Materials Management App

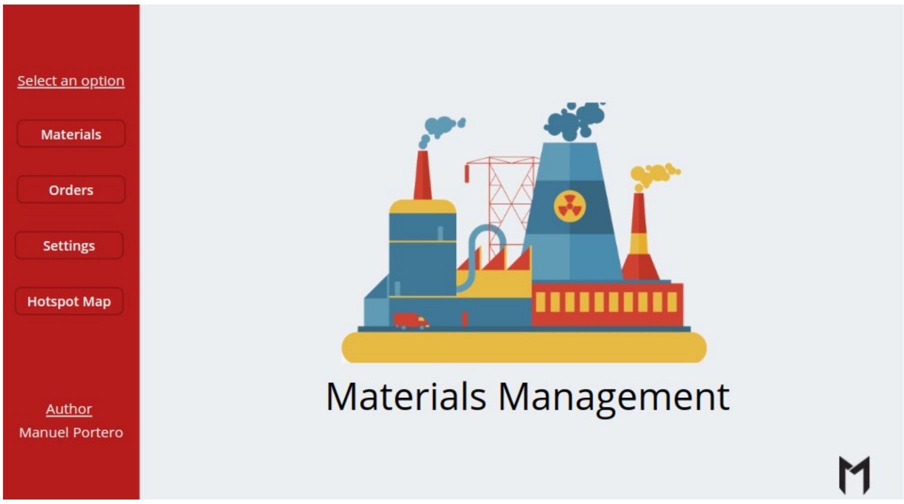
PowerApp Documentation

28/08/2022

Author: Manuel Portero Leiva

Introduction

This document has the purpose to explain the different parts of the Material Management App, its code and functionalities, for understanding and replication purposes. The different parts of the architecture solution are show below.

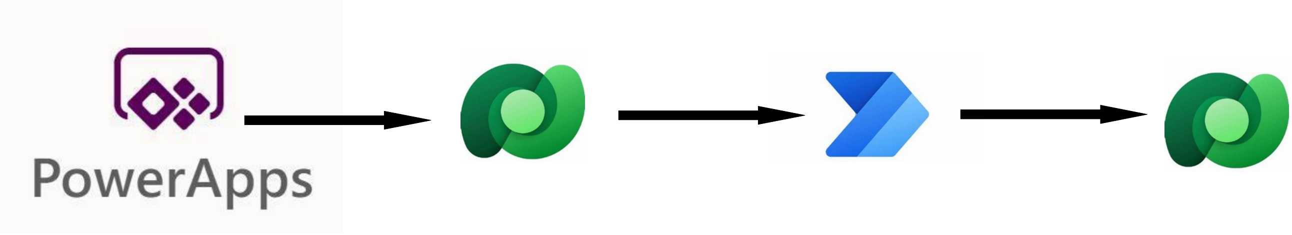


Picture 1: Materials Management Main Layout

Architecture

The composition of the architecture starts in the PowerApp, The travel requests are added to different Dataverse tables and are represented in the diffent screens. After any adding or modification of the Orders Table, a flow is activated and a new location updates the material pointed in the order's row.

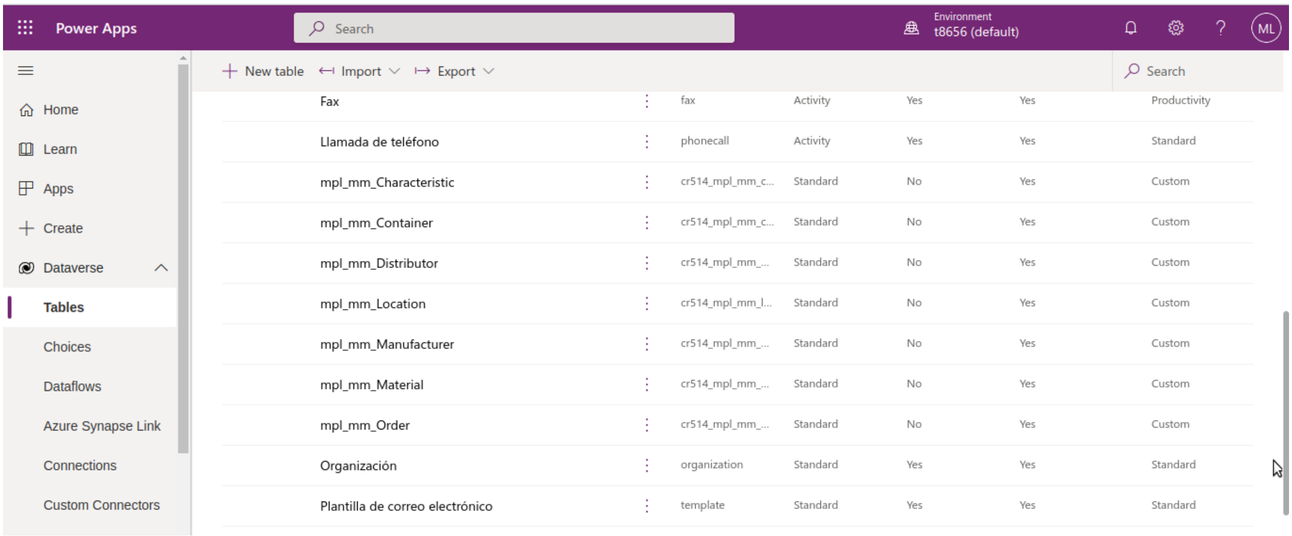
A full diagram of the solution is shown below.



Picture 2: Materials Manager App solution Architecture

Dataverse Database

The Materials Management App solution will have a Dataverse database as a Data storage and manipulation, a sample of the database is shown below.

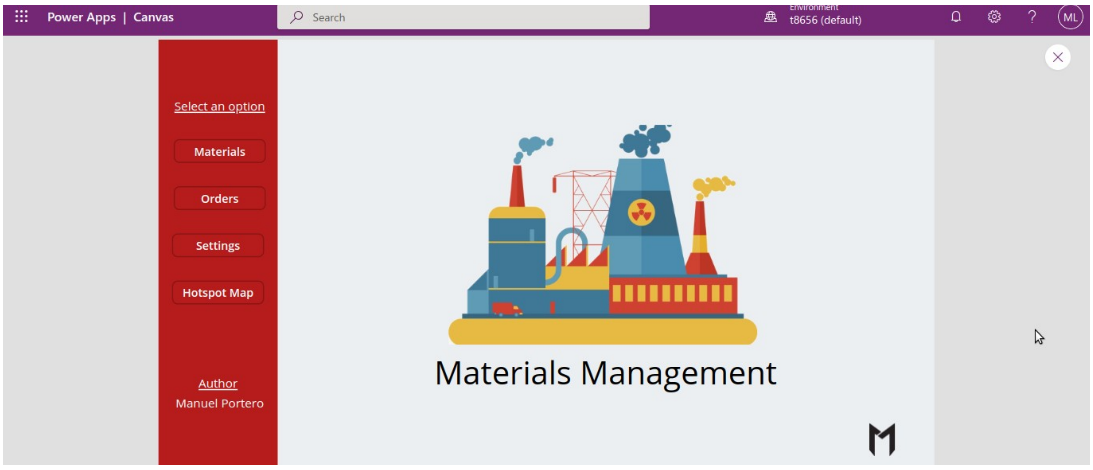


Fax	⋮	fax	Activity	Yes	Yes	Productivity
Llamada de teléfono	⋮	phonecall	Activity	Yes	Yes	Standard
mpl_mm_Characteristic	⋮	cr514_mpl_mm_c...	Standard	No	Yes	Custom
mpl_mm_Container	⋮	cr514_mpl_mm_c...	Standard	No	Yes	Custom
mpl_mm_Distributor	⋮	cr514_mpl_mm_...	Standard	No	Yes	Custom
mpl_mm_Location	⋮	cr514_mpl_mm_l...	Standard	No	Yes	Custom
mpl_mm_Manufacturer	⋮	cr514_mpl_mm_...	Standard	No	Yes	Custom
mpl_mm_Material	⋮	cr514_mpl_mm_...	Standard	No	Yes	Custom
mpl_mm_Order	⋮	cr514_mpl_mm_...	Standard	No	Yes	Custom
Organización	⋮	organization	Standard	Yes	Yes	Standard
Plantilla de correo electrónico	⋮	template	Standard	Yes	Yes	Standard

Picture 3: Dataverse database sample

Main Screen

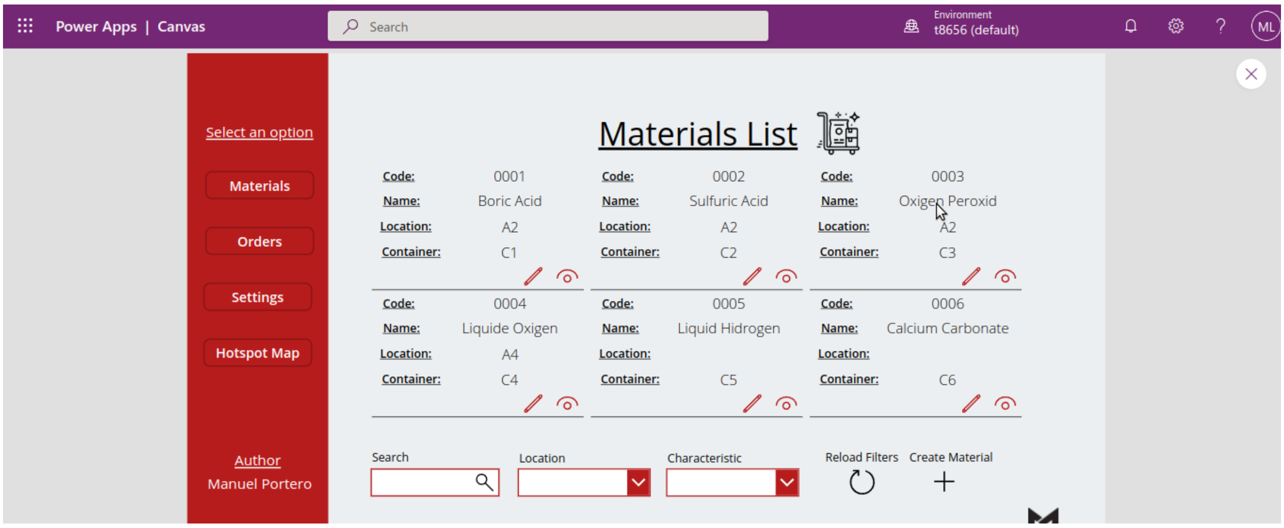
The main screen is composed by the main landscape picture and the navigation buttons to the others screens.



Picture 4: Materials Management App main Screen

Materials Screen

The Materials Screen has a Gallery displaying the Materials of the factory, a name search text input field, two dropdown filters and a reload button and a create button.



Materials List

Code: 0001
Name: Boric Acid
Location: A2
Container: C1

Code: 0002
Name: Sulfuric Acid
Location: A2
Container: C2

Code: 0003
Name: Oxigen Peroxid
Location: A2
Container: C3

Code: 0004
Name: Liquide Oxigen
Location: A4
Container: C4

Code: 0005
Name: Liquid Hidrogen
Location:
Container: C5

Code: 0006
Name: Calcium Carbonate
Location:
Container: C6

Picture 5: Materials Screen

The code of the materials gallery is the following one :

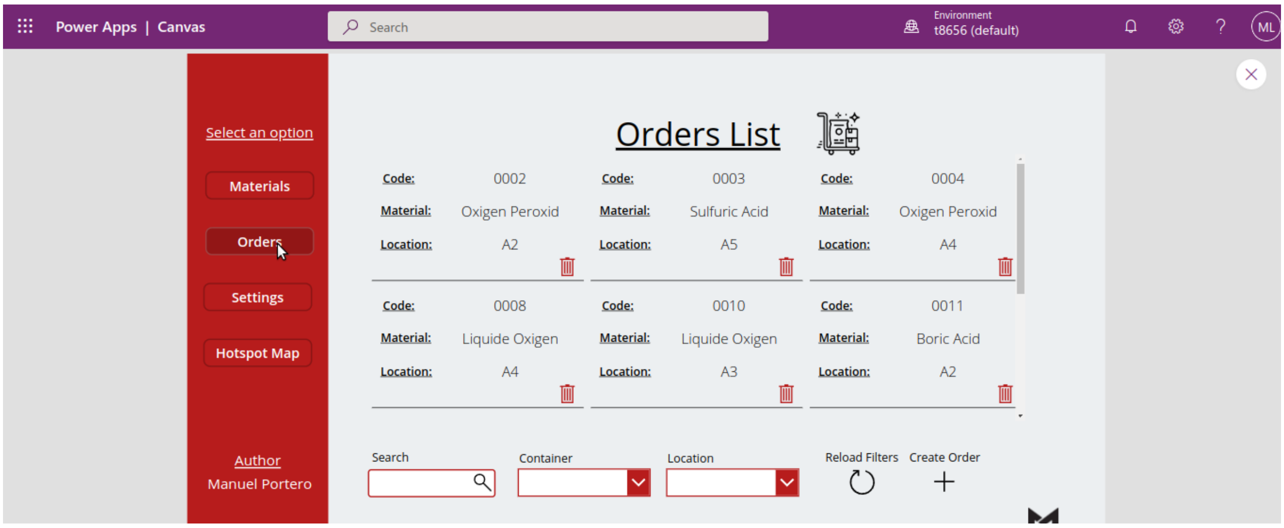
```
Filter(
    Search(
        mpl_mm_Materials;
        SearchTextInput8.Text;
        "cr514_code";
        "cr514_name"
    );
    If(
        LocationDropdown.Selected.Name = Blank();
        true;
        Location.Name = LocationDropdown.Selected.Name
    );
     If(
        CharacteristicDropdown.Selected.Name = Blank();
        true;
        Characteristic.Name = CharacteristicDropdown.Selected.Name
    )
)
```

The code of the reload filters button is the following one :

```
Reset(SearchTextInput8);;
Reset(LocationDropdown);;
Reset(CharacteristicDropdown);;
Notify(
    "Filters reseted successfully";
    Success
)
```

Orders Screen

The Orders Screen has a Gallery displaying the Orders of the factory, a name search text input field, two dropdown filters and a reload button and a create button.



Code: 0002
Material: Oxigen Peroxid
Location: A2

Code: 0003
Material: Sulfuric Acid
Location: A5

Code: 0004
Material: Oxigen Peroxid
Location: A4

Code: 0008
Material: Liquide Oxigen
Location: A4

Code: 0010
Material: Liquide Oxigen
Location: A3

Code: 0011
Material: Boric Acid
Location: A2

Picture 5: Orders Screen

)

The Code of the Orders Gallery is the following one :

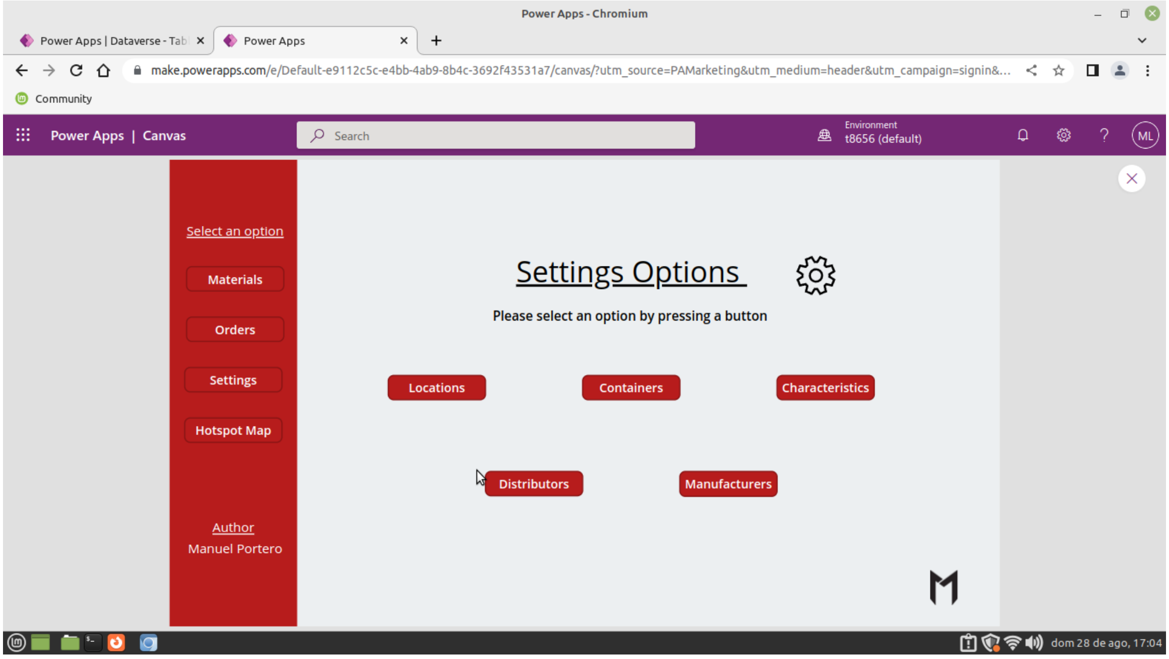
```
Filter(
    Search(
        mpl_mm_Orders;
        OrdersTextInput.Text;
        "cr514_code"
    );
    If(
        ContainerDropdown2.Selected.Code = Blank();
        true;
        Code = ContainerDropdown2.Selected.Code
    );
    If(
        LocationDropdown2.Selected.Name = Blank();
        true;
        Location = LocationDropdown2.Selected.Name
    )
)
```

The reload filters button is the following one :

```
Reset(OrdersTextInput);;
Reset(ContainerDropdown2);;
Reset(LocationDropdown2);;
Notify(
    "Filters reseted successfully";
    Success
)
```

Settings Screens

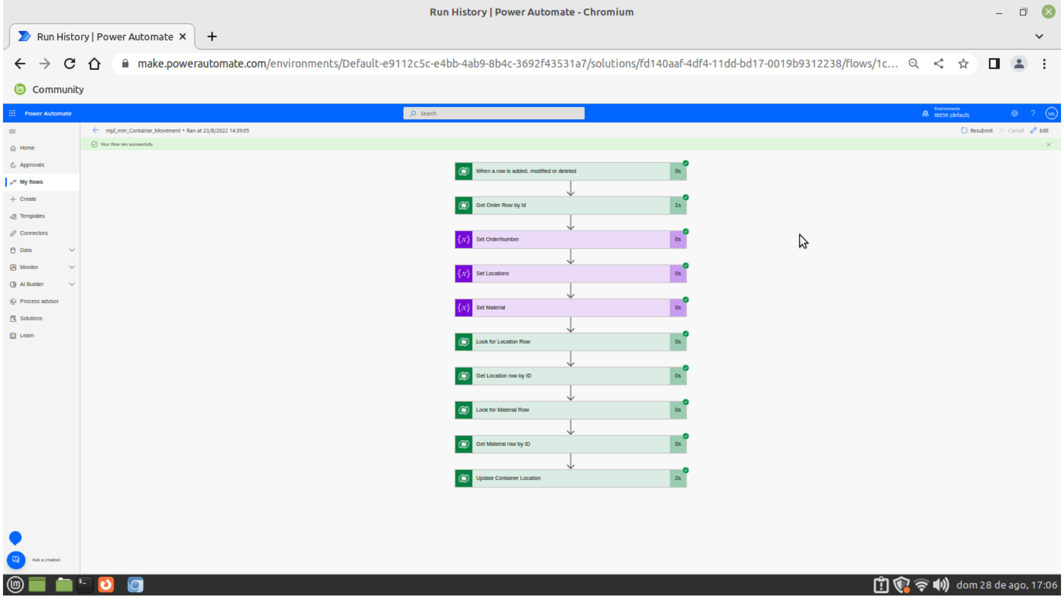
The settings screens are composed by a gallery, a text search input and a create button (not text search for location and containers).



Picture 6: Setting Screen

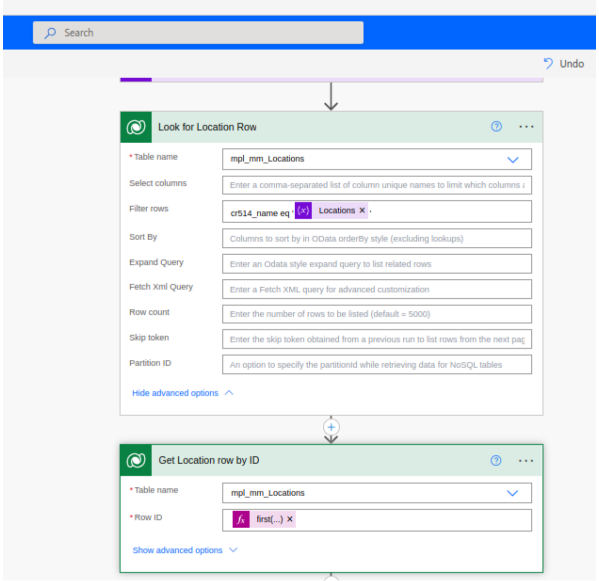
Dataverse Flow

Every time a row is created in dataverse the materials table will be updated with the new order's location.



Picture 6: Dataverse flow

The flow will take the order list and search in Get Orders row by id taking al the row elemens, after that the ordernumber, location and material will be setted with the Get Orders row by id outputs. The Look for Location row will list all the elements that match with the locations variable. The Get location row by id will take the first value output of the Look for Location row action.



Look for Location Row

* Table name: mpl_mm_Locations

Select columns: Enter a comma-separated list of column unique names to limit which columns a

Filter rows: cr514_name eq ' Locations '

Sort By: Columns to sort by in OData orderBy style (excluding lookups)

Expand Query: Enter an Odata style expand query to list related rows

Fetch Xml Query: Enter a Fetch XML query for advanced customization

Row count: Enter the number of rows to be listed (default = 5000)

Skip token: Enter the skip token obtained from a previous run to list rows from the next pag

Partition ID: An option to specify the partitionId while retrieving data for NoSQL tables

Get Location row by ID

* Table name: mpl_mm_Locations

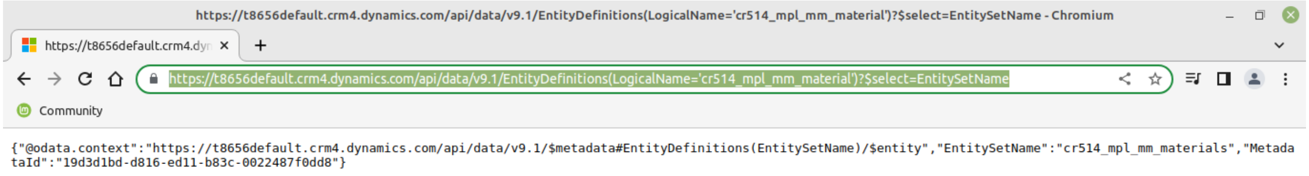
* Row ID: first(...)

Picture 7: Look for Location action

The Look Material row and the get material row by id will perform the same actions than Look for Location row and get location row by id.

We get the EntityName of the mpl_mm_Materials table calling the dynamics API putting this string in the browser (may be the developer would be to sign in in the powerapp tenant).

https://t8656default.crm4.dynamics.com/api/data/v9.1/EntityDefinitions(LogicalName='cr514_mpl_mm_material')?$select=EntitySetName



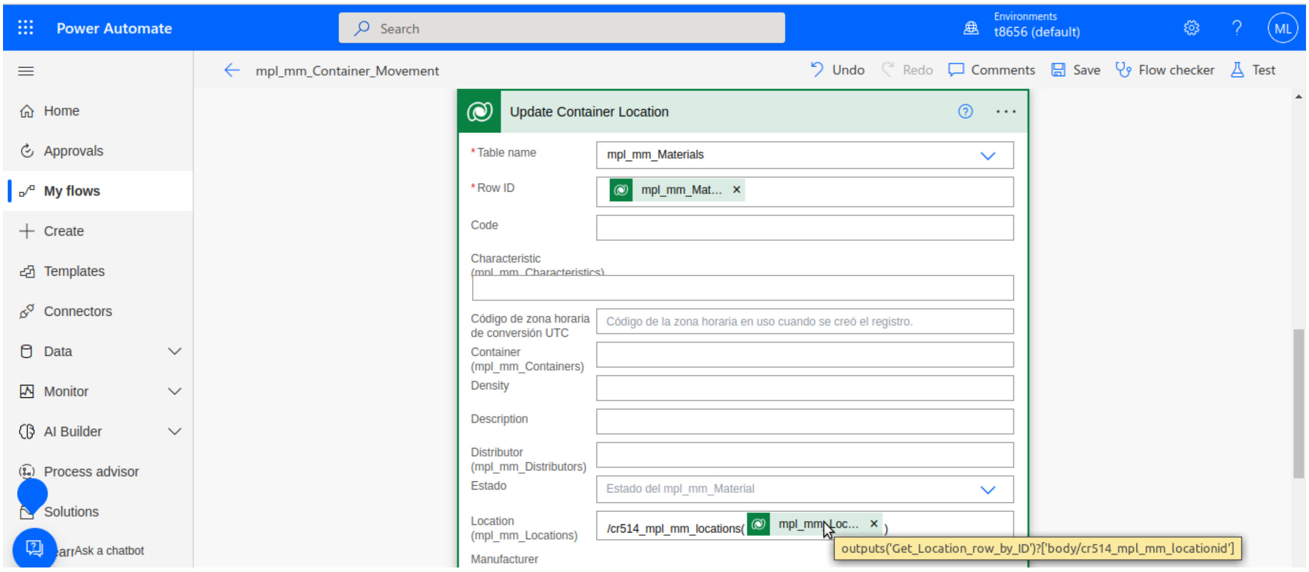
https://t8656default.crm4.dynamics.com/api/data/v9.1/EntityDefinitions(LogicalName='cr514_mpl_mm_material')?$select=EntitySetName - Chromium
https://t8656default.crm4.dyn
Community

https://t8656default.crm4.dynamics.com/api/data/v9.1/EntityDefinitions(LogicalName='cr514_mpl_mm_material')?$select=EntitySetName

```
{"@odata.context":"https://t8656default.crm4.dynamics.com/api/data/v9.1/$metadata#EntityDefinitions(EntitySetName)/$entity","EntitySetName":"cr514_mpl_mm_materials","MetadataId":"19d3d1bd-d816-ed11-b83c-0022487f0dd8"}
```

Picture 8: Dynamics API query call

Finally the update actions will update the location setting in the place /EntitySetName(uniqueidentifierOfTheTable)



Update Container Location

* Table name	mpl_mm_Materials
* Row ID	mpl_mm_Mat...
Code	
Characteristic (mpl_mm_Characteristics)	
Código de zona horaria de conversión UTC	Código de la zona horaria en uso cuando se creó el registro.
Container (mpl_mm_Containers)	
Density	
Description	
Distributor (mpl_mm_Distributors)	
Estado	Estado del mpl_mm_Material
Location (mpl_mm_Locations)	/cr514_mpl_mm_locations(mpl_mm_Loc...)
Manufacturer	

outputs('Get_Location_row_by_ID')?['body/cr514_mpl_mm_locationid']

Picture 9: Update MaterialsAction