Wellington Global Power Platform Bootcamp Instructions

# Scenario

Throughout the day we are going to build an end-to-end application to capture images, analyse them using Power Automate and AI Builder, store the results in Dataverse, run analytics using Power BI and display interactive content on Power Portals based on the captured information.

This scenario is typically used in retail stores to understand the current clientele entering the premise and display relevant information on screens so that customers can better relate to the environment.

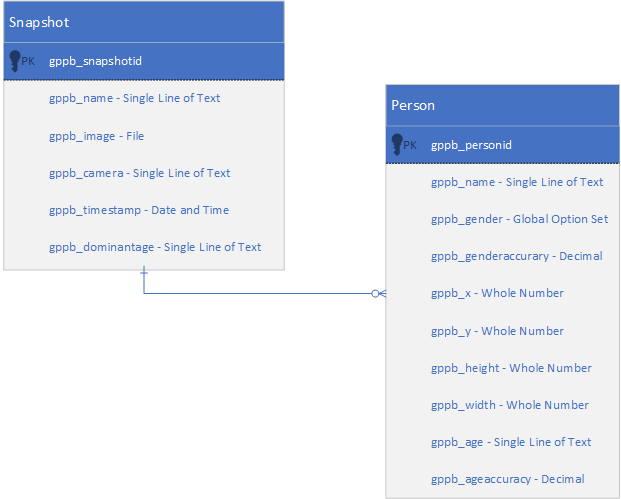
# Getting Started

Set up a Dynamics 365 trial environment using <https://trials.dynamics.com/>

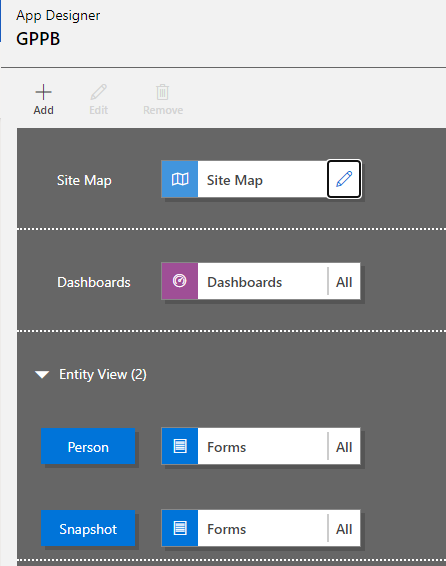
Exercise 1 Building the Data Model

During the first session we will set up the data model that will hold the captured images, the analysis results, as well as the videos and associated metadata to display on the portal.

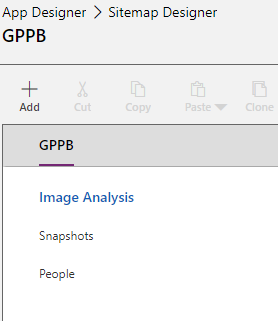
1. In your new instance create a new solution with a publish unique to yourself (e.g. for Geroge Smith prefix it with gs\_)
2. Create the following data model using your newly created prefix



3- Time permitting, you can create a model driven app with the newly created entities.



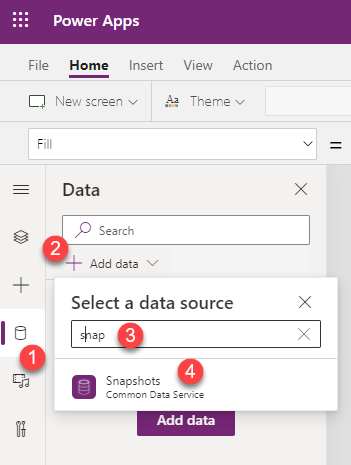
With the following site map

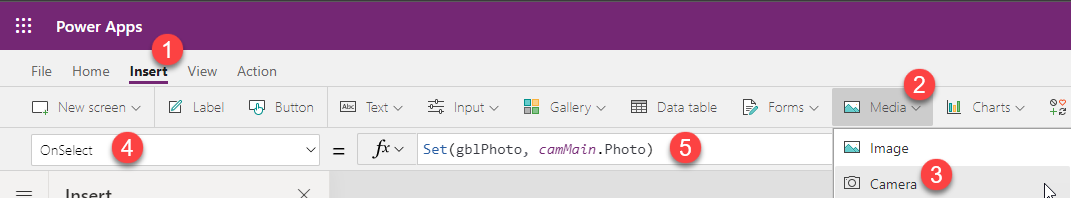


# Exercise 2 Building the Photo Capturing Canvas App

If you have not done the Exercise 1, download the [data model solution .zip file](https://nz21-my.sharepoint.com/:u:/g/personal/linn_nz21_onmicrosoft_com/EZpYOuALMbpFsig-eeOl0g0BfOMVyx3mfsujC_TgTbX2aQ?e=gexPEH) and import into your environment.

Create the canvas app from the solution.

On the Left Navigation Pane > Data > click on + Add data > select Snapshots table as Data Source.  


On the Ribbon > Insert Menu > Media > add the Camera control and set the properties as follows.  


* Name: camMain
* OnSelect: Set(gblPhoto, camMain.Photo)
* StreamRate: 100

Add the Media > Image control and set the properties as follows.

* Name: imgPhoto
* Image: gblPhoto

Add the Input > Text input control and set the properties as follows.

* Name: txtCameraName
* Default: "Webcam"

Add the Input > Timer control and set the properties as follows.

* Name: timShutter
* AutoStart: true
* Duration: 10000
* OnTimerEnd:



# Exercise 3 Building the AI Builder Photo Analyser AI Model

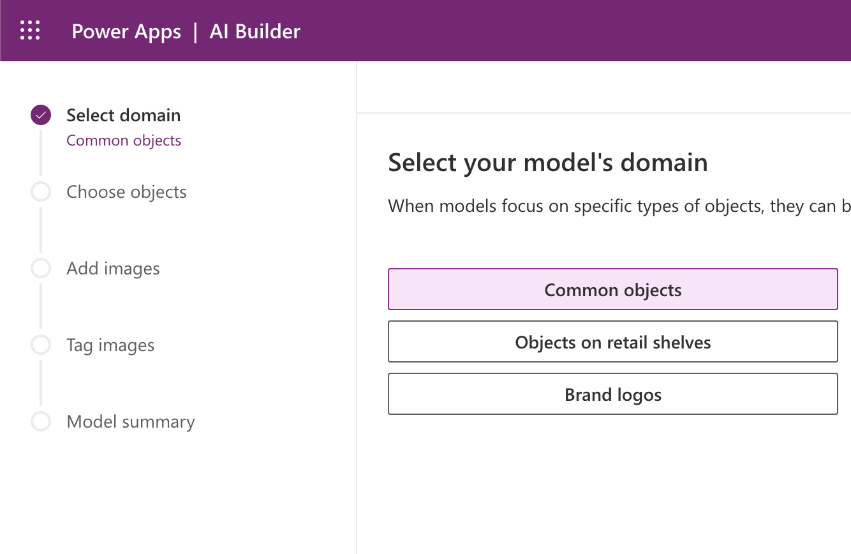
In this exercise, we are going to build 2 AI Models: a model to determine the gender from an uploaded photo of a person, and a model to guess the age range from an uploaded photo of a person.

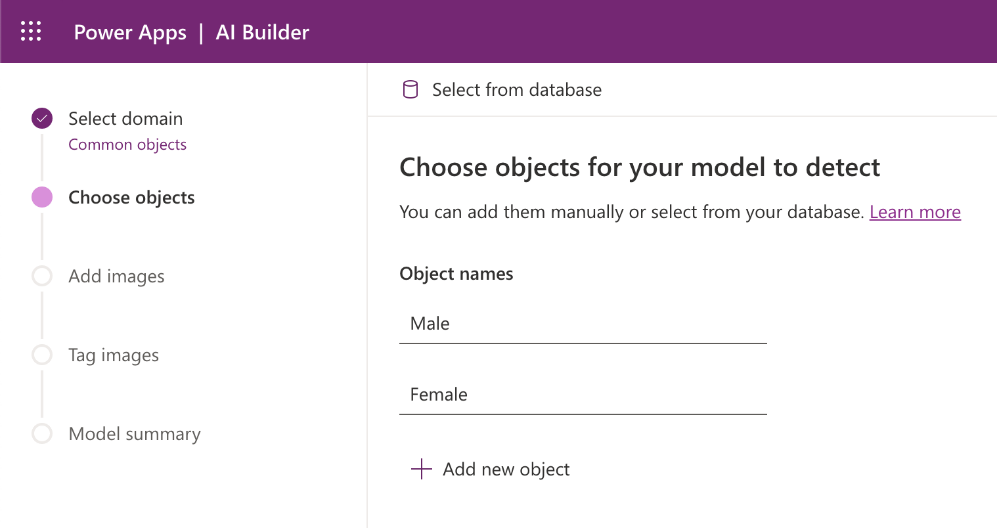
## Build **Gender** model

Navigate to *make.powerapps.com*

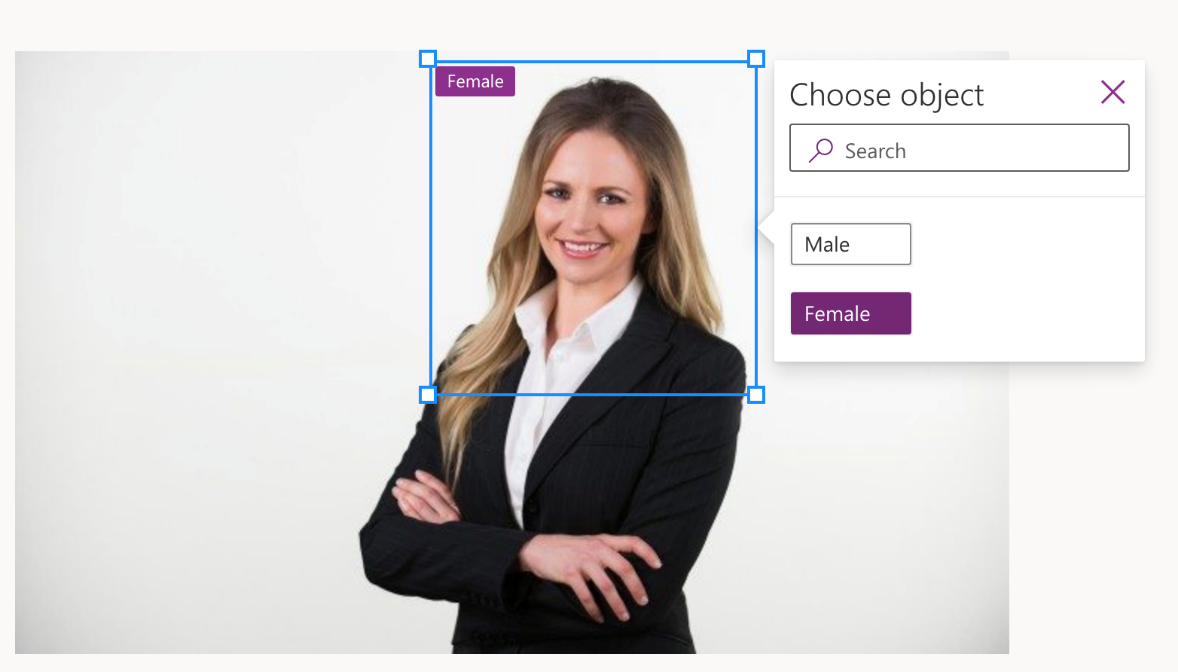
Go to **Build** under **AI Builder**

Choose **Object Detection**.  


Create a new **Gender Model**. In the next screen, choose **Common Object**.  


Add 2 new objects, **Male** and **Female**, in the next screen.  


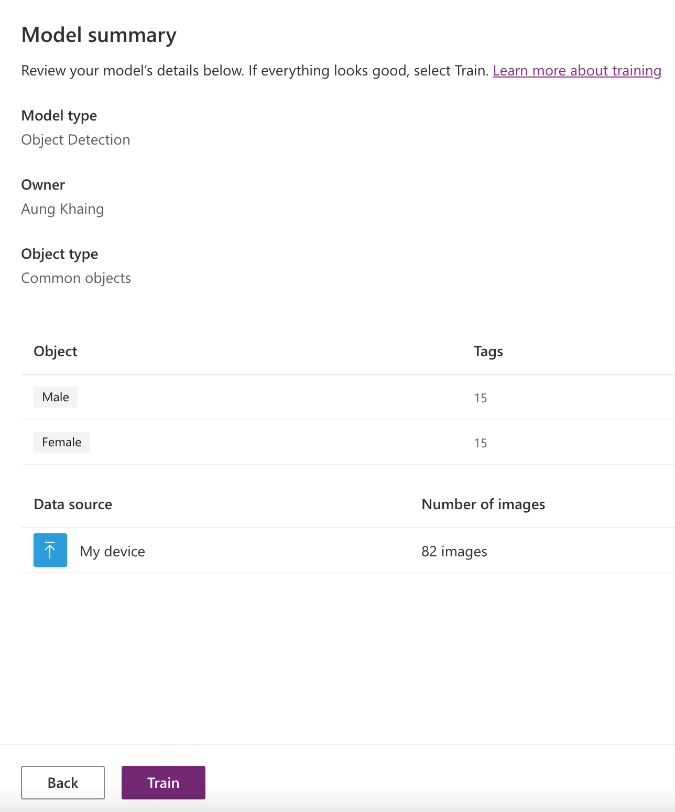
Upload some images to train Gender model in the next screen. Each object needs at least 15 images. We have 2 objects created from the previous step, so will we need at least 15 male images and 15 female images. Sample images can be found at <https://1drv.ms/u/s!AhsOOghZC8Ctipxs5Ux0pQ6FY5plPg?e=ggdLzQ>

Next step is **Tagging**. We are going to teach the AI model the difference between male or female images. The AI model will learn in its own way to differentiate between genders. To do that, select the facial area and tag the object correctly.  


Repeat the steps for at least 15 images per object. (Don’t grab a coffee yet!)

After tagging a total of 30 images, your hard work will pay off soon.

Now, you can train your AI model. COFFEE time! It will take a few minutes to train.



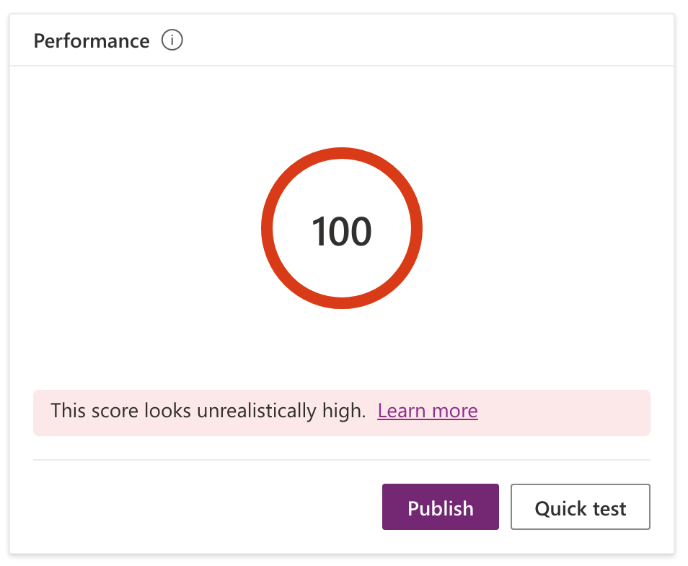
## Build **Age** model

Repeat the previous process. But this time, you need 3 objects; infant, adult and young-adult.

Ouch! 45 images (15 images x 3 objects).

## Test AI model

After another coffee and a chat, you will see the model is trained.  


Open the model and do a **Quick test**. If you see 100% performance like this, you better add more images to train the model.  


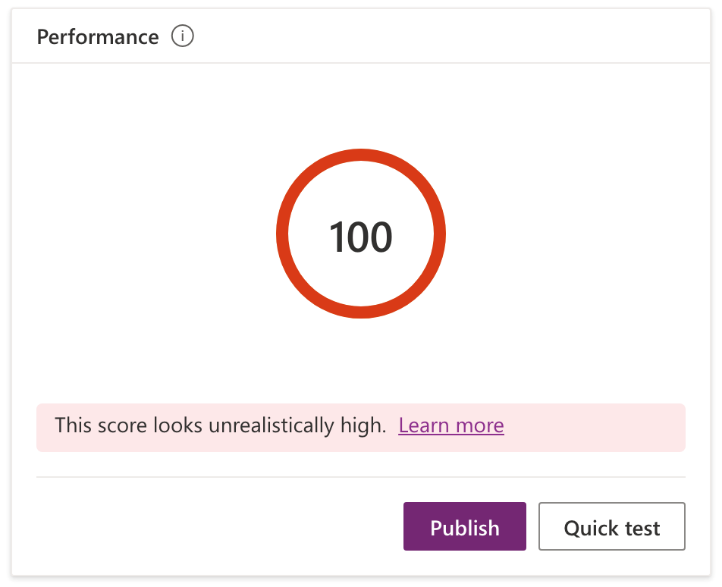
You can upload a person photo or your selfie and check your gender.

Sometimes, you may get No objects found, which means the model needs further training.  


But of the time you’ll get the correct result.  


## Publish the model

Once you are happy with the result or you are so done with AI Builder, remember to publish the model so you can use the models in the next step.



## Add to the solution

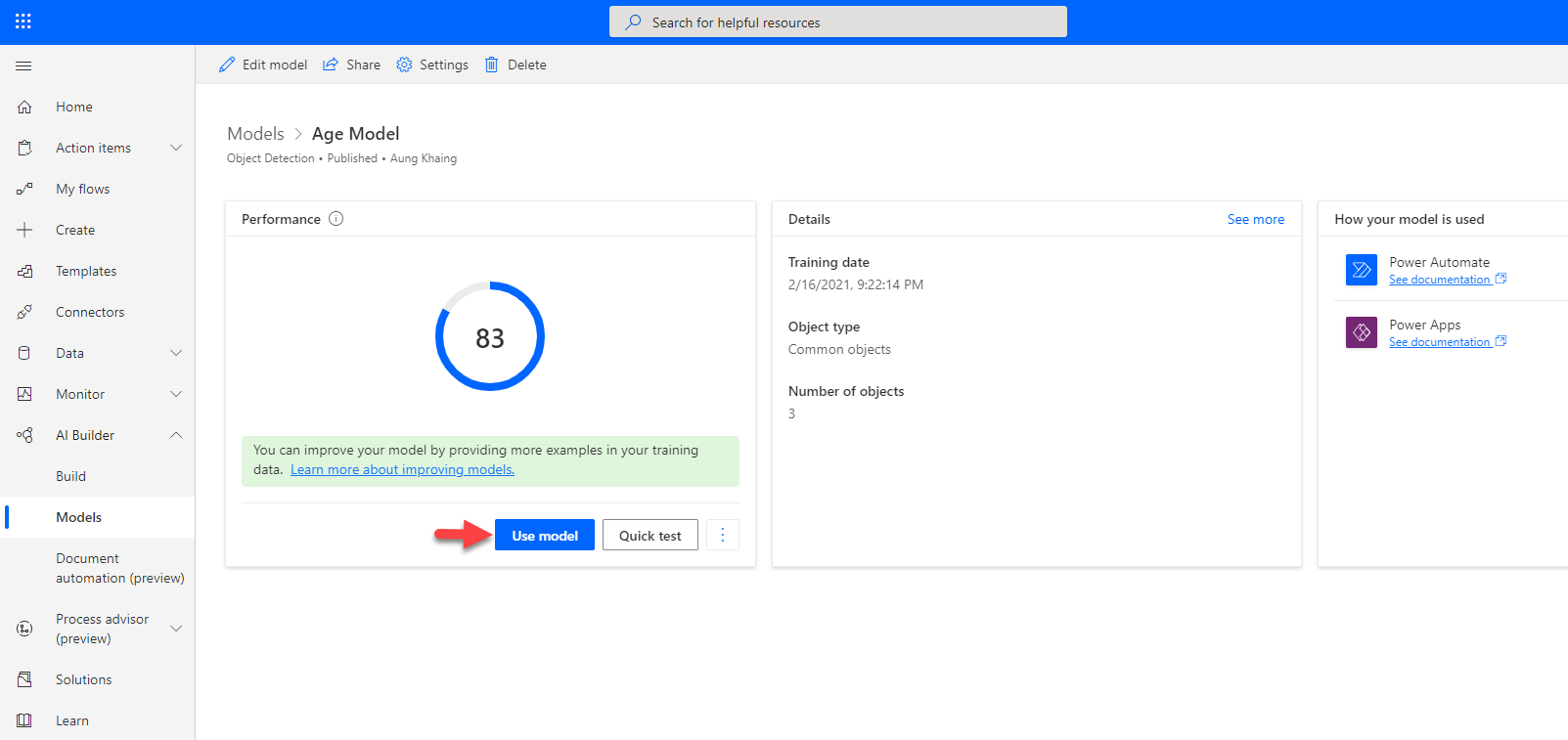
Add 2 AI models into the solution. So you can easily transfer AI model from environments to environments.

## For adventurous

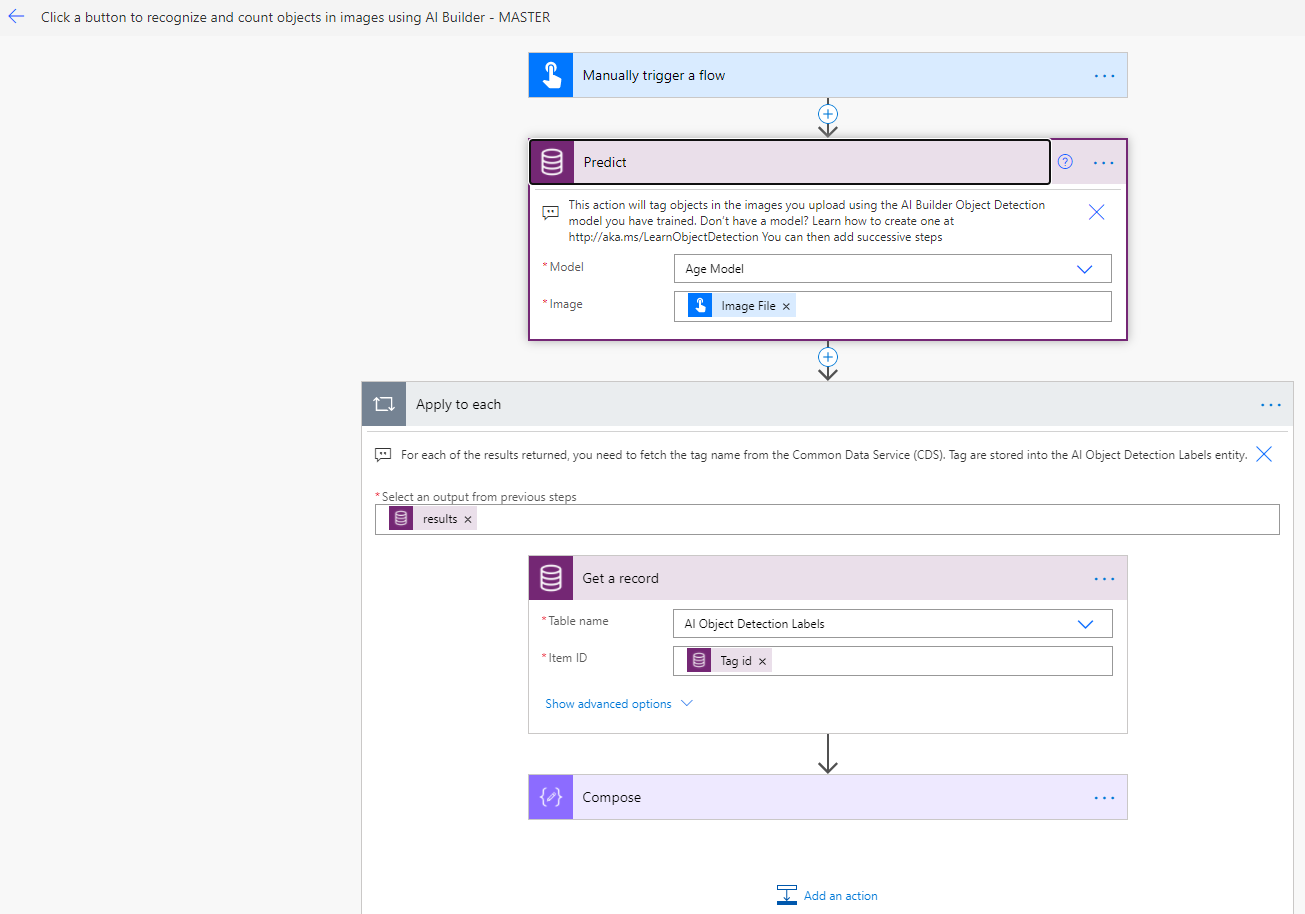
In above instructions, we create 2 separate models. In AI Builder, you can use only 1 model to tag different objects (age and gender) and you can multi tag in each image.

# Exercise 4 Building the Cloud Flow that Calls the AI Model and Stores the Results

In AI Builder, from the model select Use Model.

  
  
Power Automate maker site will load in a new browser tab where a template is displayed and a prompt will appear to use your credentials for the connectors.

A new cloud flow will next appear with the template applied. The template can now be configured.



There will be some actions that will need to be removed or updated.

## Simple Exercise - Predict Age AI Model

For the simple exercise, let’s retrieve the *Age* predicted from the *Predict Age AI Model*.

The response from the AI Model will be the following

    "@odata.context": "https://nz21.crm6.dynamics.com/api/data/v9.1/$metadata#Microsoft.Dynamics.CRM.PredictResponse",

    "responsev2": {

      "@odata.type": "#Microsoft.Dynamics.CRM.expando",

      "operationStatus": "Success",

      "predictionOutput": {

        "@odata.type": "#Microsoft.Dynamics.CRM.expando",

        "results@odata.type": "#Collection(Microsoft.Dynamics.CRM.crmbaseentity)",

        "results": [

          {

            "@odata.type": "#Microsoft.Dynamics.CRM.expando",

            "tagId": "5f037741-3753-4cdd-b160-48fa221ee1d5",

            "tagName": "Adult",

            "confidence": 0.530878,

            "boundingBox": {

              "@odata.type": "#Microsoft.Dynamics.CRM.expando",

              "left": 0.01884067,

              "top": 0.028233528,

              "width": 0.78930706,

              "height": 0.74064994,

              "polygon": {

                "@odata.type": "#Microsoft.Dynamics.CRM.expando",

                "coordinates@odata.type": "#Collection(Microsoft.Dynamics.CRM.crmbaseentity)",

                "coordinates": [

                  {

                    "@odata.type": "#Microsoft.Dynamics.CRM.expando",

                    "x": 0.01884067,

                    "y": 0.028233528

                  },

                  {

                    "@odata.type": "#Microsoft.Dynamics.CRM.expando",

                    "x": 0.80814773,

                    "y": 0.028233528

                  },

                  {

                    "@odata.type": "#Microsoft.Dynamics.CRM.expando",

                    "x": 0.80814773,

                    "y": 0.768883468

                  },

                  {

                    "@odata.type": "#Microsoft.Dynamics.CRM.expando",

                    "x": 0.01884067,

                    "y": 0.768883468

                  }

                ]

              }

            }

          }

        ]

      }

    }

  }

### What is the trigger?

Currently it will show as a Manual trigger, where a button is selected.  
The trigger needs to be update where it will be when a Snapshot row is created from the canvas app.

### What else do I need to do?

The image that would have been uploaded to the app needs to be retrieved.  
Hint: There’s a CDS CE action that is dedicated to retrieving an Image file.

The Person row that is created needs to reference the Tag name and Tag confidence value from the Predict Age AI Model. The Age and Age Confidence columns in Microsoft Dataverse is single line of text and is therefore straightforward to do.

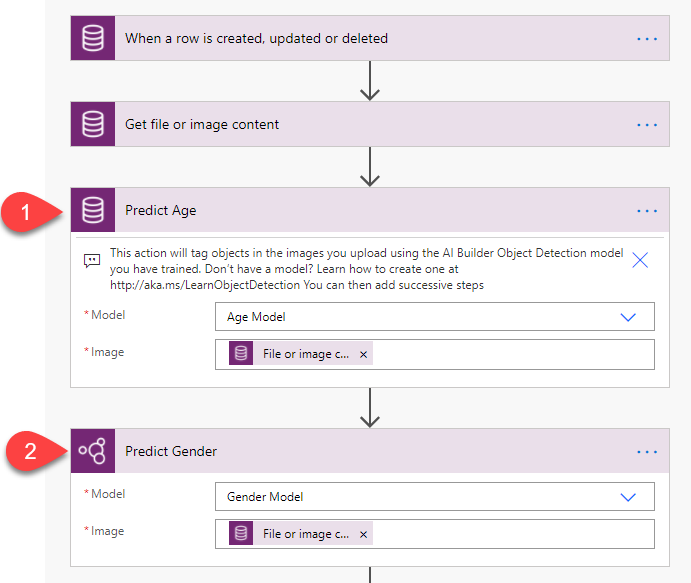
## Advanced Exercise - Predict Age AI Model and Predict Gender AI Model

There are two methods applied which is advanced learning

1. How to retrieve result from both the Predict Age AI Model and Predict Gender AI Model
2. How to populate a Choices column for a table in Microsoft Dataverse

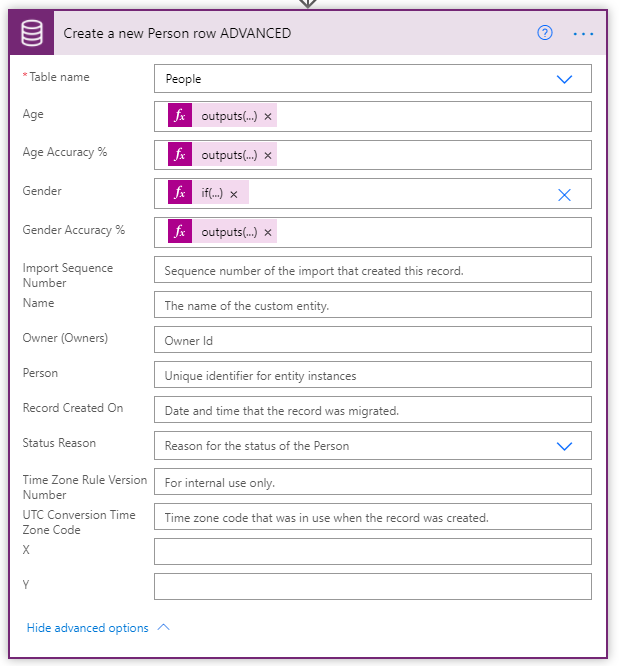
#### How to retrieve result from both the Predict Age AI Model and Predict Gender AI Model

In this scenario both the AI Models will be used in the cloud flow. This means there should be two actions like the following in the cloud flow.



The assumption here is only a single result is returned for each AI models.

This means only the first row in the array needs to be referenced through an expression in the action that creates a new Person row.



One way of achieving this is through the following expression

outputs('Predict\_Age')?['body/responsev2/predictionOutput/results'][0]?['tagName']

The same pattern needs to be applied to both AI Models.

* Age tagName
* Age confidence
* Gender tagName – refer to the following section as this expression requires an additional function
* Gender confidence

#### How to populate a Choices column for a table in Microsoft Dataverse

Gender is a global choice where there values are defined for the end user to choose from.  
In the Gender AI Model the tags defined are Male and Female where each of these are what we call Display Name Values but behind the scenes at the database level they have an associated integer value.

* Male
  + 141870000
* Female
  + 141870001

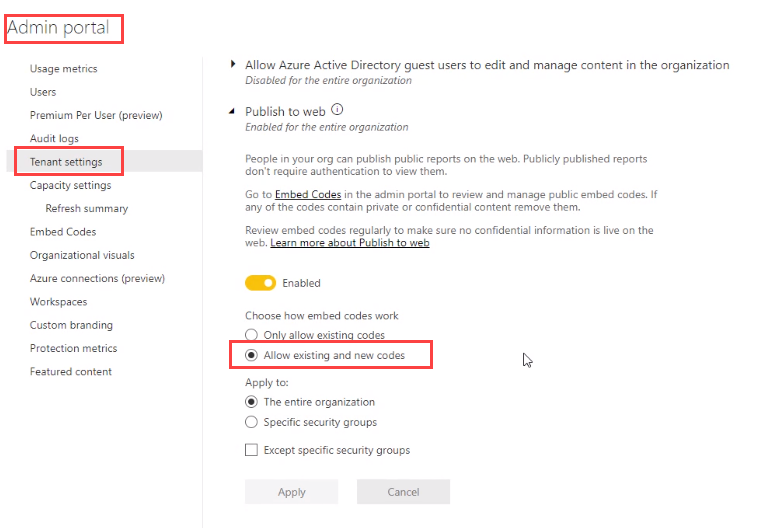
This means based on the tag name identified by the Gender AI Model, some logic needs to be applied to check that if the tag name is identified, use the corresponding integer value of the choices field in the create a new Person row action.

For example if the Gender AI Model tagName equals Male, set the Gender column to be 141870000

This requires the following [If function](https://docs.microsoft.com/en-us/azure/logic-apps/workflow-definition-language-functions-reference#if) to be used.

# Exercise 5 Building the Power BI Dashboard

If you are using your own environment ensure you have enabled ”Publish to web” and that you are ”Allowing existing and new codes” in the Power BI administrator side.



# Exercise 6 Building the Power Portal that Displays the Ads

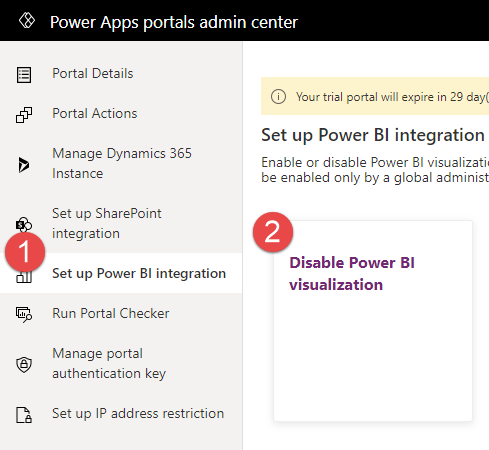
In this last exercise we will be setting up a Power Portal to display age tailored content based on the previous image analysis results.

## Portal Setup

### Provision

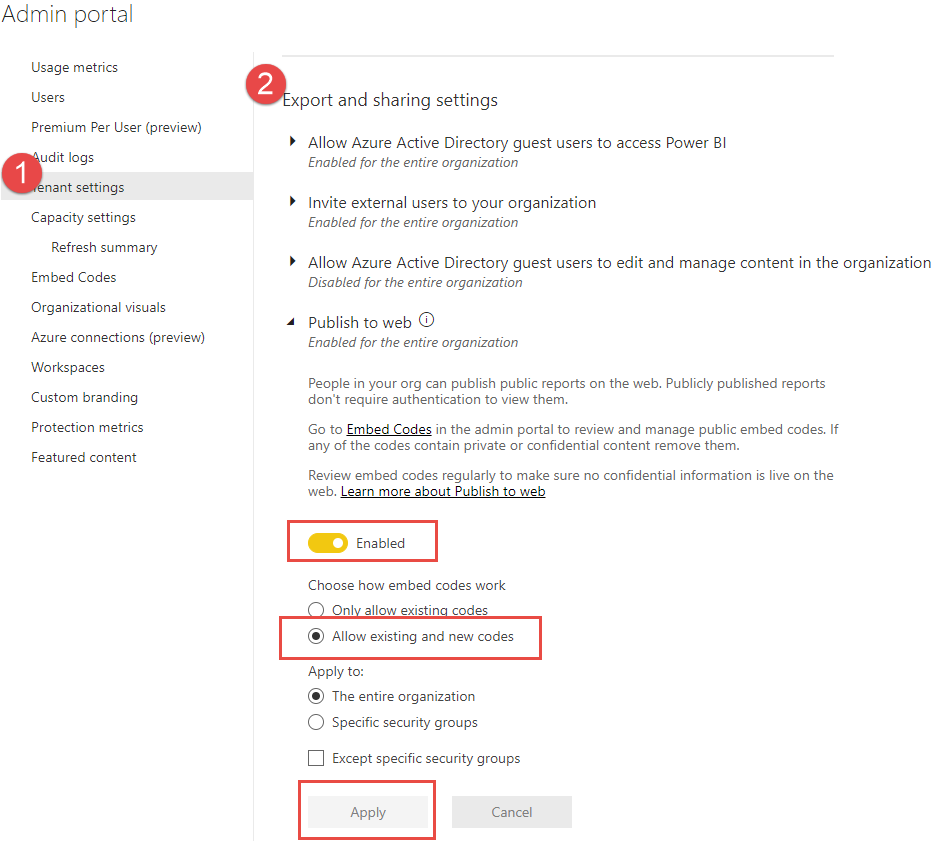
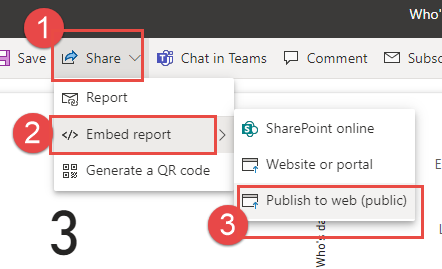
1. <make.powerapps.com> 🡪 Apps 🡪 New App 🡪 Portal
2. Pick name and URL. Will take around 1h to complete

### Enable Power BI

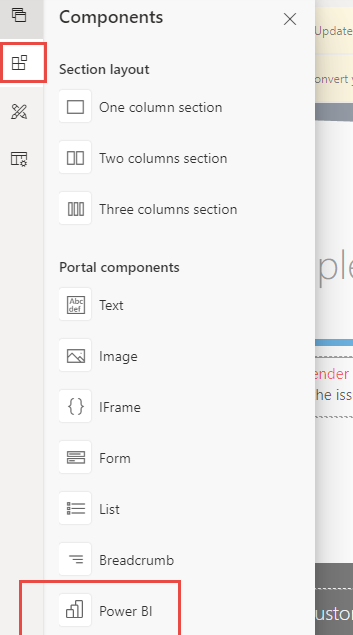
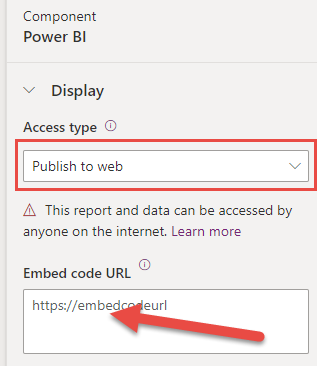
1. <make.powerapps.com> 🡪 Select portal 🡪 Settings 🡪 Administration
2. In portal admin 🡪 Set up Power BI integration 🡪 Enable Power BI Visualization  
   

## Configure Power BI Sharing

### Configure Power BI to allow to share on Portals

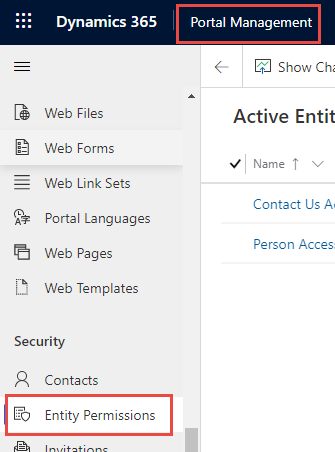
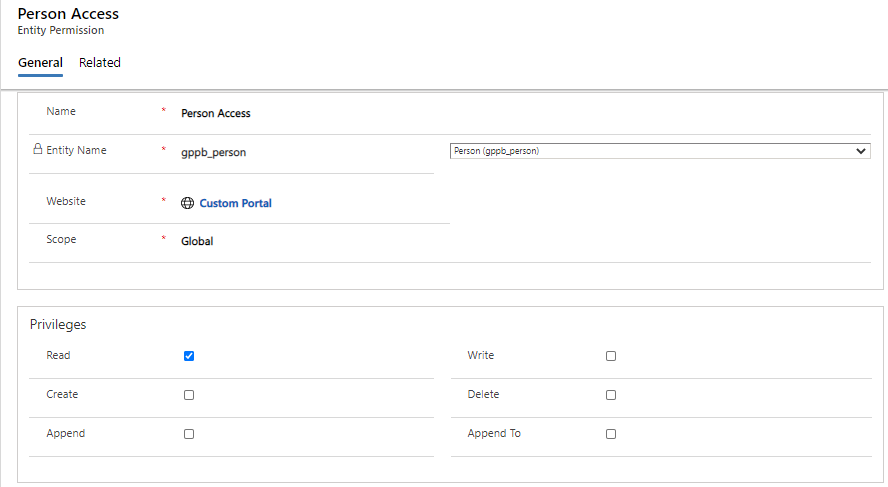
1. As a Power BI Admin, go to <https://app.powerbi.com/admin-portal>
2. Select Tenant settings 🡪 Export and sharing settings 🡪 expand Publish to the web  
   
3. Go to your report in <https://app.powerbi.com/> and share:  
   
4. Copy and save the report URL for later!

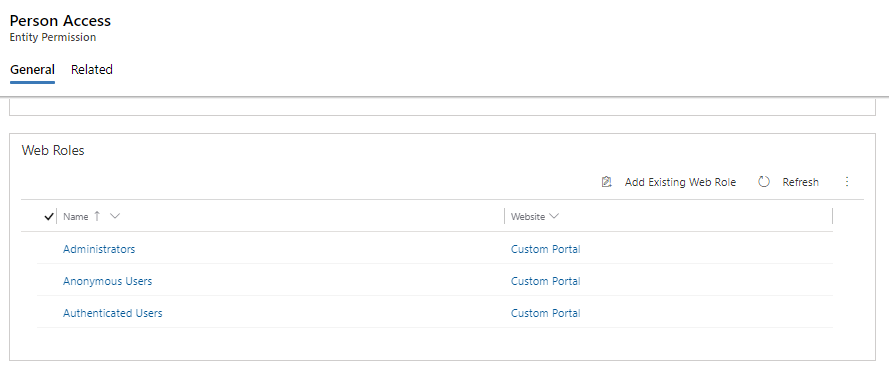
## Add report to web page

1. [make.powerapps.com](file:///C:\Users\eregnier\Desktop\make.powerapps.com) 🡪 Select portal 🡪 Edit
2. In Portal Studio 🡪 select HomePage (or page of your preference) 🡪 select desired section 🡪 Add Power BI component to desired area  
   
3. Select “Publish to web” as access type and paste the report URL  
   

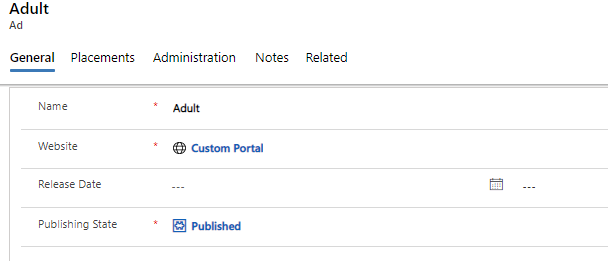
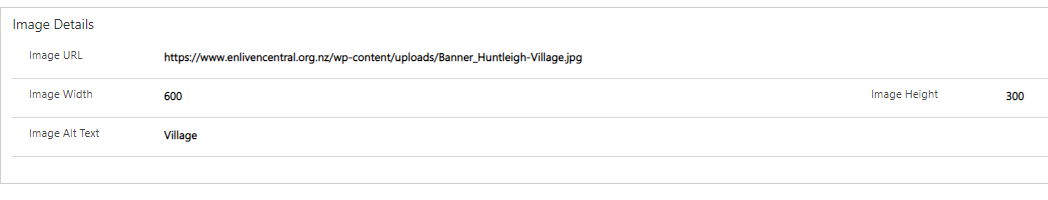
## Configure Ads in Dataverse

### Setup permissions to display ads

1. Pick the Portal Management app 🡪 Entity Permissions  
   
2. Add a new permission for Person entity, give it any name and give **read** access to **all web roles** to **Person** entity/table:  
   



### Setup new Ads

1. In Portal Management app 🡪 Ads
2. Add new Ads, ensure the following:
   1. Name equals the Age Group from Person. This will be the convention to determine which ad to display
   2. Publish state is “Published”  
      
   3. Configure Image details:  
      

## Configure Ads on web page

### Create a Content Snippet for the ad

First step is to determine which ad to display by querying Dataverse data with the help of liquid and snippets

1. In Portal Management app 🡪 Content Snippets
2. Add new snippet
   1. give it a **Name**
   2. set the **Type** to “Text”
   3. tag to the **Website** “Custom Portal”
3. Add the liquid tag for the FetchXML type of query in the **Value** field  
   Tip! you can use XrmToolBox FetchXmlBuilder to build queries: <https://www.xrmtoolbox.com/plugins/Cinteros.Xrm.FetchXmlBuilder/>

{% fetchxml topAd %}

<fetch top="1" aggregate="true">

<entity name="gppb\_person">

<attribute name="gppb\_agetest" aggregate="count" alias="AgeCount"></attribute>

<attribute name="gppb\_agetest" groupby="true" alias="AgeGroup"></attribute>

<order alias="AgeCount" descending="true"></order>

</entity>

</fetch>

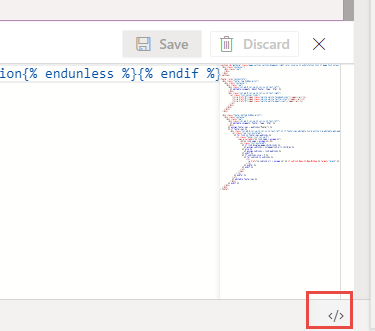
{% endfetchxml %}  
  
{% assign name = topAd.results.entities[0].AgeGroup %}

1. Then one we know which add to add, need to fetch and display it.  
   Add more liquid tags below to then display the ad based on the query:  
     
   {% assign ad = ads[name] %}

<h4>{{ ad.title }}</h4>

<img src={{ ad.image.url }} alt={{ ad.image.alternate\_text }} height={{ ad.image.height }} width={{ ad.image.width }} />

### Add the snippet to the web page

1. In Portal Studio 🡪 Select page and section to place the ads 🡪 select source code editor  
   
2. Paste the following liquid tag to add the snippet. Update the name to your snippet name.  
     
   {% include 'snippet' snippet\_name:'GPPB Ad' %}

More details on liquid and Ad:   
<https://docs.microsoft.com/powerapps/maker/portals/liquid/liquid-overview>   
<https://docs.microsoft.com/powerapps/maker/portals/liquid/liquid-objects>