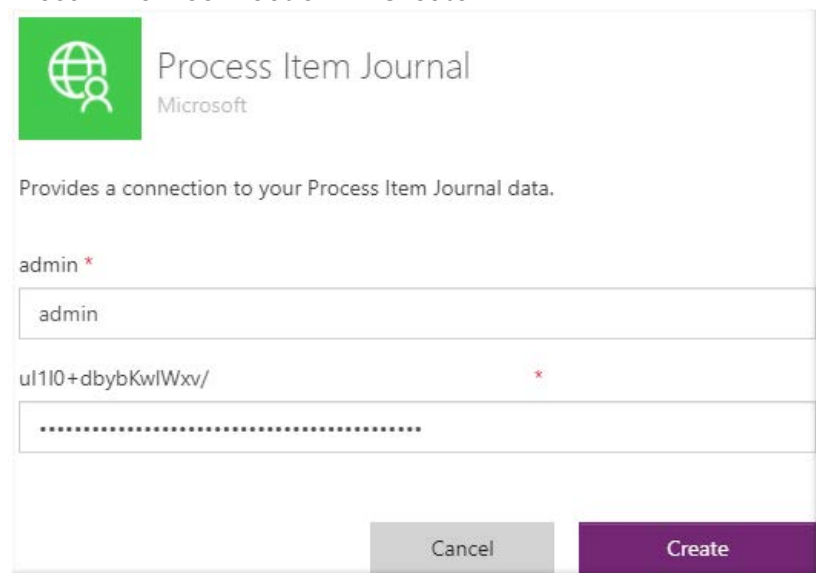
Test → Press Create Connector



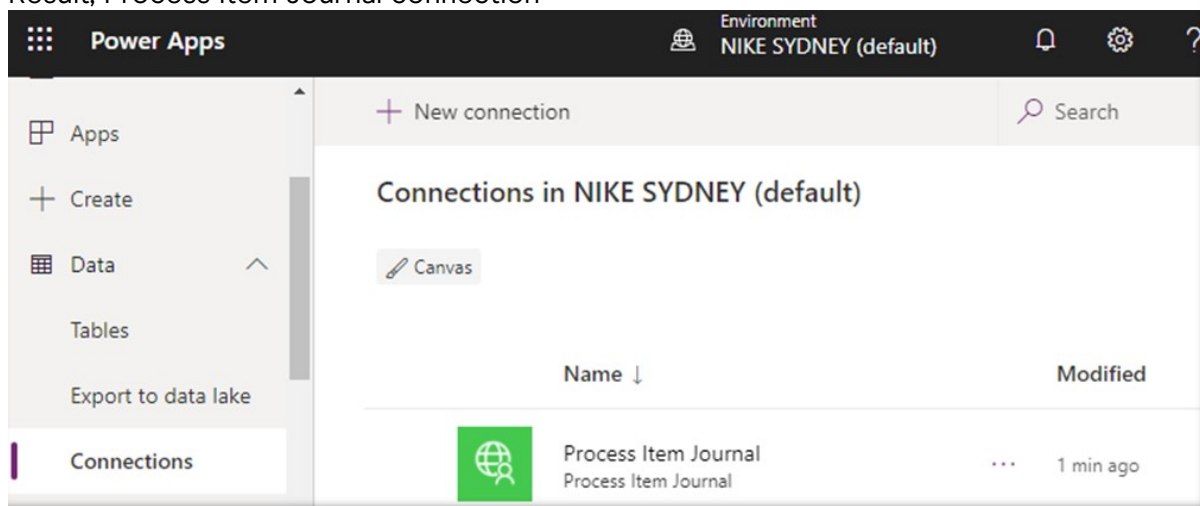
Result



Press + New connection → Create



Result, Process Item Journal connection



Press Custom Connectors (main menu) choose the newly created Custom Connector

Press Edit

Press 4. Test

Press Process Item Journals

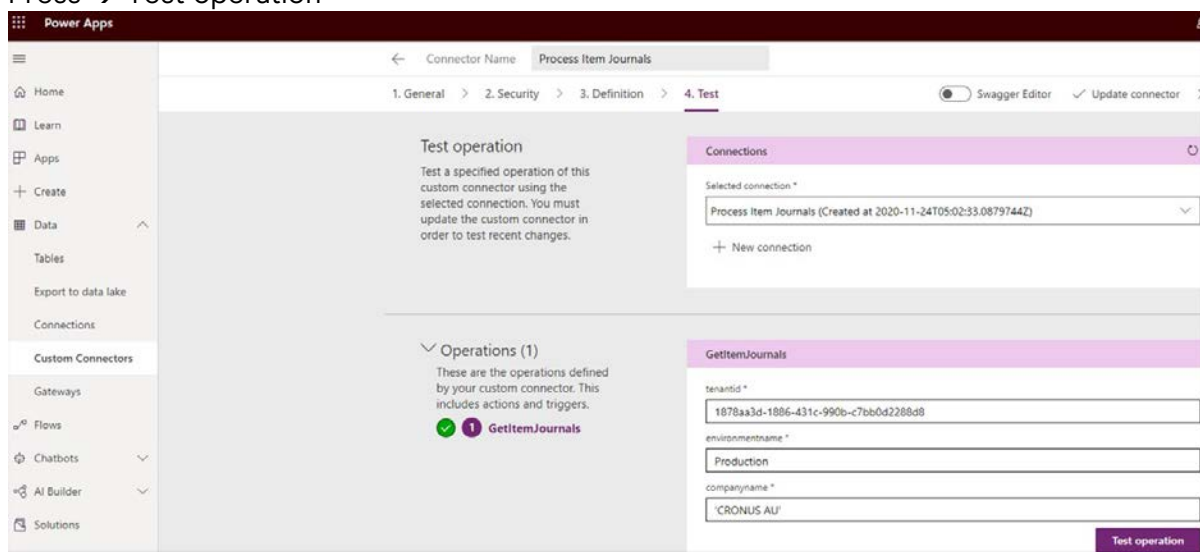
Add

tenantid = tenantGuid (can also be replaced with domain name)

environmentname = Production (can also be replaced with sandbox environment name)

companyname = 'CRONUS AU'

Press → Test operation



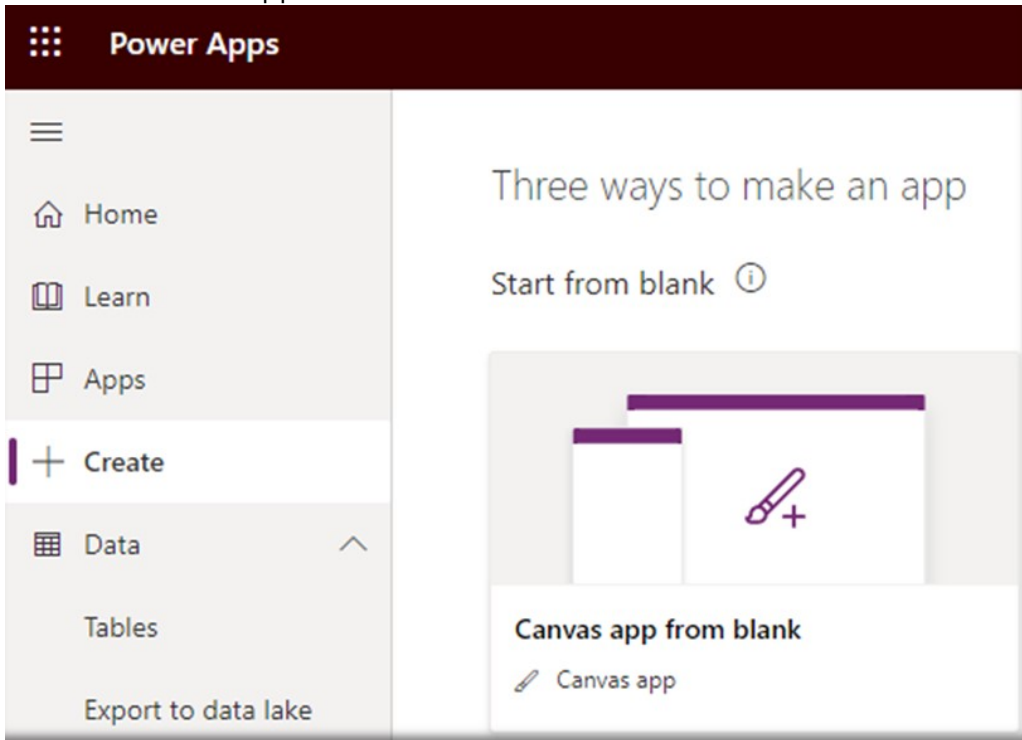
Result, in this case Status OK (200)

Request	Response
	<p>Status OK (200)</p> <p>Headers</p> <pre>{ "content-encoding": "gzip", "content-length": "651", "content-type": "application/json; odata.metadata=minimal", "date": "Tue, 24 Nov 2020 05:03:42 GMT", "ms-correlation-id": "f517e0d9-b68d-4ff1-b102-7a7575ec8710" }</pre> <p>Body</p> <pre>{ "@odata.context": "https://wiise.api.bc.dynamics.com/v2.0/1878aa3d-1886-431c-990b-c7bb0d22", "value": [{ "@odata.etag": "W/\\"JzQ00N6RTRIVHc1M0RDaTUvaWhoVWdBY1YyeUIKeHpFa25EVzJTbEptV", "Journal_Template_Name": "ITEM", "Journal_Batch_Name": "DEFAULT", "Line_No": 10000, "Posting_Date": "2019-04-01", "Document_Date": "2019-04-01", "Entry_Type": "Positive Adjmt.", "Price_Calculation_Method": "Not defined", "Document_No": "TEST", "External_Document_No": "" }] }</pre>

Create a Power App

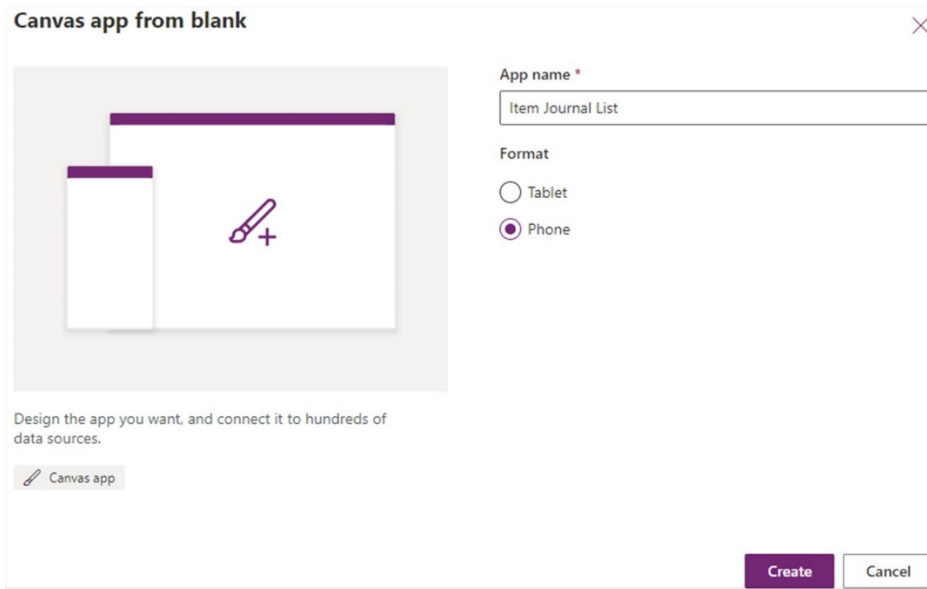
Item Journal List

Create → Canvas app from blank

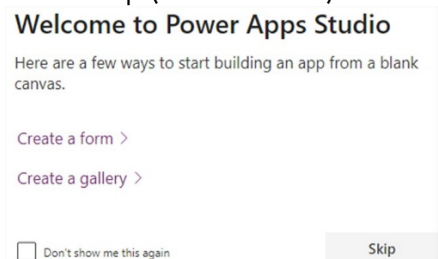


Enter a name, in this case Item Journal List

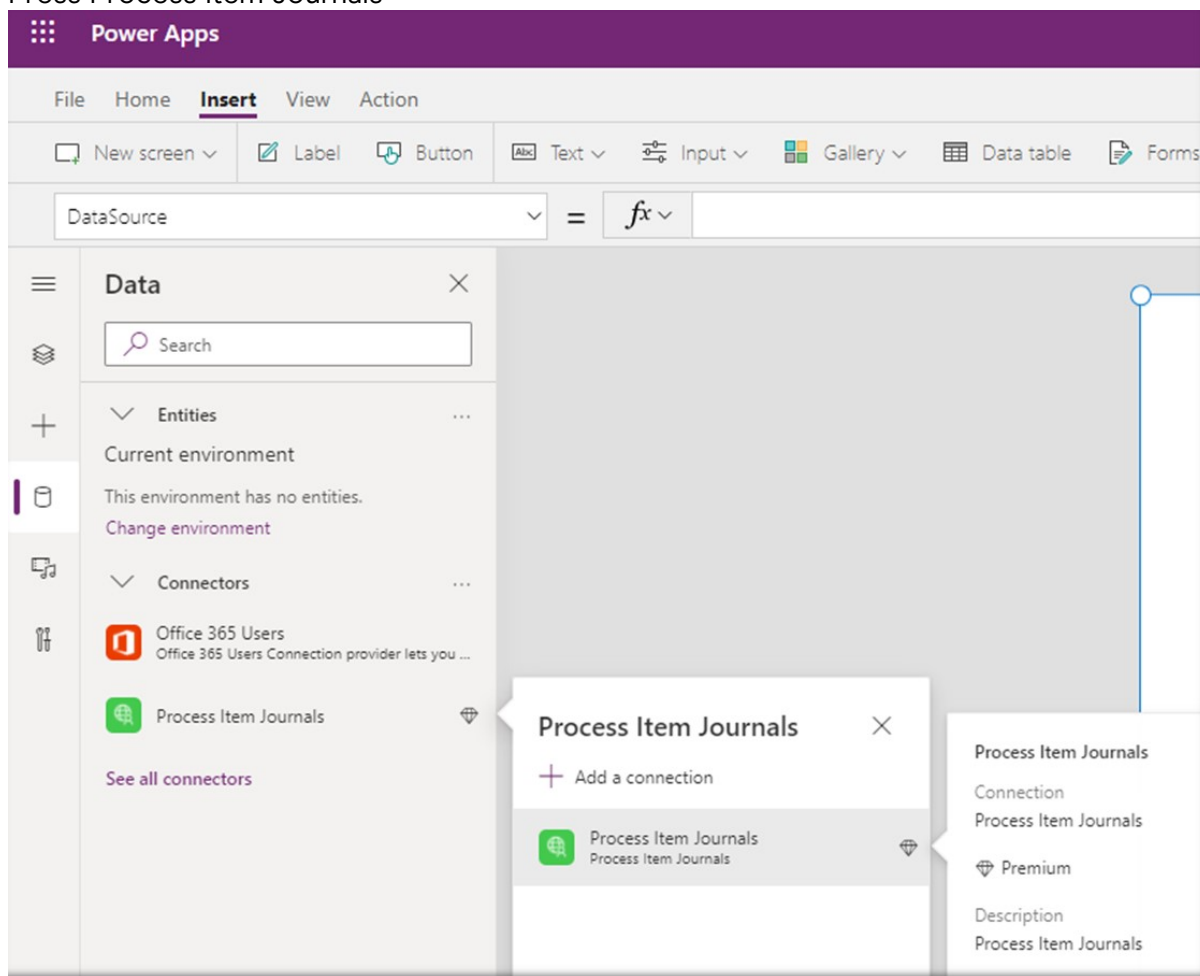
Press Create



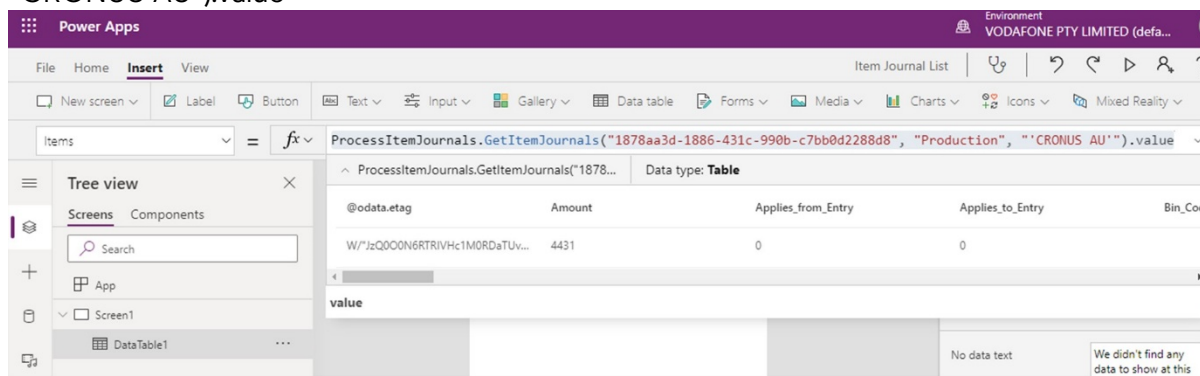
Press Skip (blank canvas)



Data → Connectors
 Press Process Item Journals

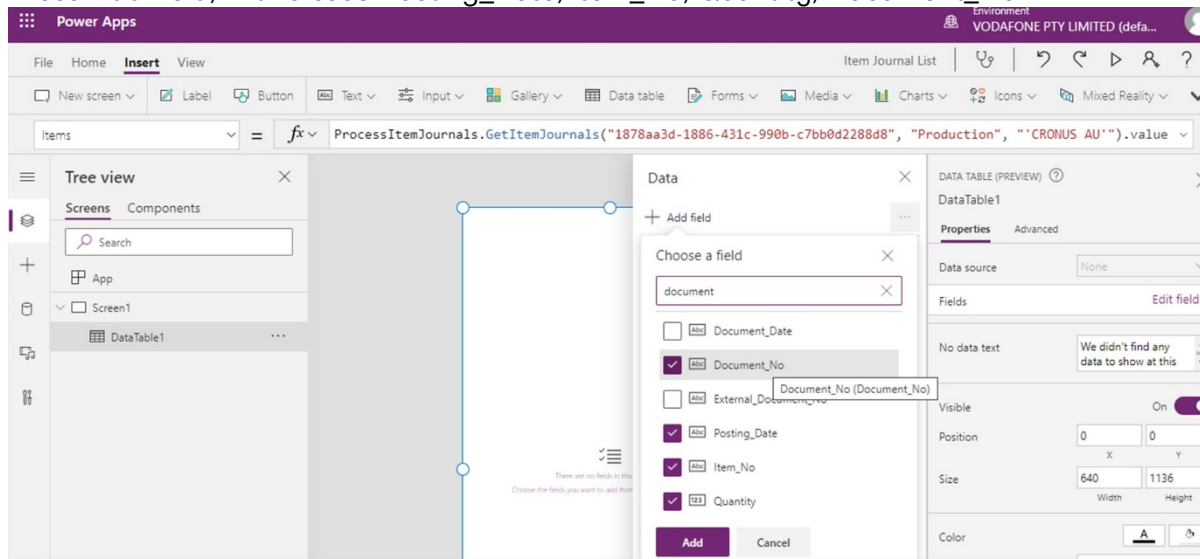


In the main form Add Data table
 Press DataTable1 and add the formula (same values from previous step)
 ProcessItemJournals.GetItemJournals("1878aa3d-1886-431c-990b-c7bb0d2288d8", "Production",
 "'CRONUS AU'").value

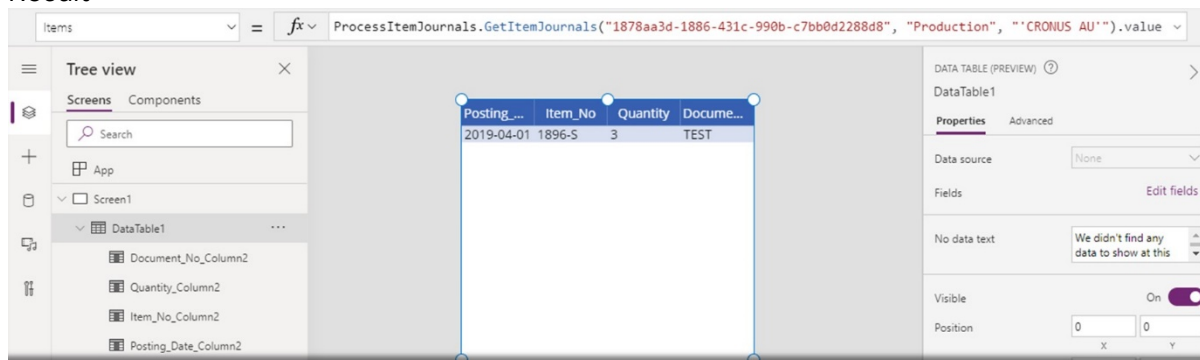


Edit fields

Press Add field, in this case Posting_Date, Item_No, Quantity, Document_No

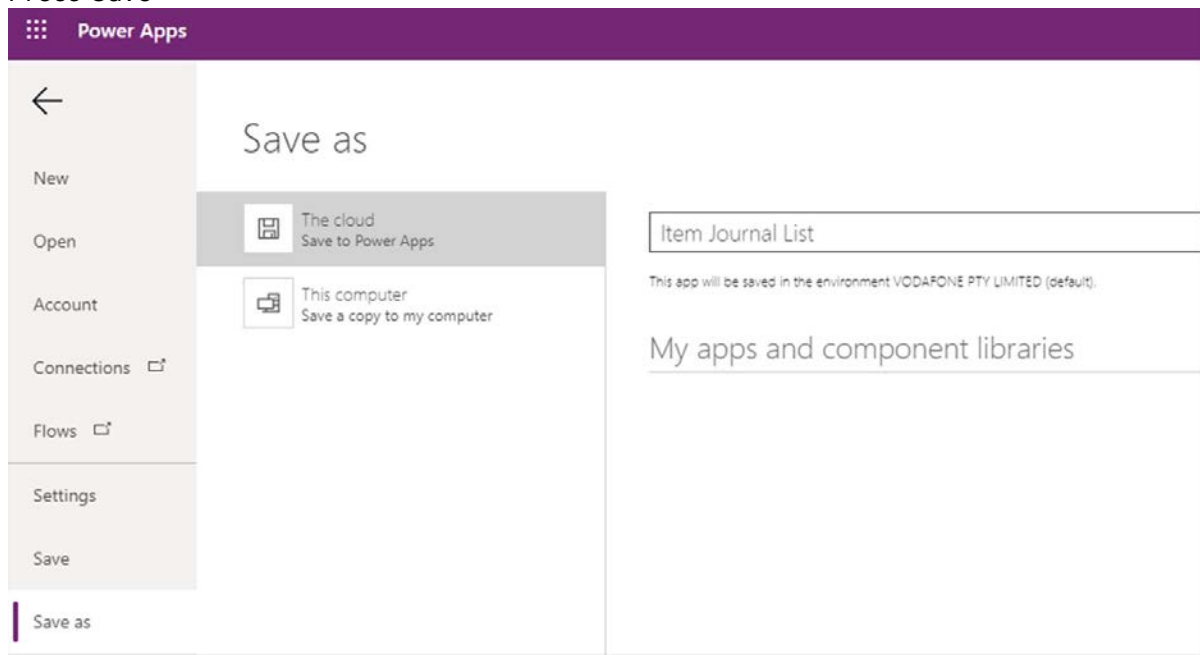


Result



File → Save as → The Cloud

Press Save

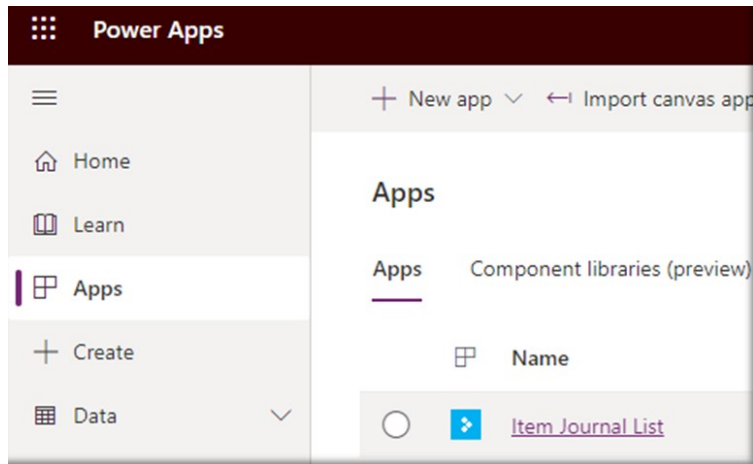


Press Close

Return to <https://make.powerapps.com/>

Apps

Press Item Journal List



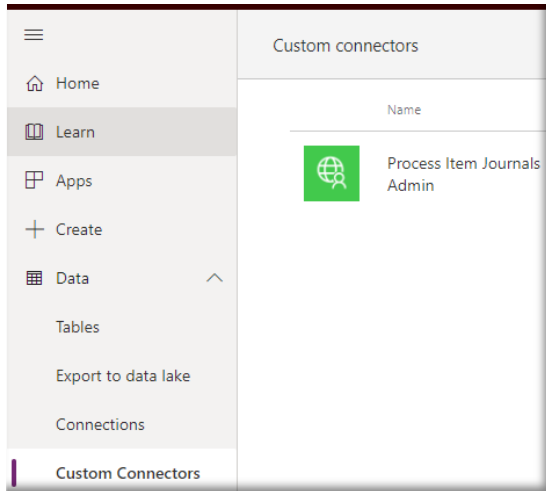
Result

The screenshot shows a data table within the Power Apps interface. The table has four columns: 'Posting_...', 'Item_No', 'Quantity', and 'Docume...'. The first row contains the values '2019-04-01', '1896-S', '3', and 'TEST'.

Posting_...	Item_No	Quantity	Docume...
2019-04-01	1896-S	3	TEST

Submit Item Journal Line

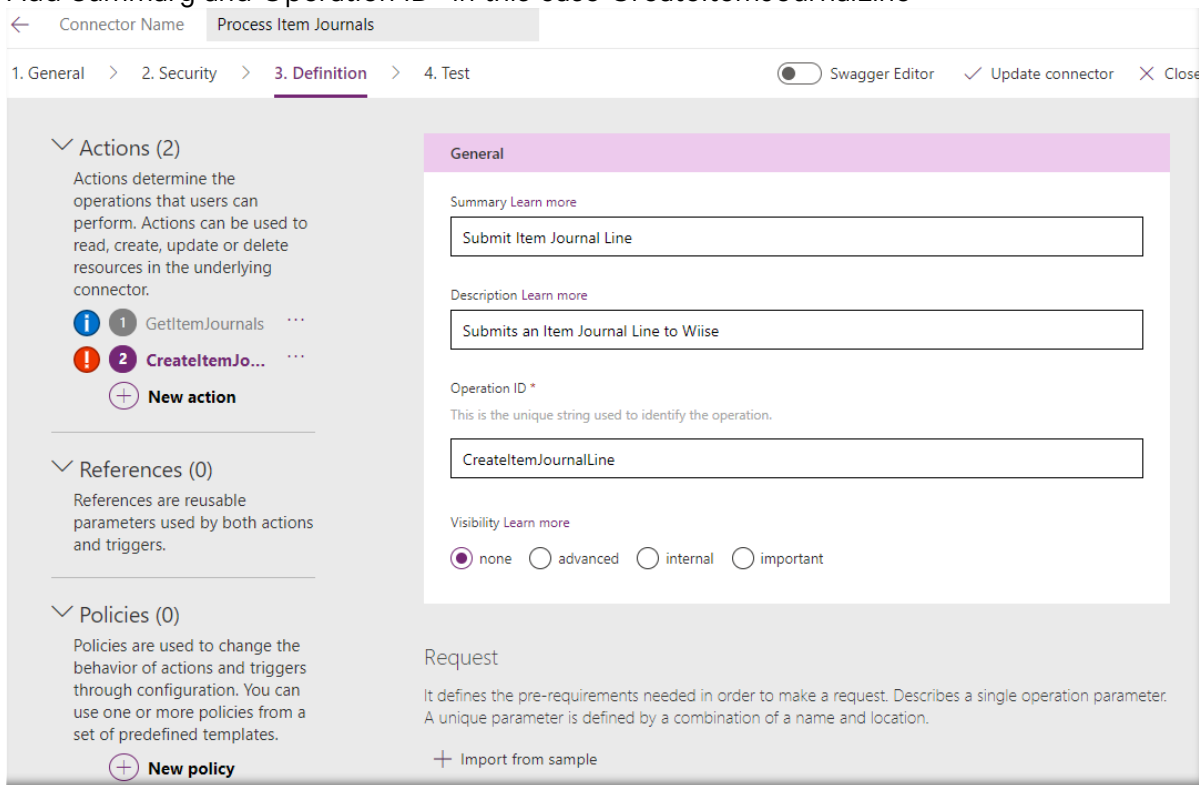
Return to <https://make.powerapps.com/Data> → Custom Connectors
 Press Process Item Journals → Edit



3. Definition

Press + New action

Add Summary and Operation ID* in this case CreateItemJournalLine



Press + Import from sample

```
Add
Verb* POST
URL*
https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}/ODataV4/Company({companyname})/nativeItemJournals
Body
{
  "Journal_Template_Name": "ITEM",
  "Journal_Batch_Name": "DEFAULT",
  "Entry_Type": "Positive Adjmt.",
  "Document_No": "CYCLECOUNT",
  "Item_No": "1896-S",
  "Quantity": 4
}
```

Note: Journal Template Name and Batch Name must exist in Wiise.
As in previous steps this operation should be run in Postman in order to get a response used in the next step

Press Import

Import from sample

Verb *

GET DELETE POST PUT HEAD OPTIONS

PATCH

URL *

https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}

This is the request URL.

Headers

Headers separated by a new line, e.g.:
Content-Type application/json
Accept application/json

These are custom headers that are part of the request.

Body

```
{
  "Journal_Template_Name": "ITEM",
  "Journal_Batch_Name": "DEFAULT",
  "Entry_Type": "Positive Adjmt.",
  "Document_No": "CYCLECOUNT",
  "Item_No": "1896-S",
  "Quantity": 4
}
```

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

Import Close

Result

Press + Add default response

Request
+ Import from sample

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

POST

URL *
This is the request URL

https://wiise.api.bc.dynamics.com/v2.0/{tenantid}/{environmentname}/ODataV4/Company({compa

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

* tenantid ...
* environmentname ...
* companyname ...

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

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.

body ...

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

default
default

+ Add default response

Body → Add the response of the POST operation from postman

Press Import

Import from sample
×

Headers

Headers separated by a new line, e.g.:
 Content-Type application/json
 Accept application/json

These are custom headers that are part of the response.

Body

```

{
  "Shortcut_Dimension_1_Code": "",
  "Shortcut_Dimension_2_Code": "",
  "ShortcutDimCode3": "",
  "ShortcutDimCode4": "",
  "ShortcutDimCode5": "",
  "ShortcutDimCode6": "",
  "ShortcutDimCode7": "",
  "ShortcutDimCode8": "",
  "ItemDescription": "ATHENS Desk"
}

```

The payload that is available on the response. These are the tokens that will show up as the outputs in designer.

Import
Close

Result

Validation

This helps you identify potential issues with this action.

Validation ✔

Validation succeeded.

← Security
Test →

Operations (2)

Press 2 CreateItemJournalLine

Add variable information as per previous steps

Press Test Operation

← Connector Name Process Item Journals
Swagger Editor Update connector Close

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

Test operation

Test a specified operation of this custom connector using the selected connection. You must update the custom connector in order to test recent changes.

Connections 🔄

Selected connection *

Process Item Journals (Created at 2020-11-24T05:02:33.0879744Z)
▼

+ New connection

CreateItemJournalLine

tenantid *

1878aa3d-1886-431c-990b-c7bb0d2288d8

environmentname *

production

companyname *

'CRONUS AU'

Raw Body

On

```
{
  "Journal_Template_Name": "ITEM",
  "Journal_Batch_Name": "DEFAULT",
  "Entry_Type": "Positive Adjmt",
  "Document_No": "CYCLECOUNT",
  "Item_No": "1896-S",
  "Quantity": 4
}
```

Operations (2)

These are the operations defined by your custom connector. This includes actions and triggers.

- 1 GetItemJournals
- 2 CreateItemJourn...

Result

Request
Response

Status
Created (201)

Headers

```
{
  "content-encoding": "gzip",
  "content-length": "650",
  "content-type": "application/json; odata.metadata=minimal",
  "date": "Tue, 24 Nov 2020 12:44:59 GMT"
}
```

Body

```
{
  "@odata.context": "https://wiise.api.bc.dynamics.com/v2.0/1878aa3d-1886-431c-990b-c7bb0d22",
  "@odata.etag": "W/\^JzQ003hmdnd1K2xlZ3Jmb0NuZUdaVVc4QW55WCsvZVo2WVNweTRiY1BLk",
  "Journal_Template_Name": "ITEM",
  "Journal_Batch_Name": "DEFAULT",
  "Line_No": 20000,
  "Posting_Date": "2019-04-01",
  "Document_Date": "2019-04-01",
  "Entry_Type": "Positive Adjmt.",
  "Price_Calculation_Method": "Not defined",
  "Document_No": "CYCLECOUNT",
  "External_Document_No": "",
  "Item_No": "1896-S",
  "Variant_Code": ""
}
```

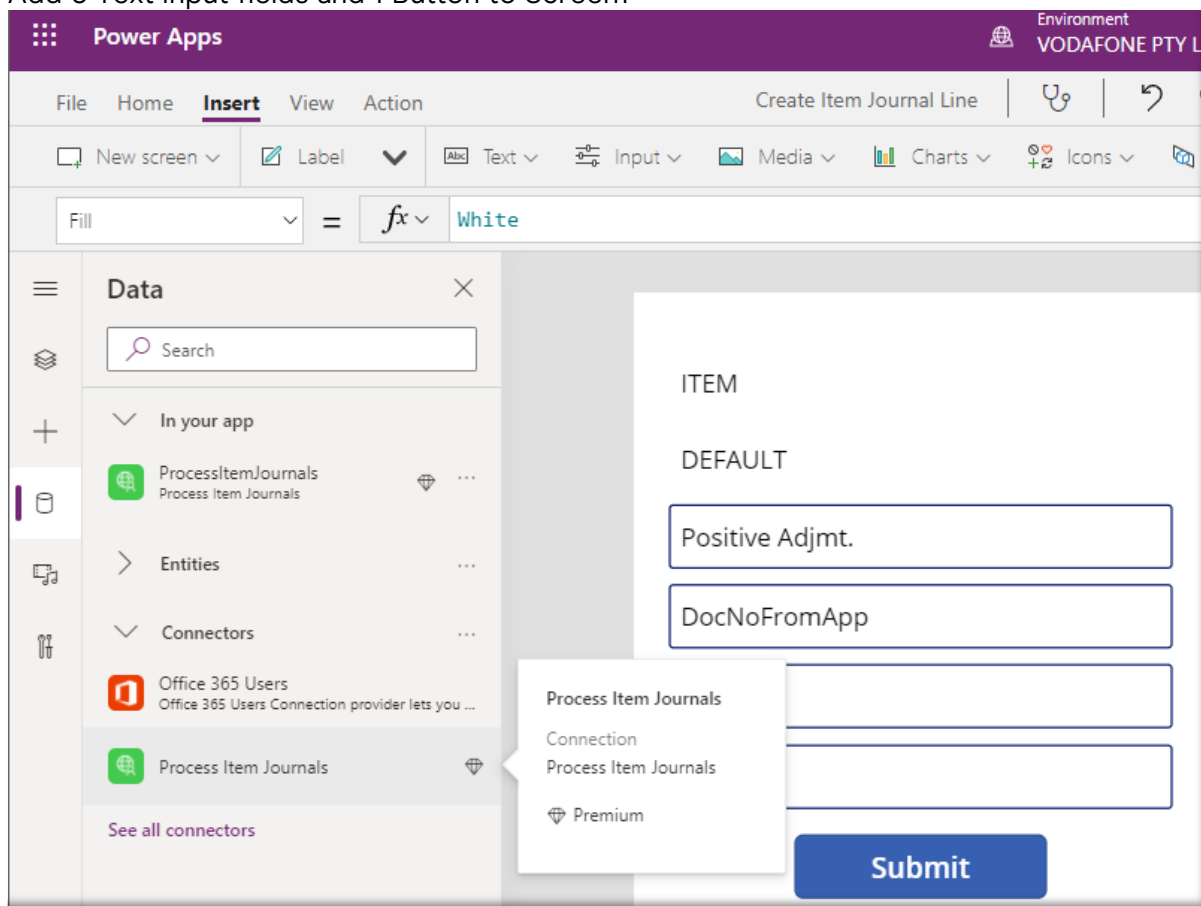
Schema validation

Validation succeeded

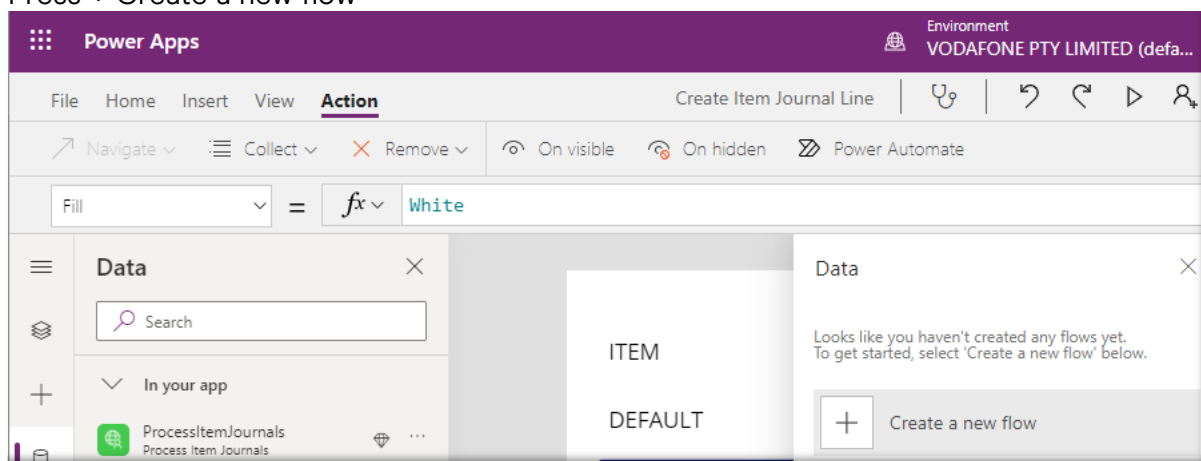
+ Create → Canvas app from blank add App Name, in this case Create Item Journal Line

The screenshot shows the Power Apps interface with the 'Create' option selected in the left sidebar. A dialog box titled 'Canvas app from blank' is open. The dialog has a close button (X) in the top right corner. It contains a preview of a blank canvas with a pencil icon and a plus sign. To the right of the preview, there is a text input field for 'App name' containing the text 'Create Item Journal Line'. Below the input field, there is a 'Format' section with two radio buttons: 'Tablet' (unselected) and 'Phone' (selected). At the bottom right of the dialog, there are two buttons: 'Create' and 'Cancel'.

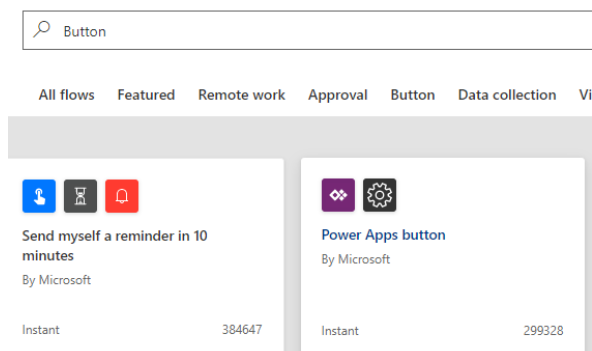
Data → Add ProcessItemJournals
 Add 6 Text input fields and 1 Button to Screen1



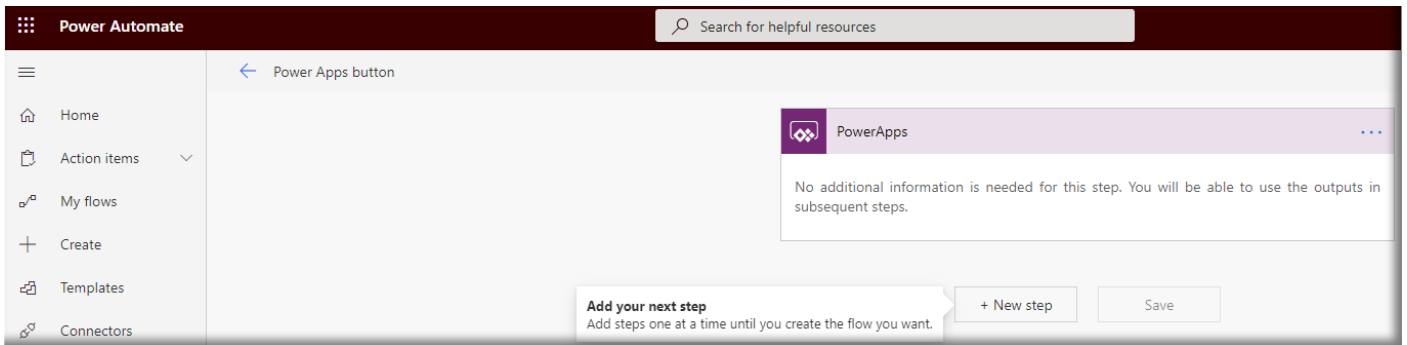
Action → PowerAutomate
 Press + Create a new flow



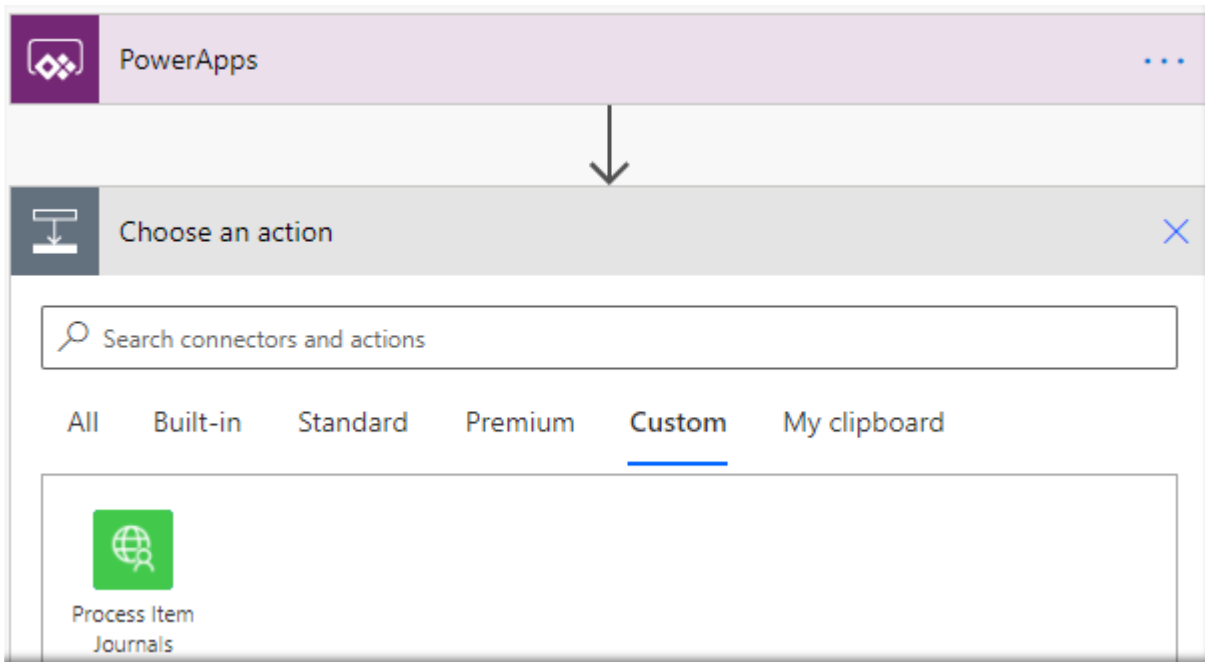
Search Button → Power Apps Button (by Microsoft)



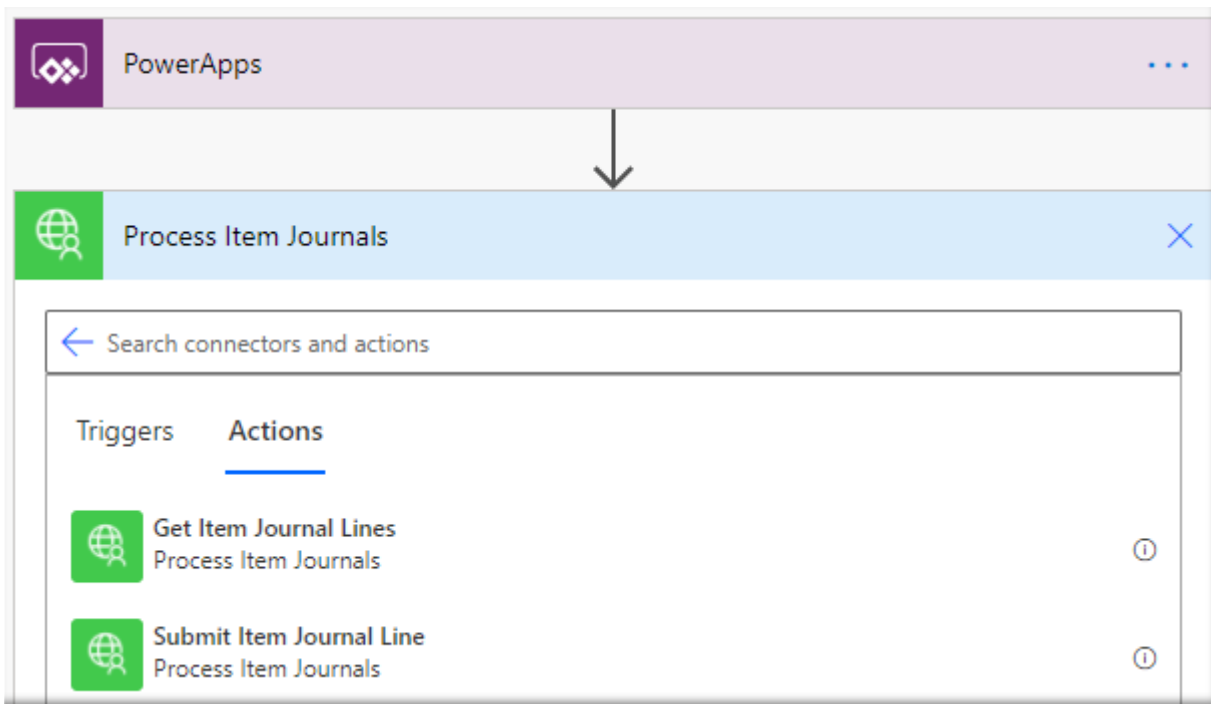
Press + New step



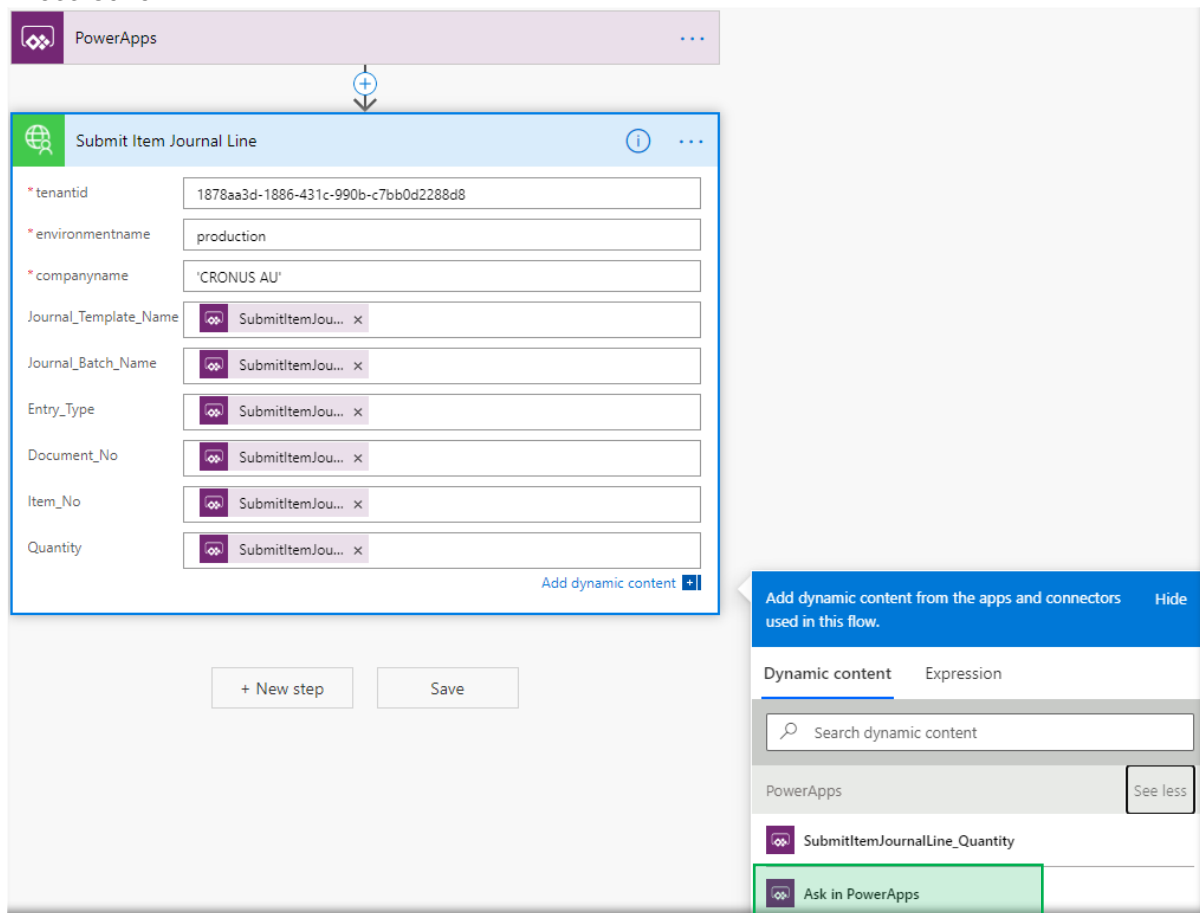
Custom → Process Item Journals



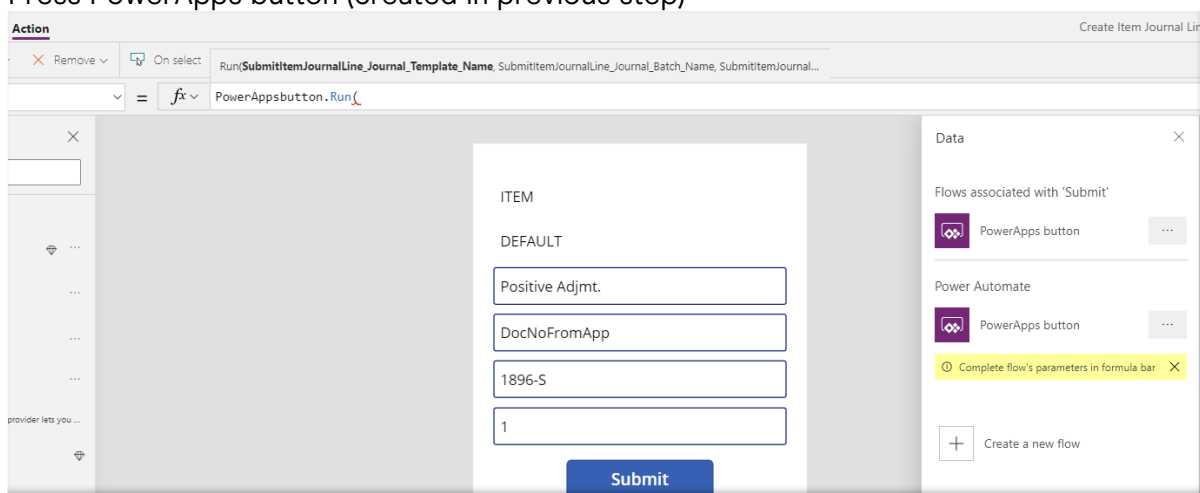
Press Submit Item Journal Line



Add details from previous steps where mandatory
 For the other 6 fields (defined in the previous step)
 Press Ask in PowerApps in each of the other fields
 Result
 Press Save

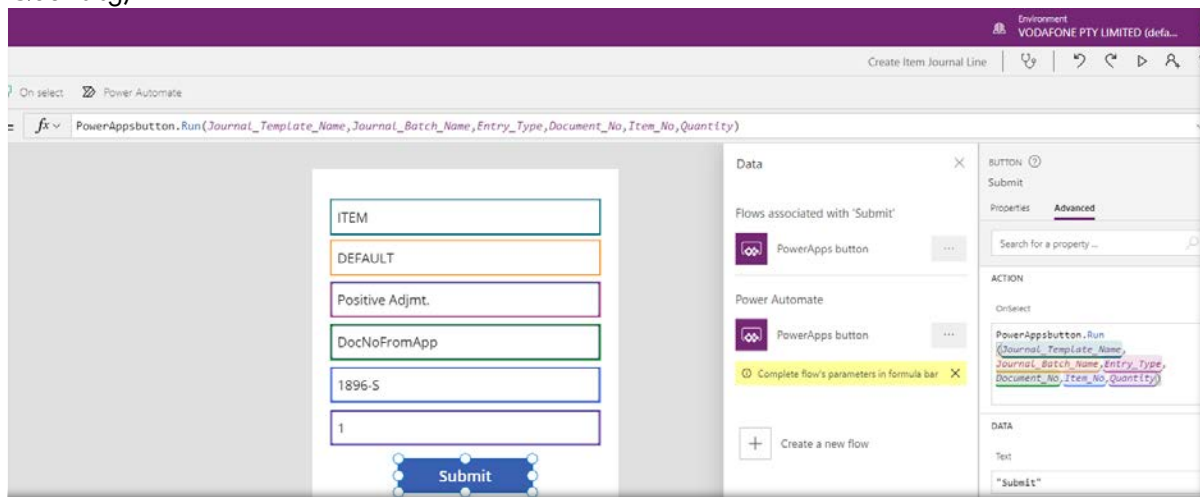


Return to the PowerApp
 Click on the button, in this case Submit
 Press Power Automate
 Press PowerApps button (created in previous step)

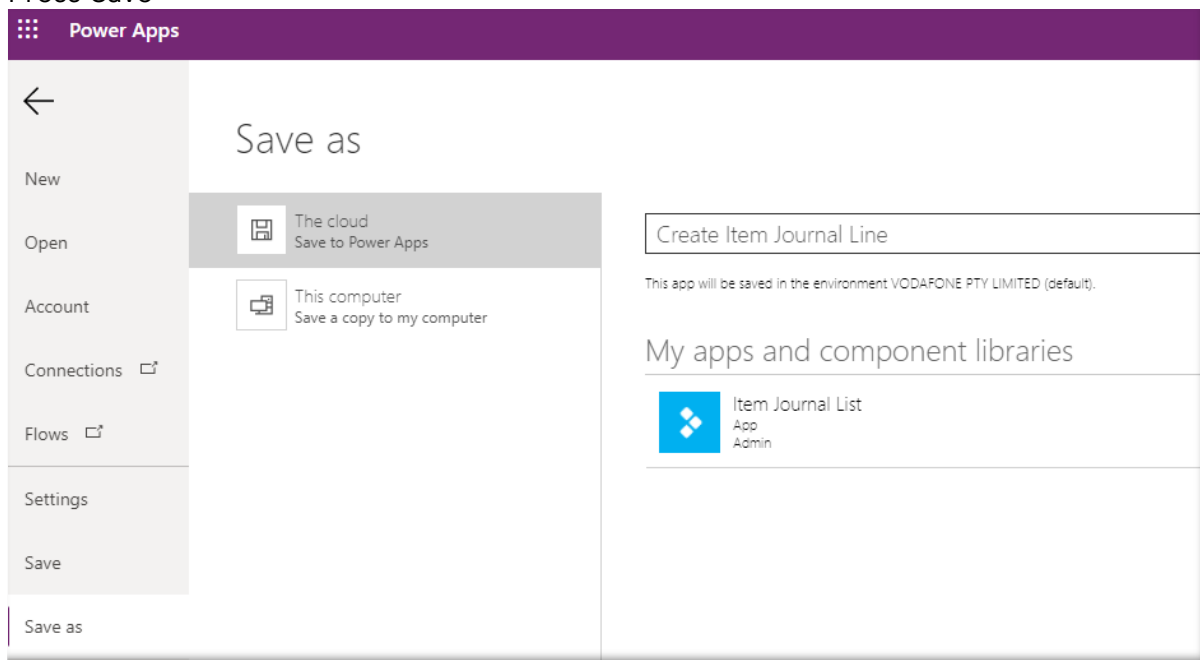


Add the variable names of the Text input fields added to the Action → OnSelect

PowerAppsbutton.Run(Journal_Template_Name,Journal_Batch_Name,Entry_Type,Document_No,Item_No,Quantity)



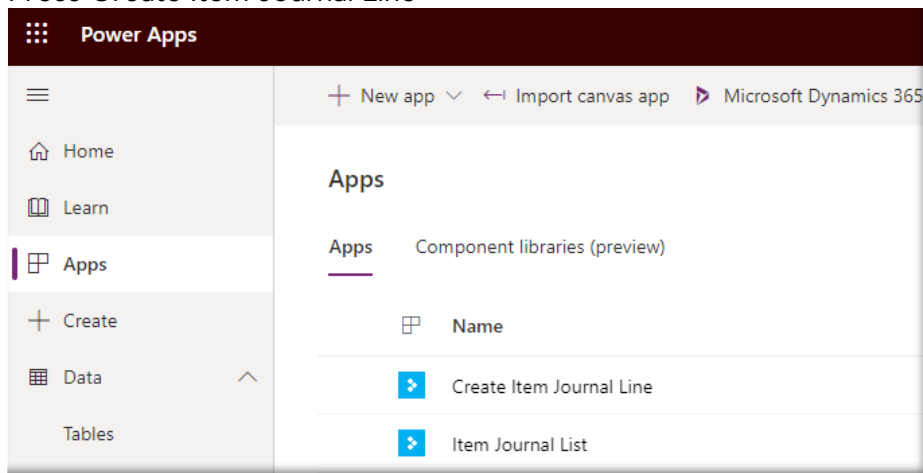
File → Save As → The cloud
Press Save



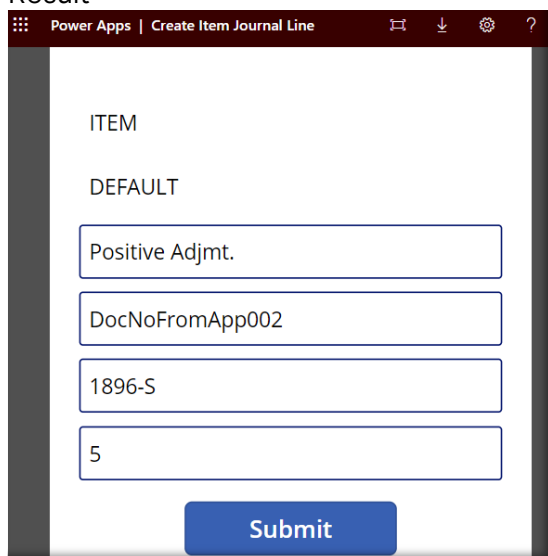
Return to <https://make.powerapps.com/>

Apps

Press Create Item Journal Line



Result



Verify Results in Wise

ITEM JOURNALS ✓ SAVED

✕ Reminder: your work date is 4/1/2019 Use today | Change to... | Turn off reminder

Batch Name: DEFAULT

Posting Date	Entry Type	Document No.	Item No.	Description	Loc... Code	Quantity
4/1/2019	Positive Adjmt.	TEST	1896-S	ATHENS Desk		3
4/1/2019	Positive Adjmt.	CYCLECOUNT	1896-S	ATHENS Desk		4
4/1/2019	Positive Adjmt.	DOCNOFROMAPP	1896-S	ATHENS Desk		1
4/1/2019	Positive Adjmt.	DOCNOFROMAPP002	1896-S	ATHENS Desk		5