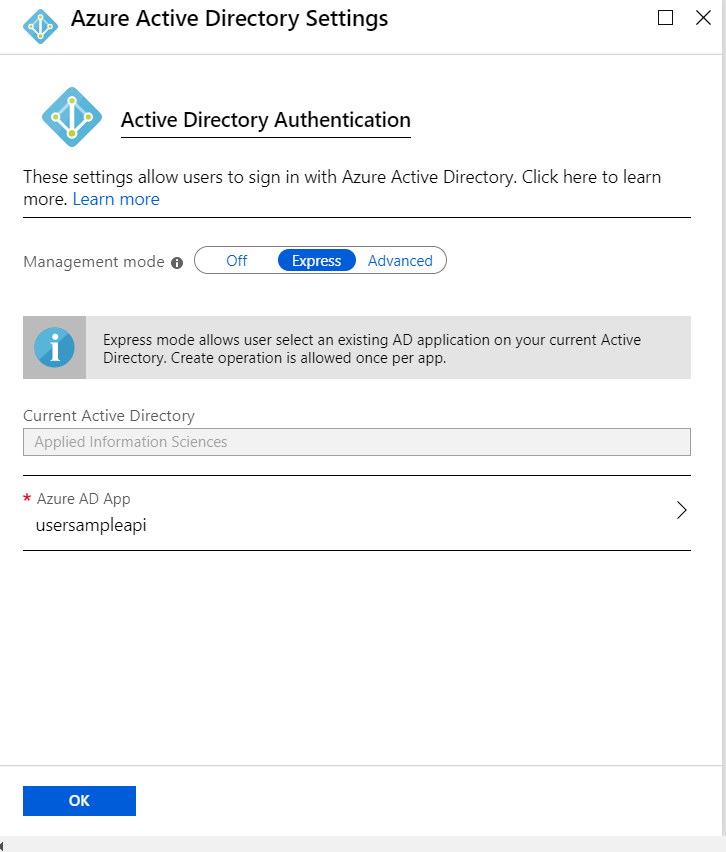
**Steps once the app is published:**1) To enable AAD authentication on the app (Backend Registration)

1. In the “Authentication / Authorization” section, select following settings:  
   Select Login with Azure Active Directory from the dropdown for Action to take when request is not authenticated.

Now Azure Active Directory from the Authenticator Provides list and select the Express mode and create a New Azure App Registration.  
Once the creation is complete, this will look like:  


Save the details. Go to Azure Active Directory and keep a note of the Client ID.

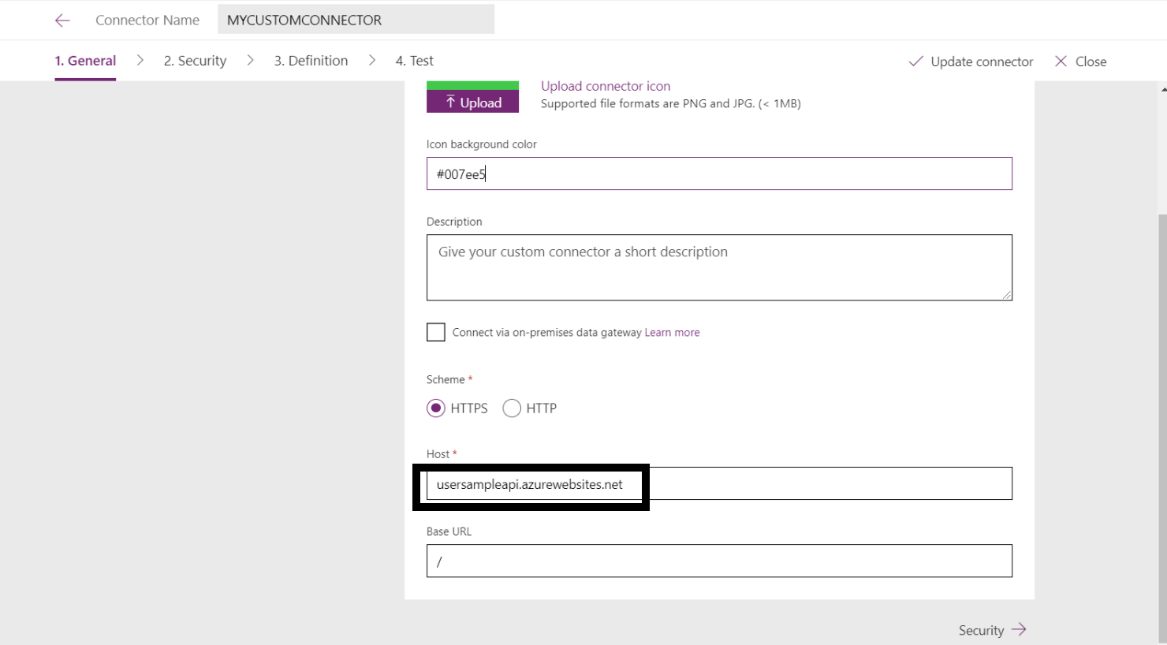
2) Now execute the PowerShell script and pass on the following parameters:

a) APIAppRegistrationName -> Name of API App Registration created in Step1.

b) PowerAppRegistrationName -> Name of App Registration for PowerApps.

c) password -> Secret Value for PowerApps App Registration.

Navigate to Azure Active Directory -> App Registrations  
Please take a note of the Client ID of PowerApps App Registration.  
  
This completes the App registration part.

3) Now the Custom Connection, Navigate to PowerApps portal and sign in. Select Data from the left menu and then go to custom connector, create a new custom connector from blank.  
  
a) Configure General settings as:  


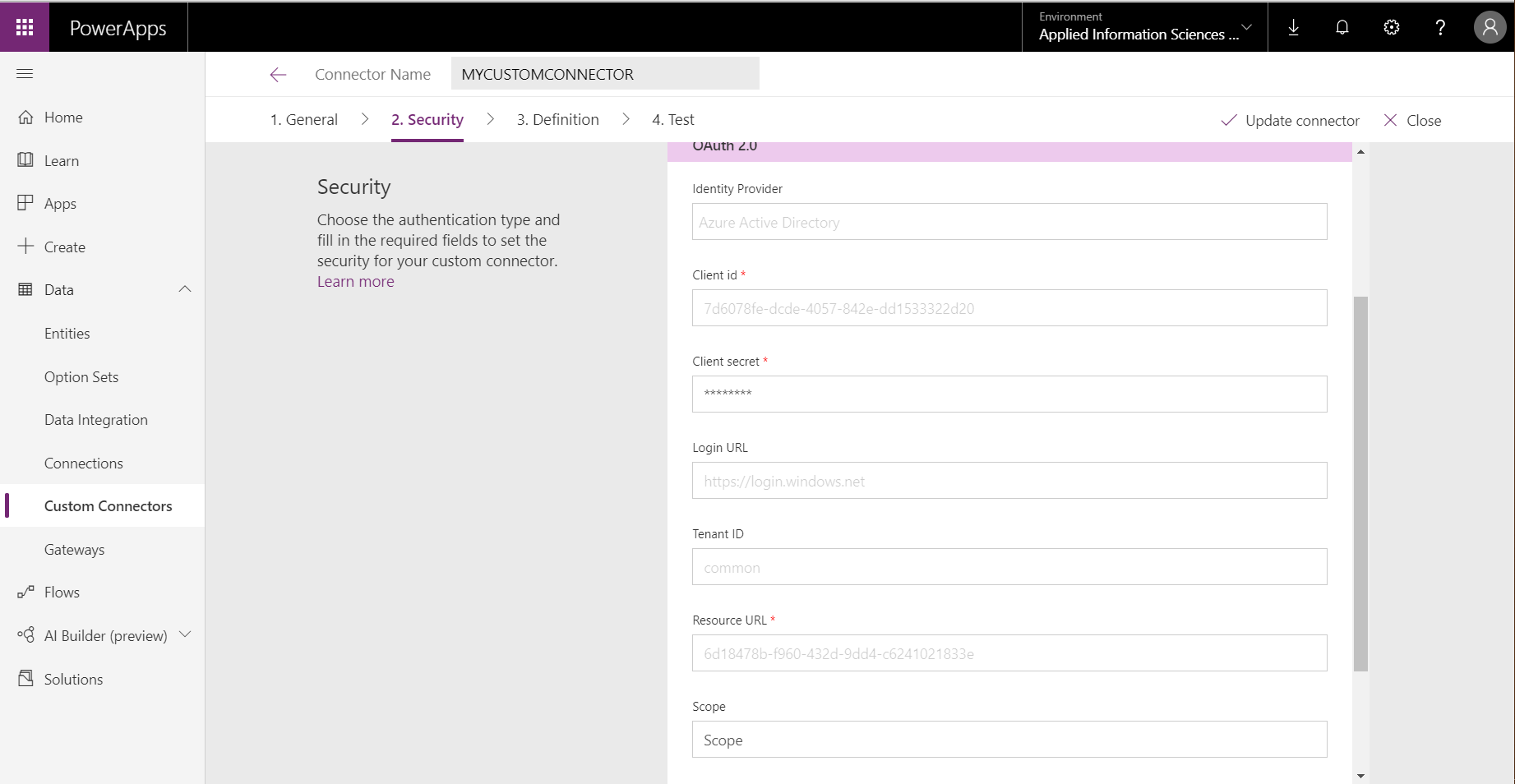
Now, add the API APP URL as the Host in General Settings.

b) Configure Security as:  
Authentication Type: OAuth 2.0  
Authentication Provider: Azure Active Directory

Client Id: Client ID noted in Step2.

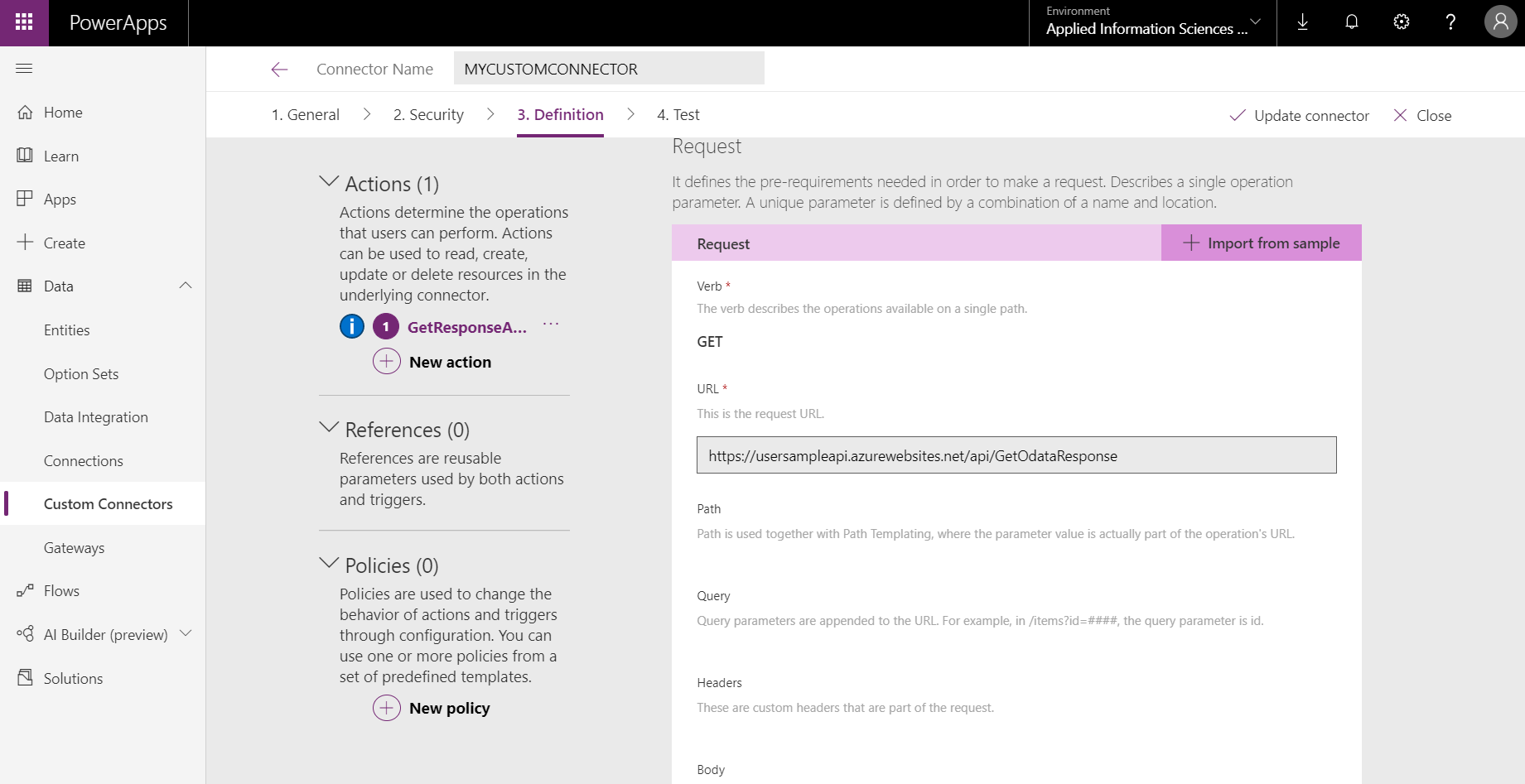
Client Secret: Secret passed in the PowerShell Script

ResourceUrl: Client ID for the app generated in Step 1.



c) Configure definition as:  
Operation Id: Name of action

Request: Import from Sample ->  
Method: Get

Url: https://usersampleapi.azurewebsites.net/api/GetOdataResponse  
  


Response: Default Response -> Import from Sample:  
{

"@odata.context": "https://mockresponseapi.azurewebsites.net/$metadata#MockApi",

"value": [

{

"FirstName": "John",

"LastName": "Smith",

"Gender": "Male"

},

{

"FirstName": "Steve",

"LastName": "Smith",

"Gender": "Male"

}

]

}

d) Create a connection in the test tab.   
  
4) Test out the Custom Connector.

5) Canvas App

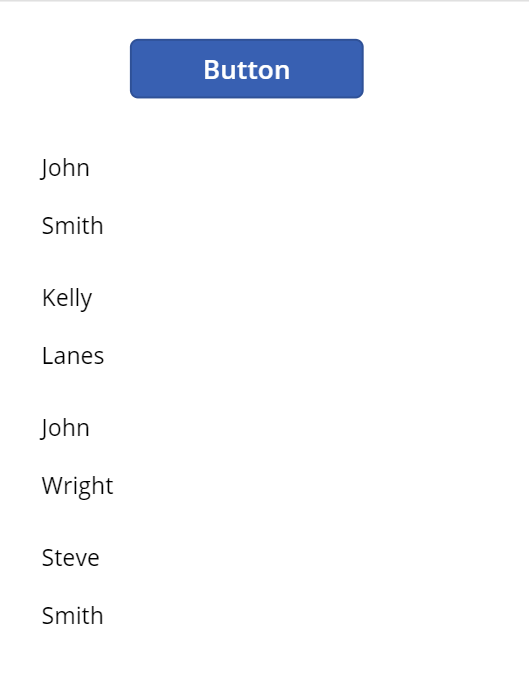
Create a New Canvas App.

Now, add data source, View -> Data Sources -> Add New Connection -> Select the created Custom Connector.

Insert Button on the screen:

Button -> OnSelect -> ClearCollect(reponseCollection, SAMPLECUSTOMCONNECTOR.getMockResponse())  
responseCollection: Name of local collection

SAMPLECUSTOMCONNECTOR: Connection name generated by PowerApps

  
This will create a local collection with the values returned by API call and this collection can be used for data retrieval.