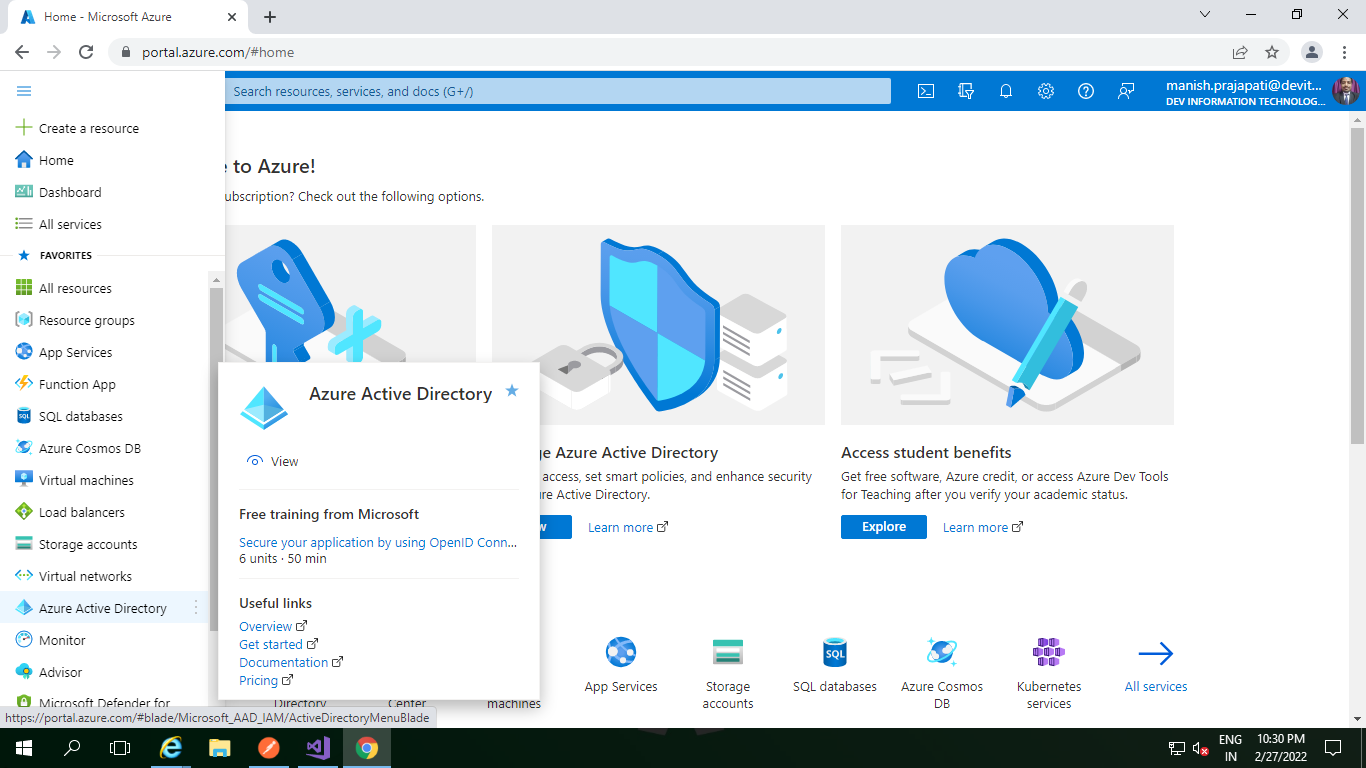
**[Step-by-step] Connect Dynamics 365 Online from Postman using WebAPI Auth 2.0 Authentication**

## Step:1 – Register an application in Azure Active Directory

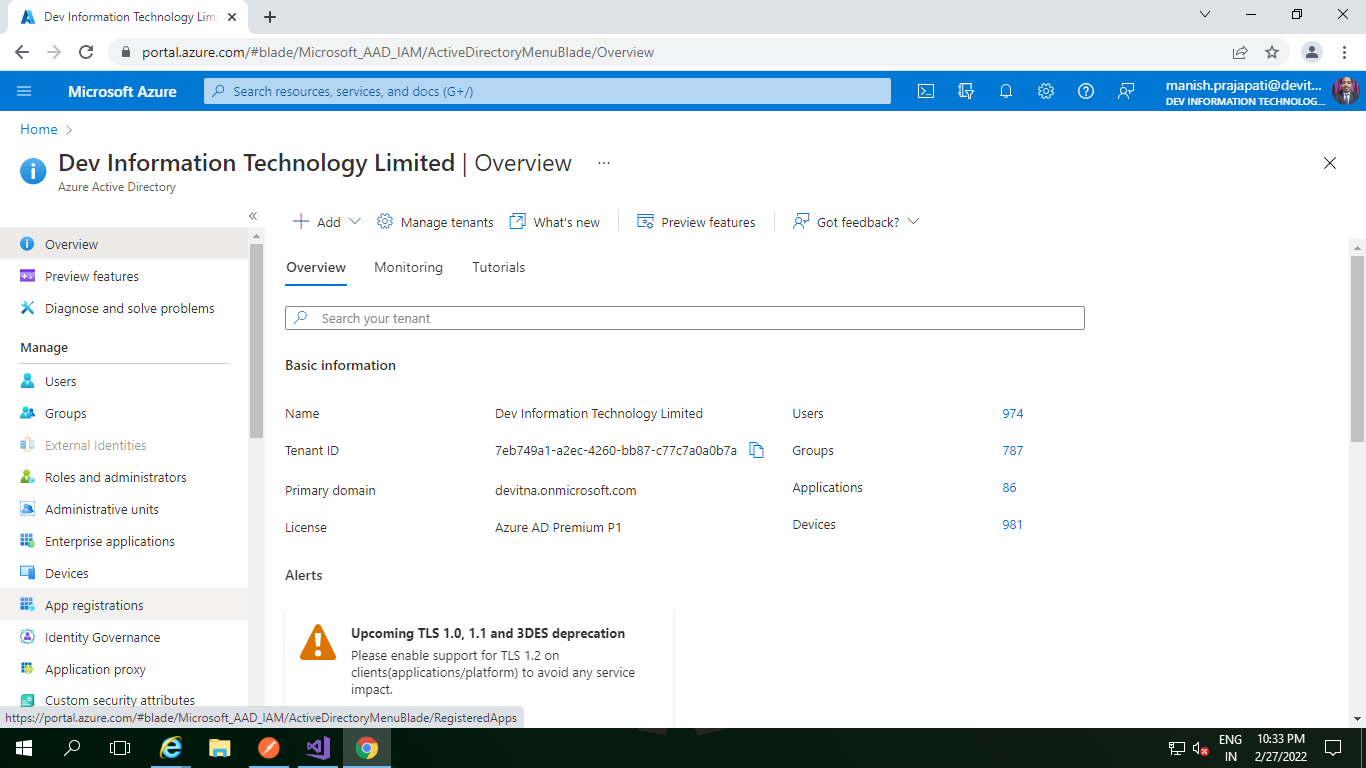
The first thing is to register an application using Azure Active Directory. Open Azure Active

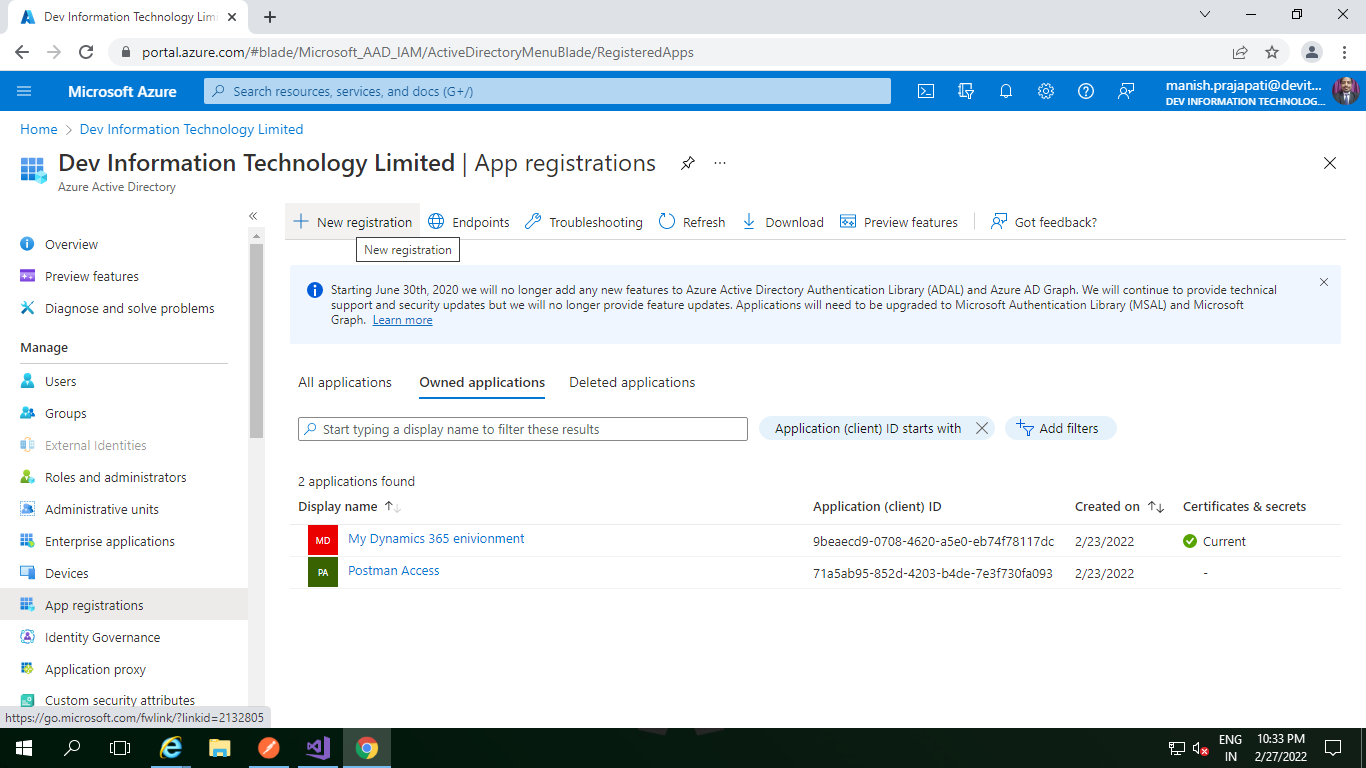
Directory by clicking the link <https://aad.portal.azure.com/> Click on the Azure Active

Directory link from left panel and Navigate to App Registration option.



Then select app registration



c

Now Create new app registration

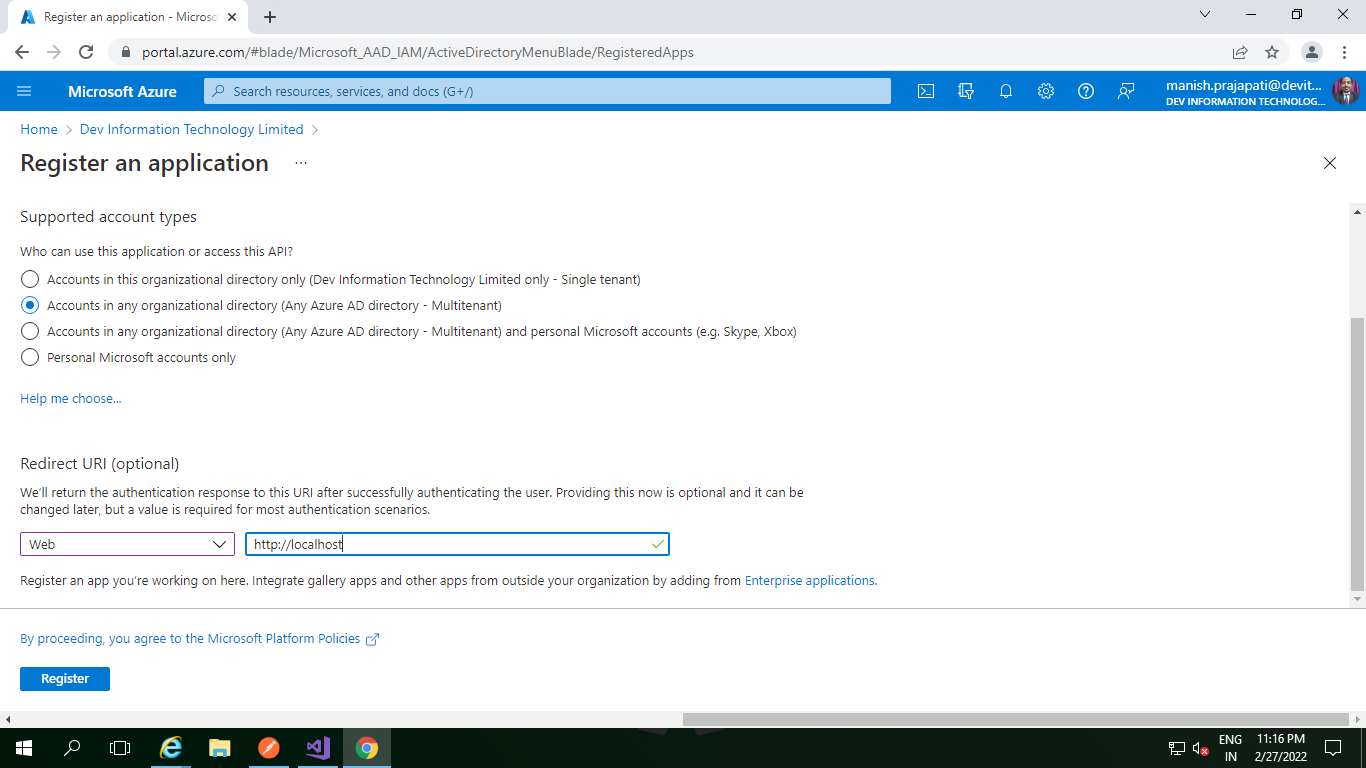
Click on **+New registration** button to register a new app. In the Register an application screen provide the below options.

Name : **Postman Access** (Note : you can give a name as per your choice)

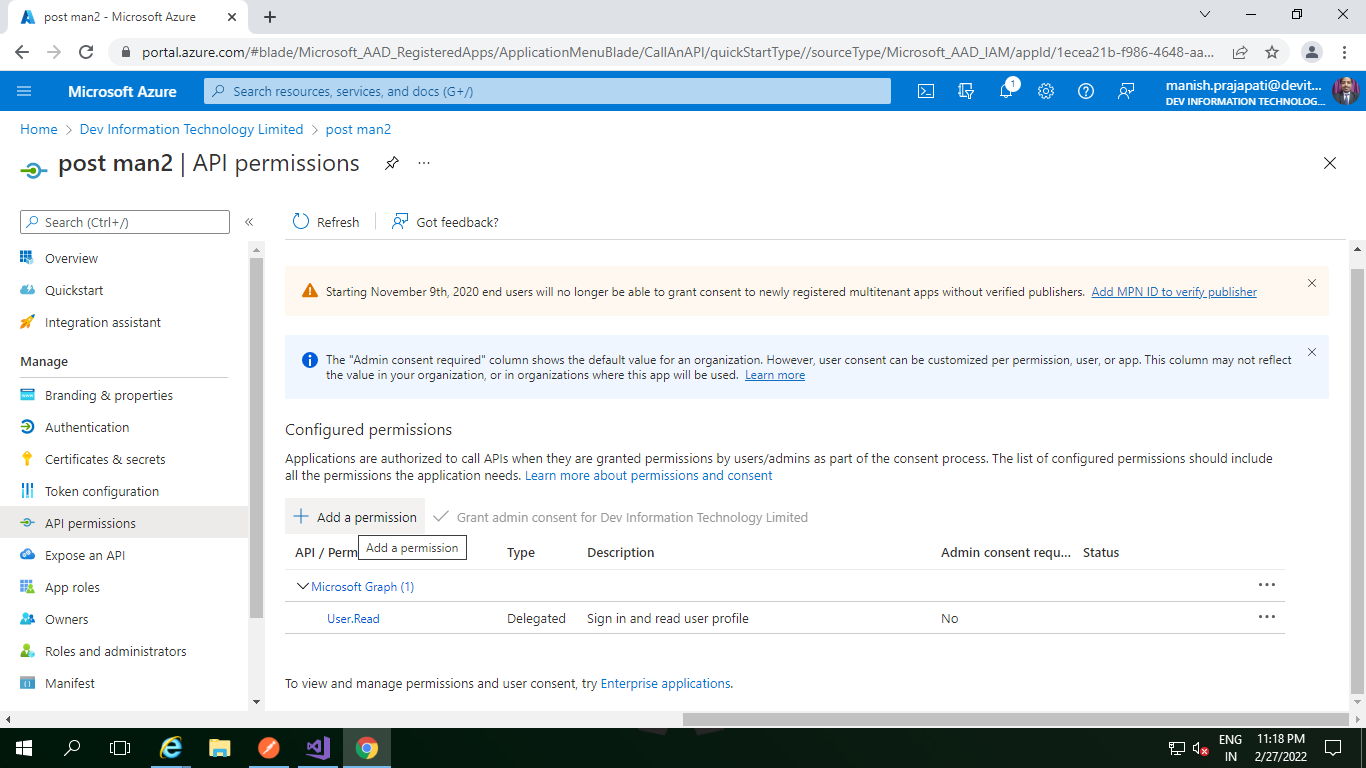
Supported account types : **Choose Accounts in any organizational directory** (Any Azure AD directory – Multitenant)

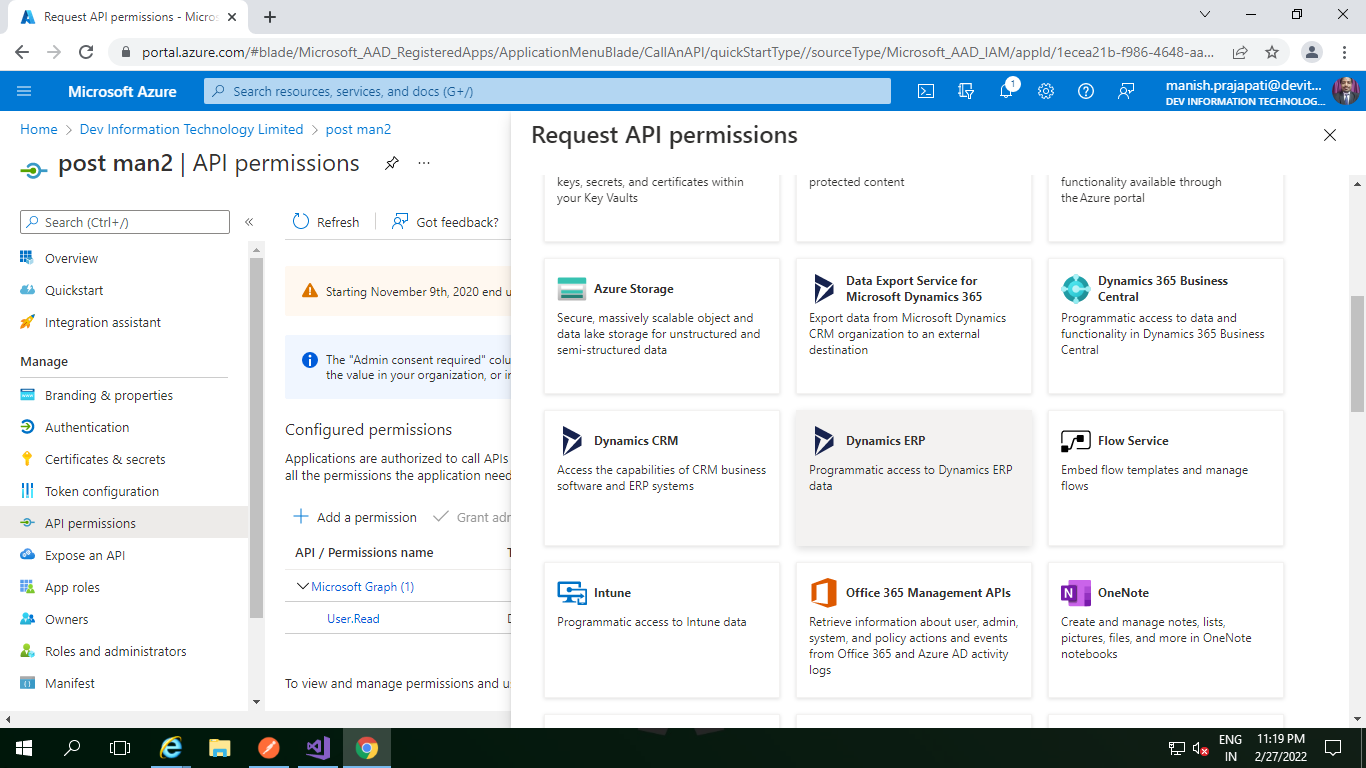
Redirect URI (optional) **: http://localhost** (Note : you can give a name as per your choice)

Now your registration form will look like as below. After that click on Register.

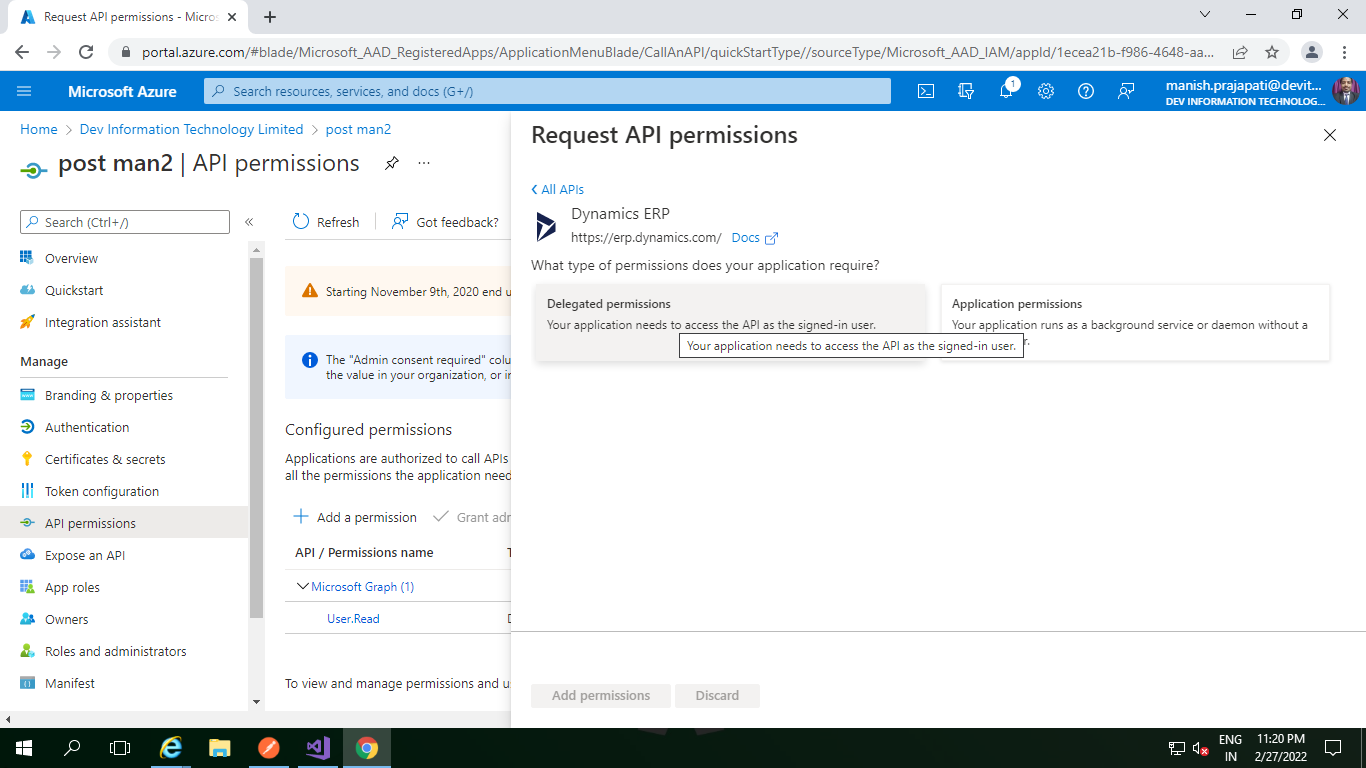


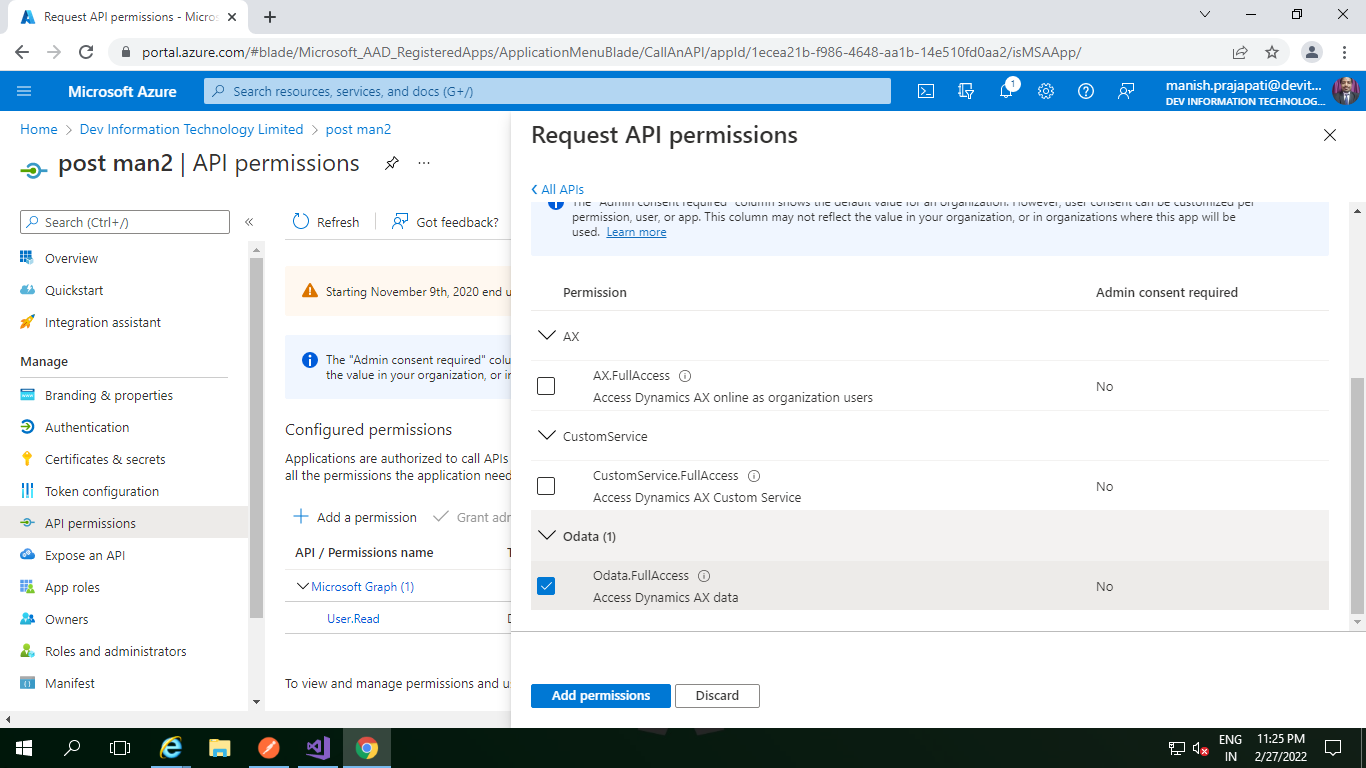
Once you save the application will register successfully. Next we have to add **API Permissio**n for the same app we have just created. Being on the same application page click on **API Permissions** option and choose **+ add a permission**. The **Request API Permission** window will appear where you need to select **Dynamics D365**



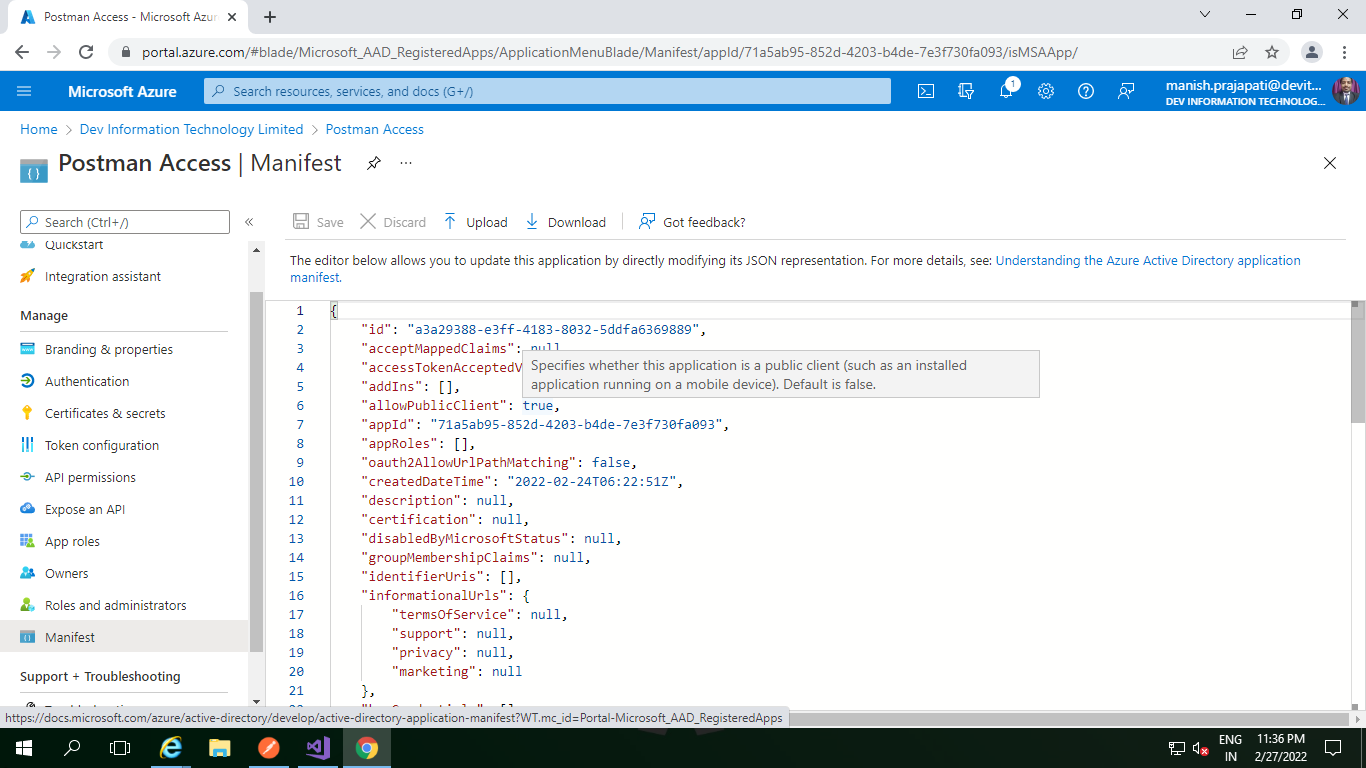


As soon as you select Dynamics D365 you will be presented the permission window. Select “**Delegated permission**” and **Check**the**User Impersonation**then click **Add Permission button.**

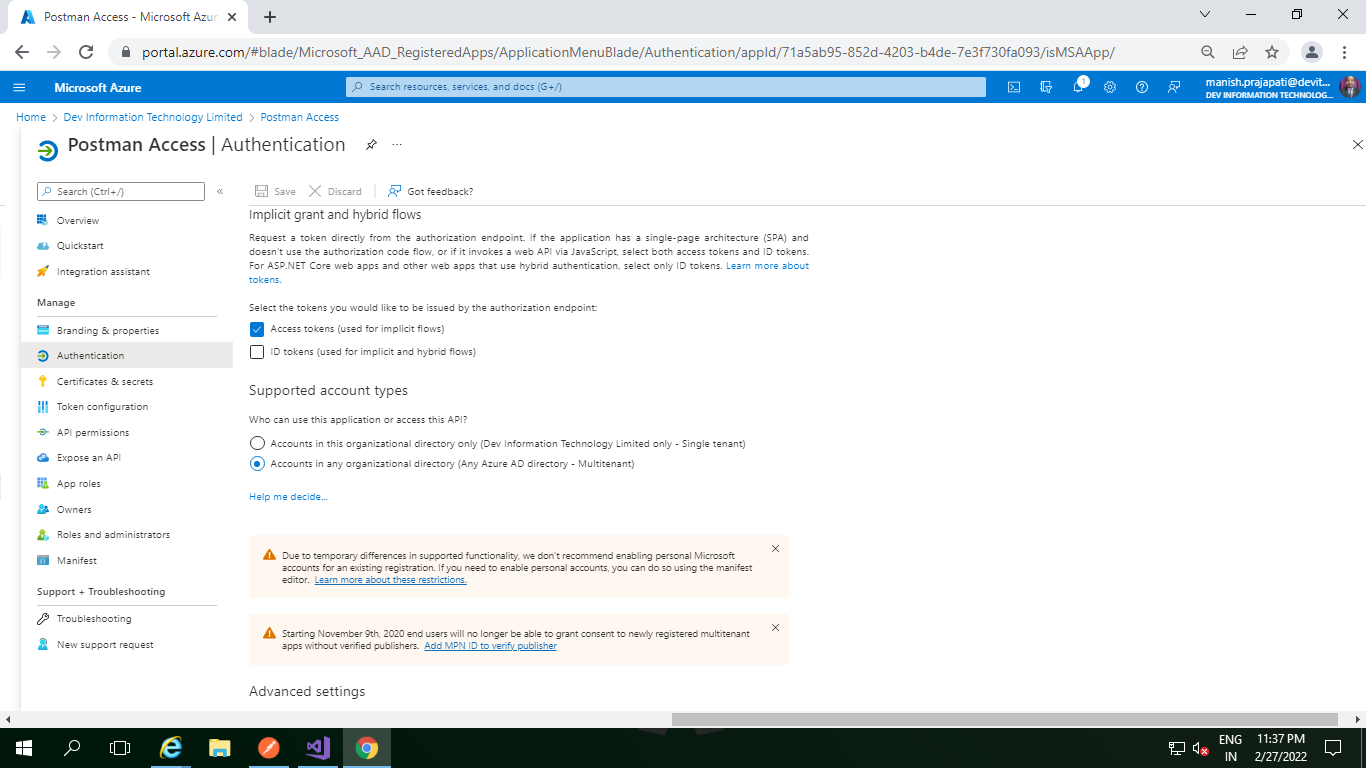




Next go to **Manifest**option and change the **allowPublicClient**flag to **true**.

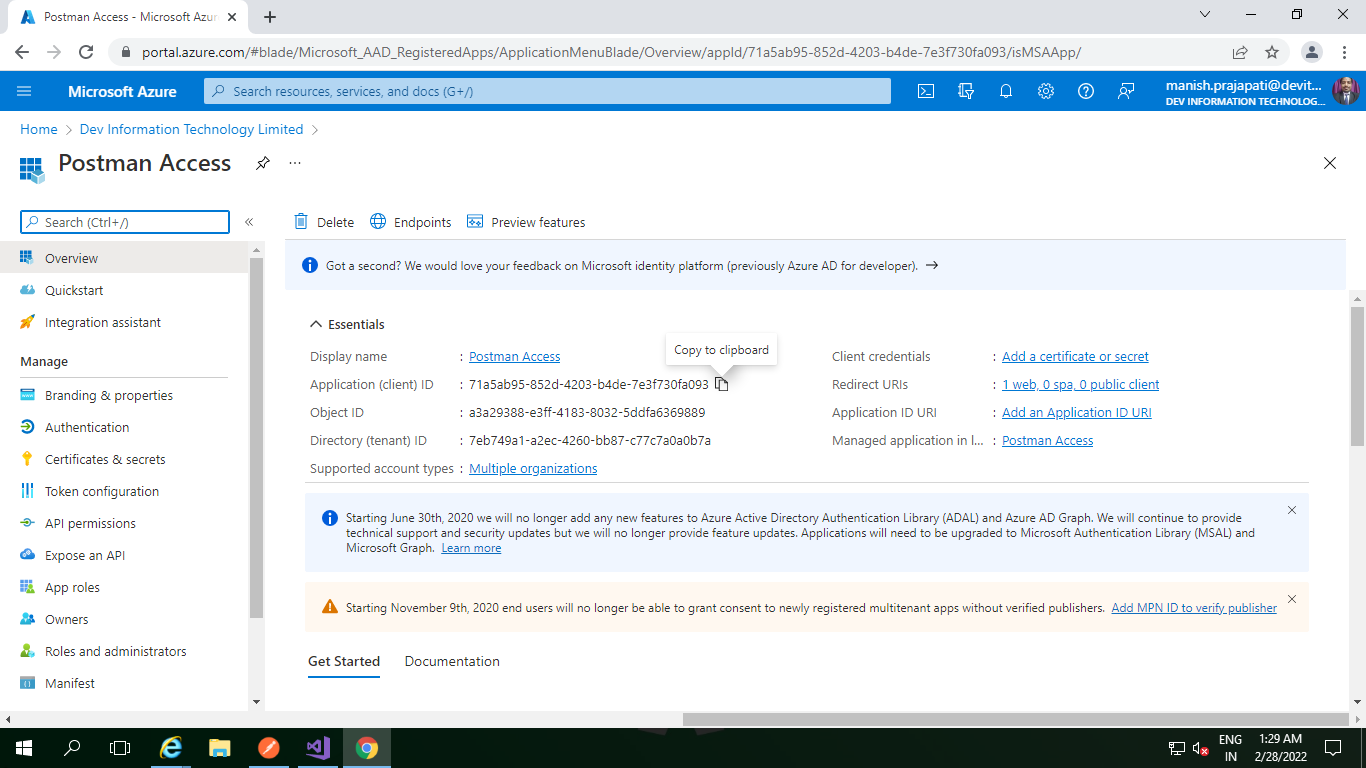


After above step we need to provide Authentication postman App. For this **click on Authentication**



Now we are ready to use the application in Postman software. Before exiting from Azure Active Directory note two things. Copy the App ID and Redirect URLs.

To copy these info navigate to Overview and click on the copy icon besides Application (client) ID option. paste in a note pad for future reference.



So now we are done with first step Now we have two things:

**Client ID: 6c79a735-77ec-4ef6-94af-75f1b0206109**

**Redirect URL : http://localhost**

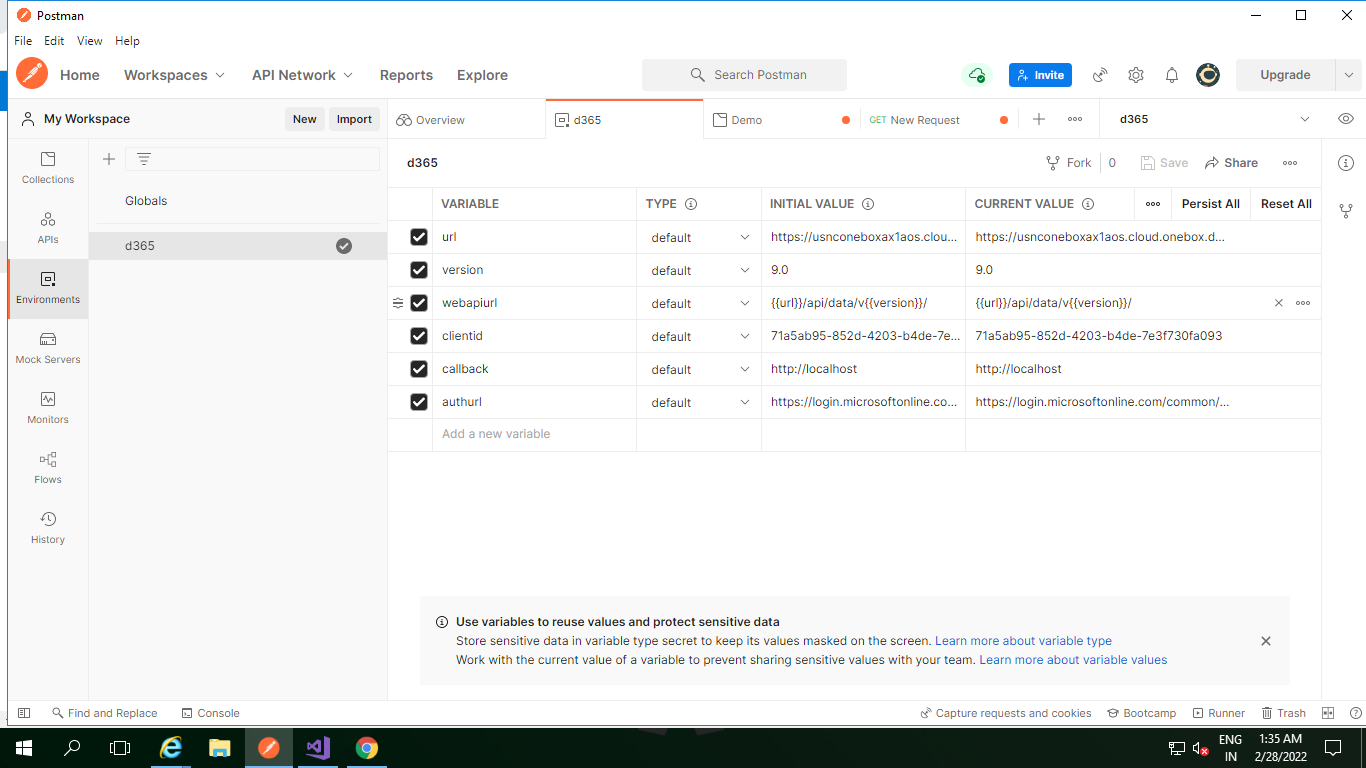
We have successfully registered an App in Azure Active Directory.

## Step:2 – Configure Postman to connect Dynamics 365 Online

Now open the Postman software if you have downloaded. Its free to use. Once you open the Postman click on +**New**from home page and choose **environment**. We are going to create an environment

Give a name of the environment and add 6 variables.

URL : use your environment URL. To get the URL open any model driven app of your environment and copy the browser address bar URL from start to till .COM as given below.

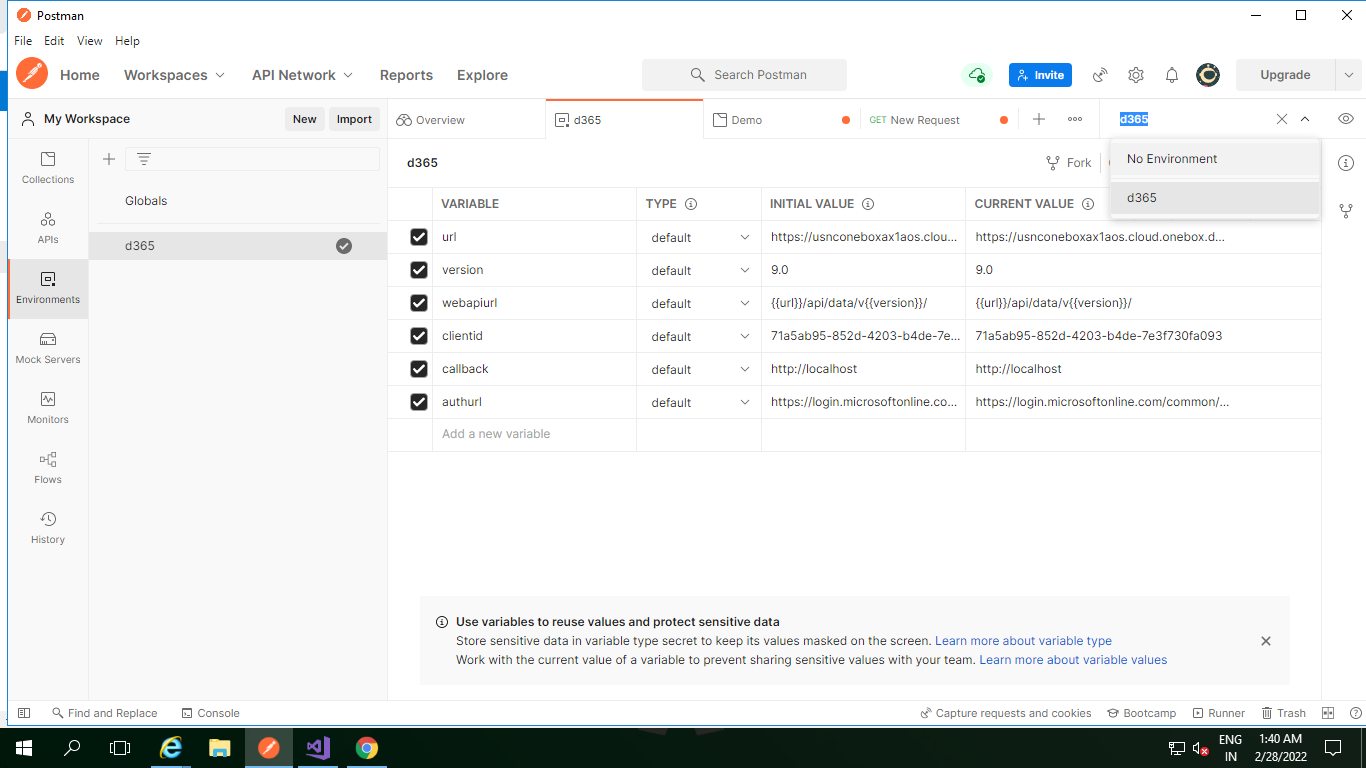


**ClientID : This is the value which is copied from Azure Active Directory App from step 1.**

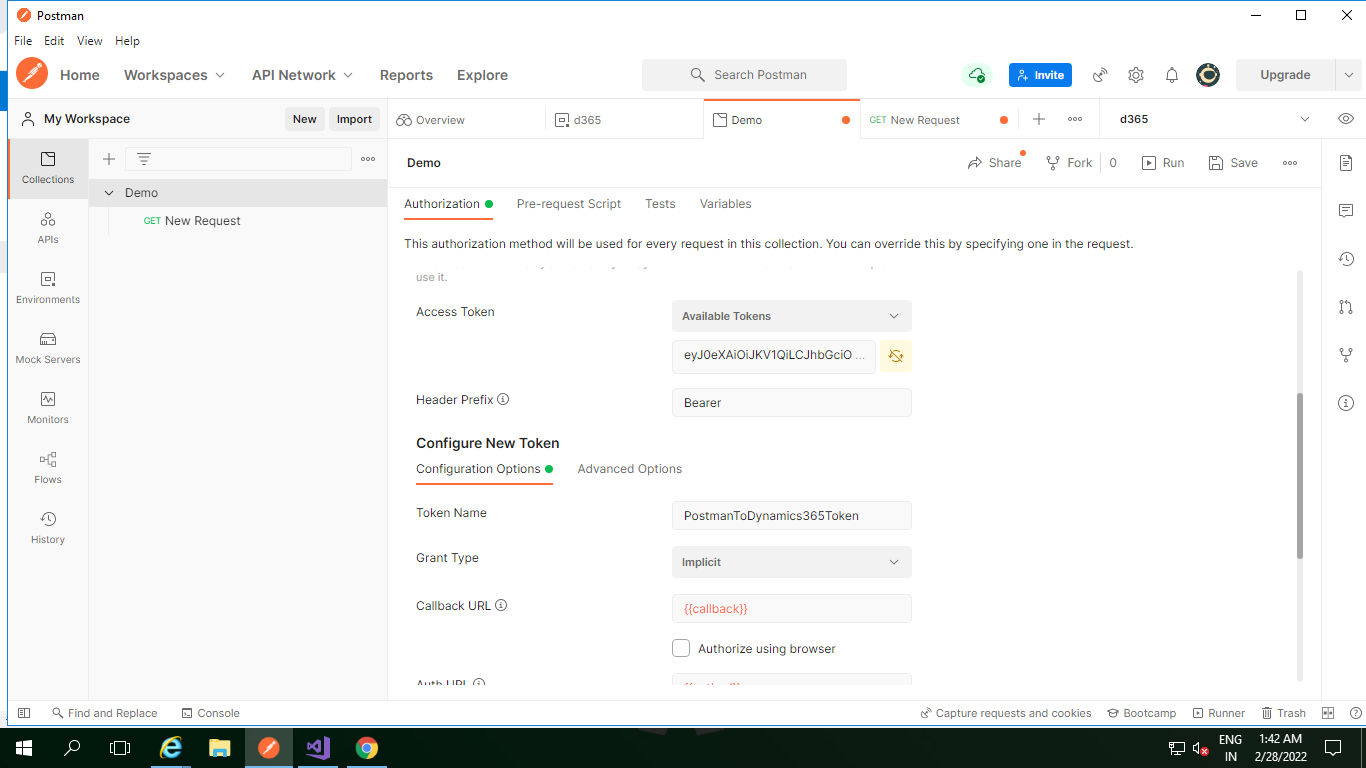
**CalllbackURL : http://localhost (This value is also captured from step 1.**

**authURL : https://login.microsoftonline.com/common/oauth2/authorize?resource={{url}}/ (Note : here the url is used as a variable in Postman)**

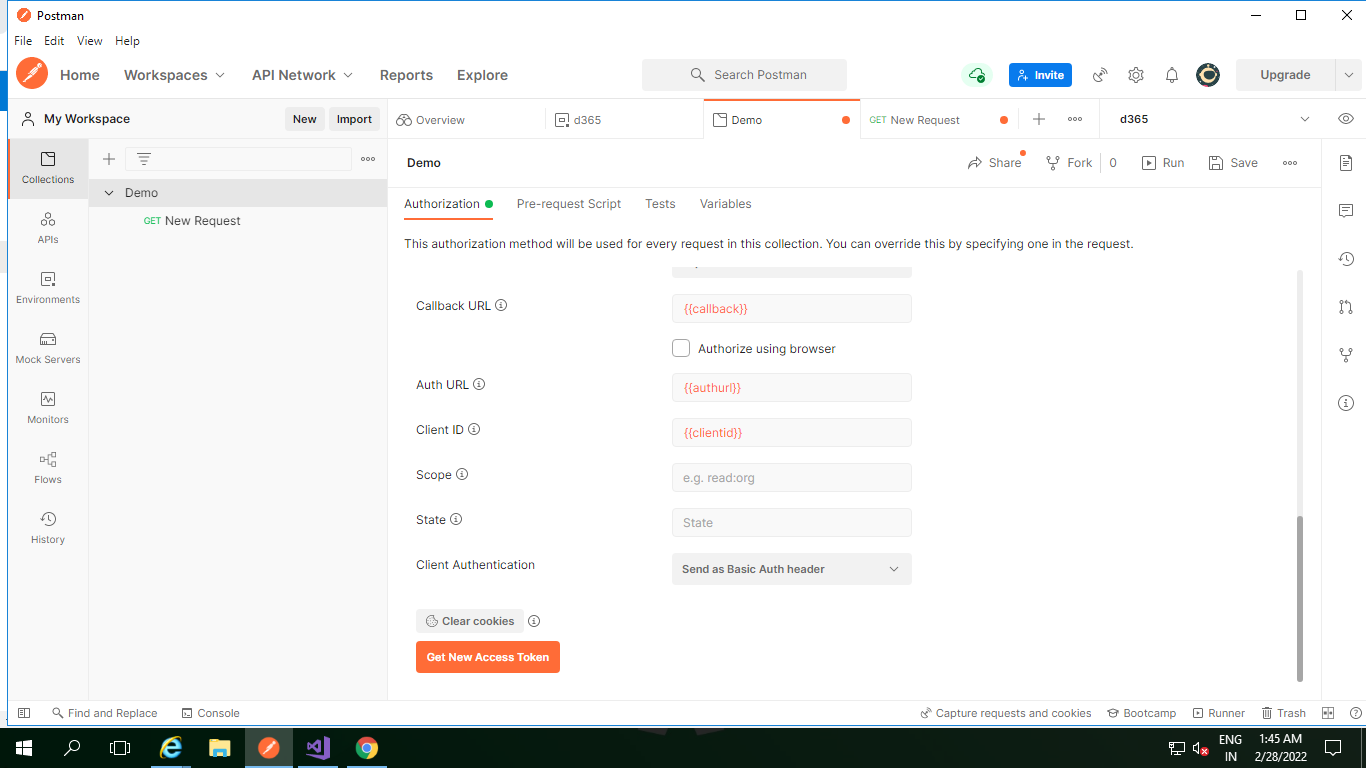
Now select the environment we just created by clicking the Environment dropdown from right top corner.

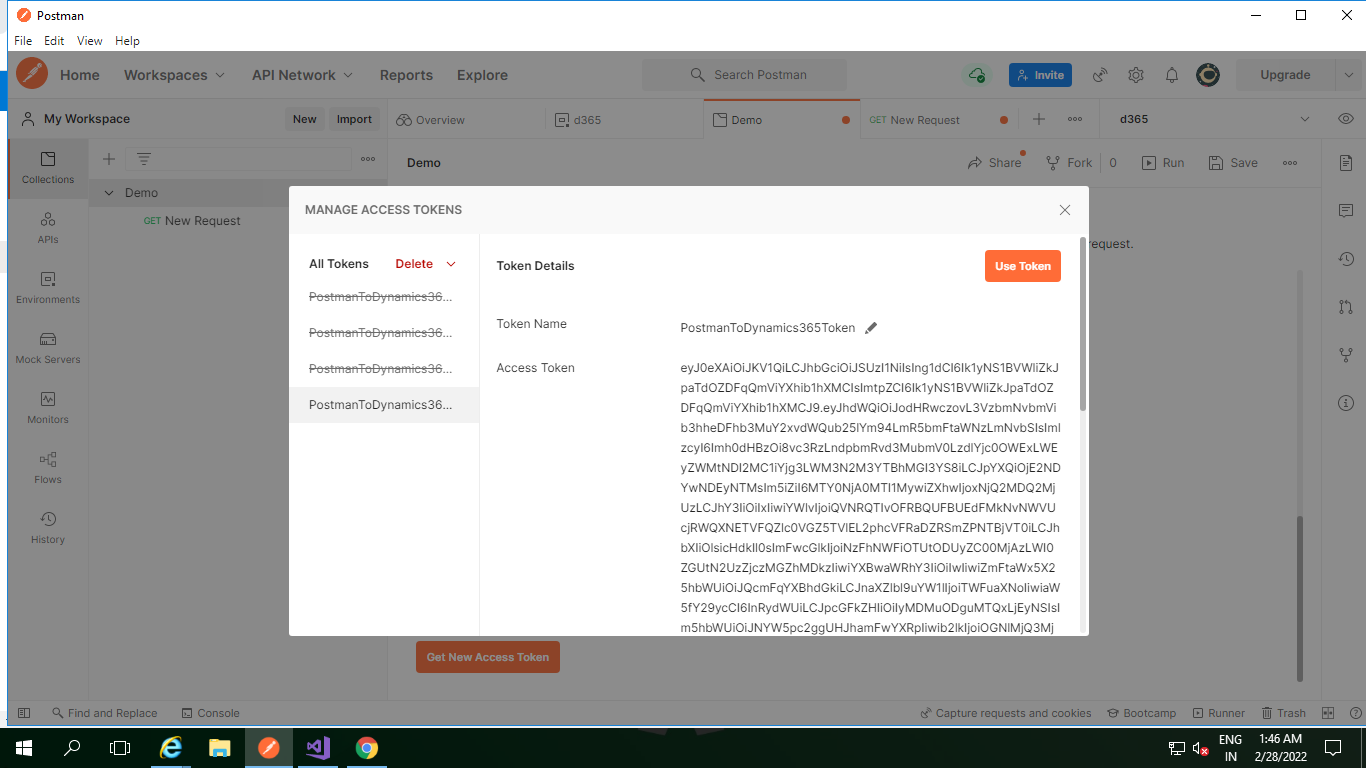


After this click on new Collections and click on + for new Request set as per below configuration.



After this click on request for token



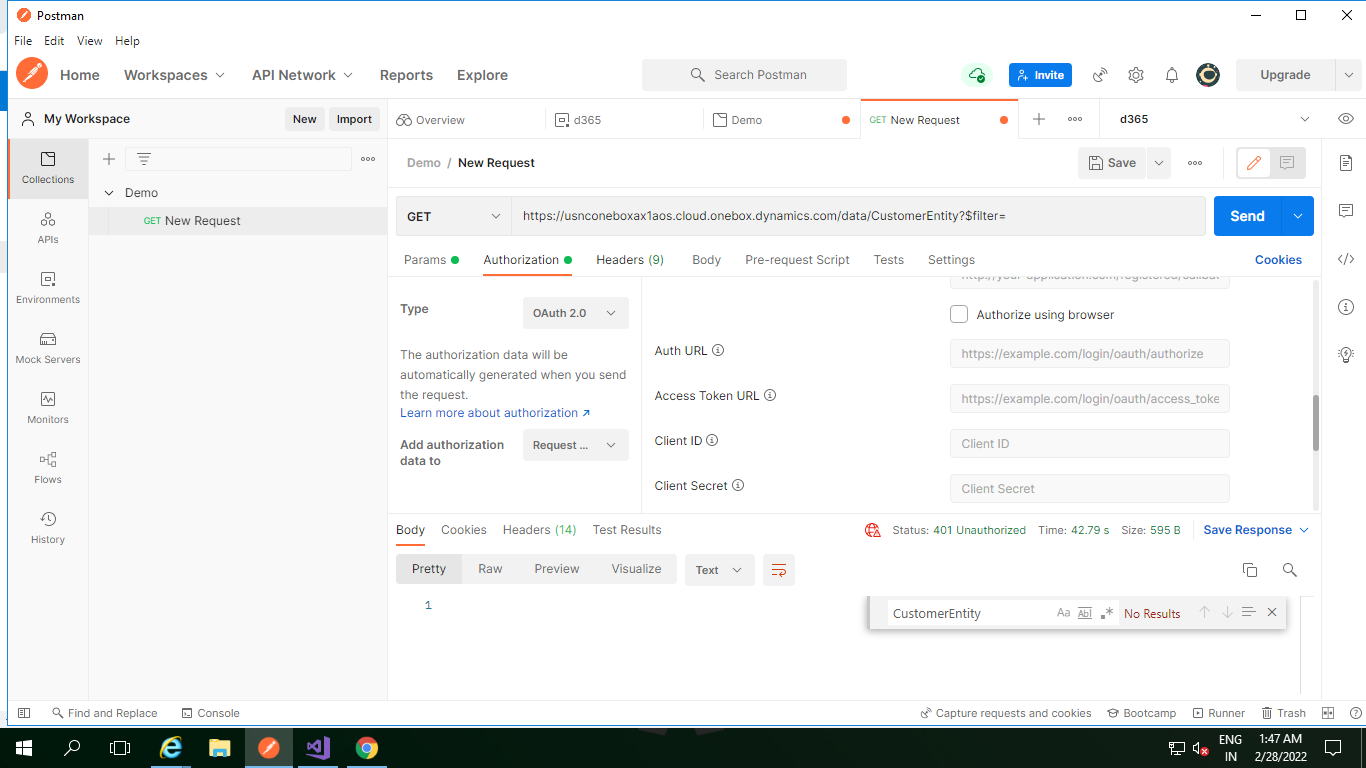


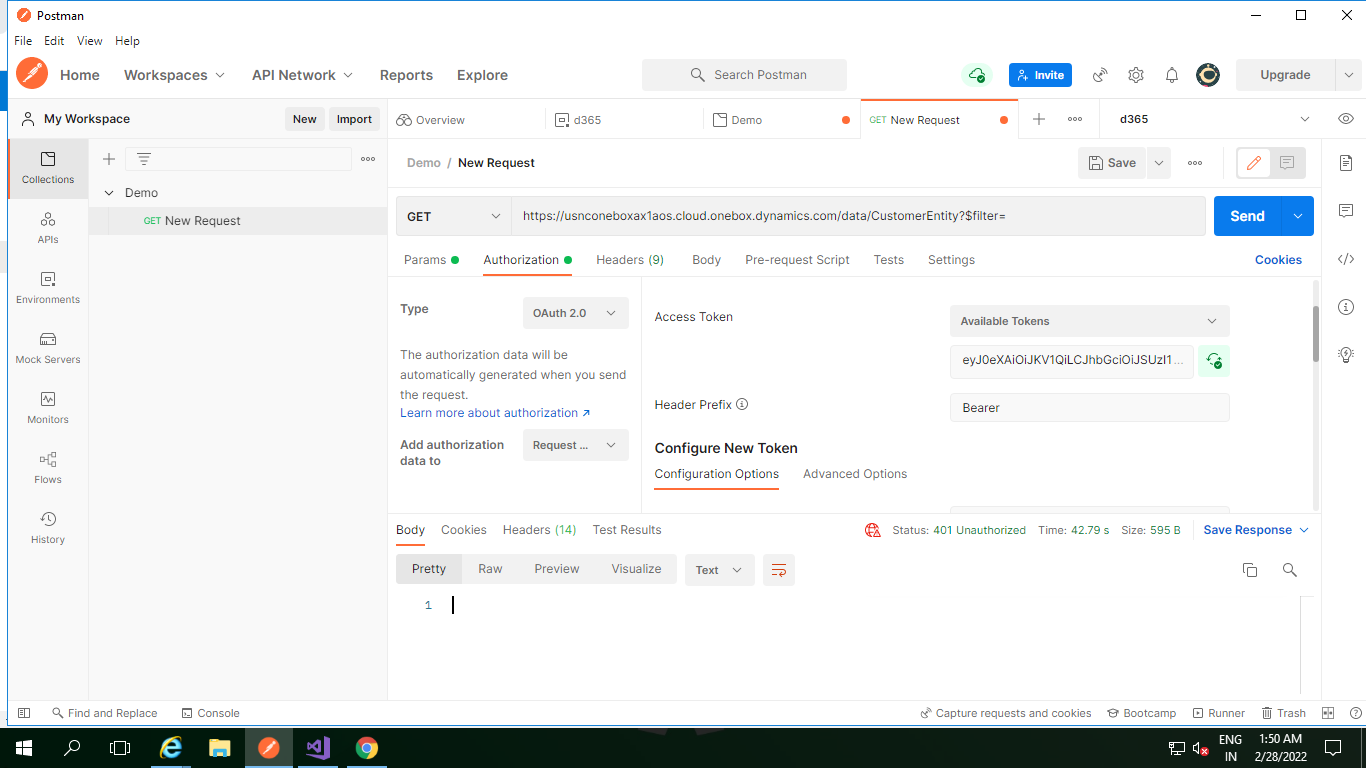
We have successfully connected with Dynamics 365 Online form Postman. Now we have to test.

## Step-3: Test API Access from Postman

To test this go to request option and choose **Get**. and in the address bar paste the address as **{{webapiurl}}/accounts** and click on **Send**.

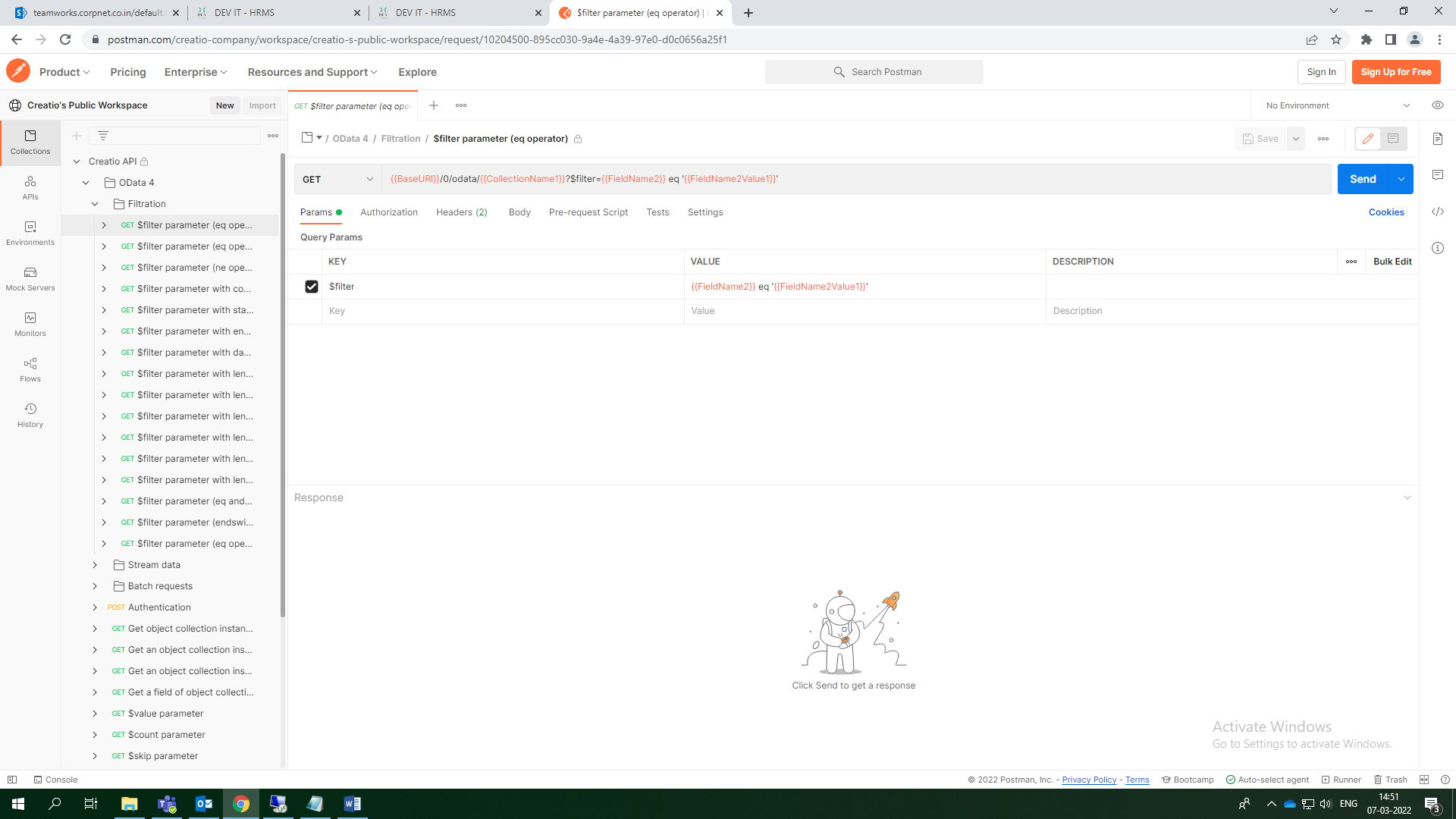
Now click on + button create Get new request set authorization OAuth 2.0





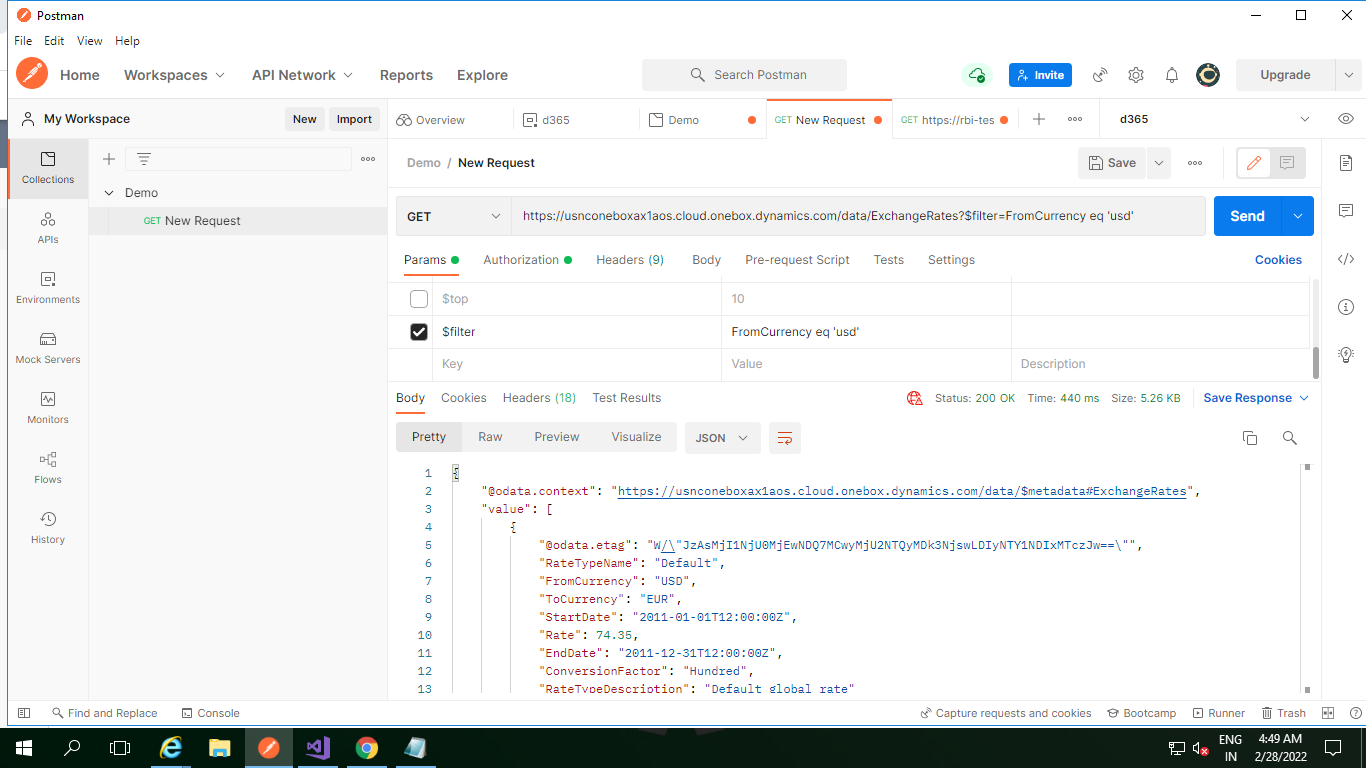
To filter data from data entity

Pass Postman filter command as per syntax as below



Now pass the value in data Entity and out put as per below Screens short

Data Entity name



Data filter command

D365 Url

## Step-4: Create Custom Data Entity

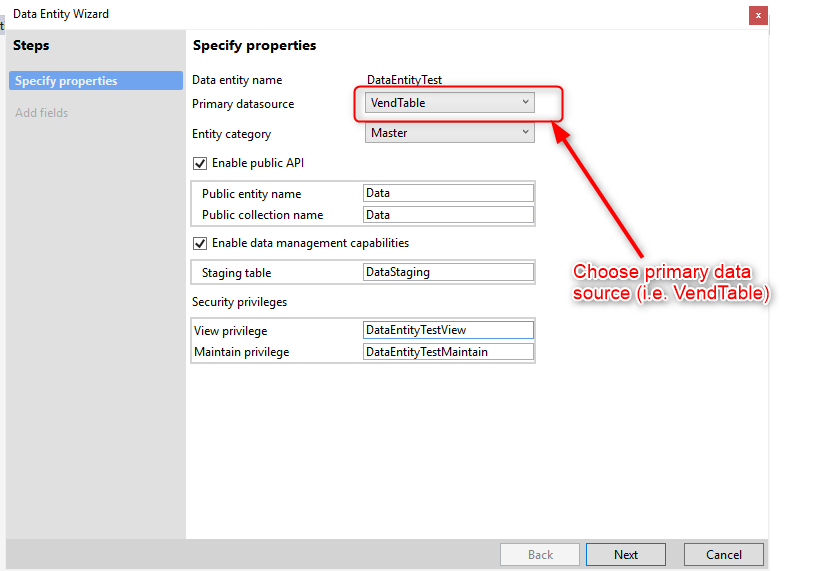
Open Visual Studio Create new Project then

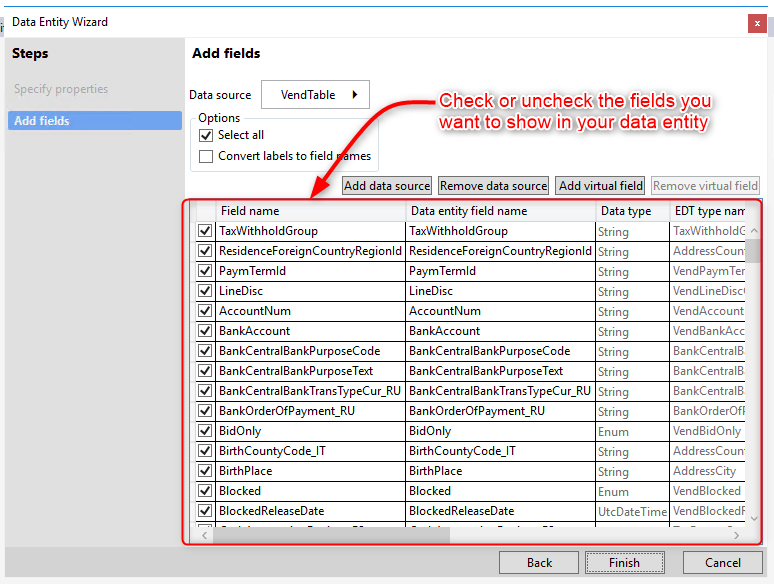
Right click on project and add new item.

## https://d365ffo.files.wordpress.com/2021/06/image-77.png?w=1024

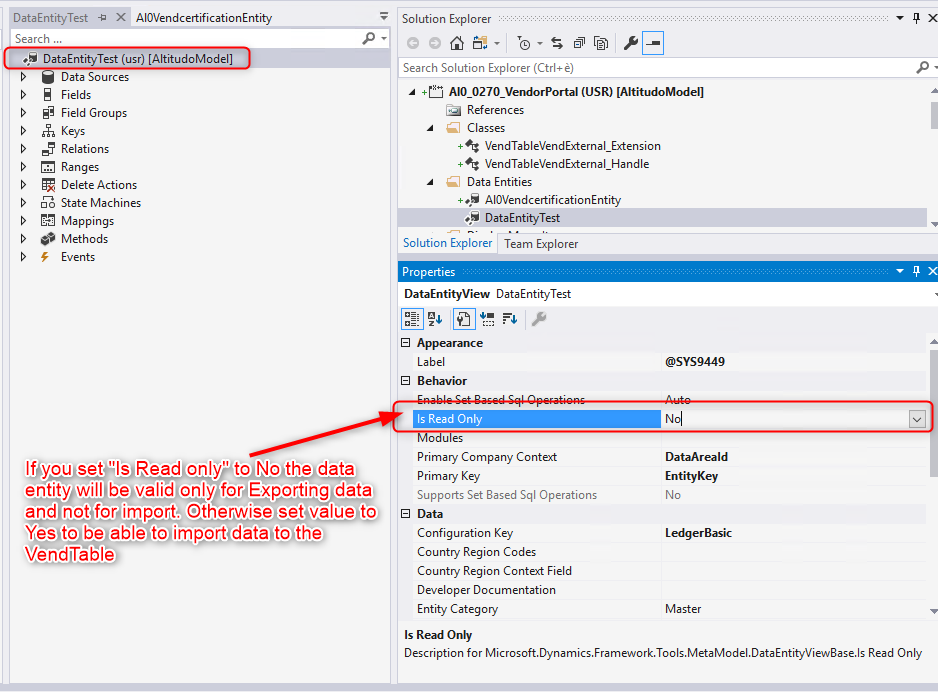
We’ll be developing a data entity for a custom table, **DataEntityTest** for this scenario.

Step 2 :- Specify data entity properties as shown below and select VendTable table as the **Primary datasource**. Click **Next**.

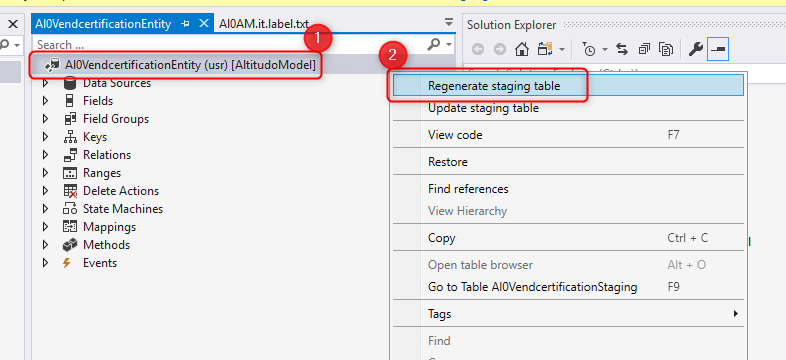


Step 3:- On the next screen of the wizard, review the data entity fields. You may choose to **Convert labels to field names**. Click **Finish**. 

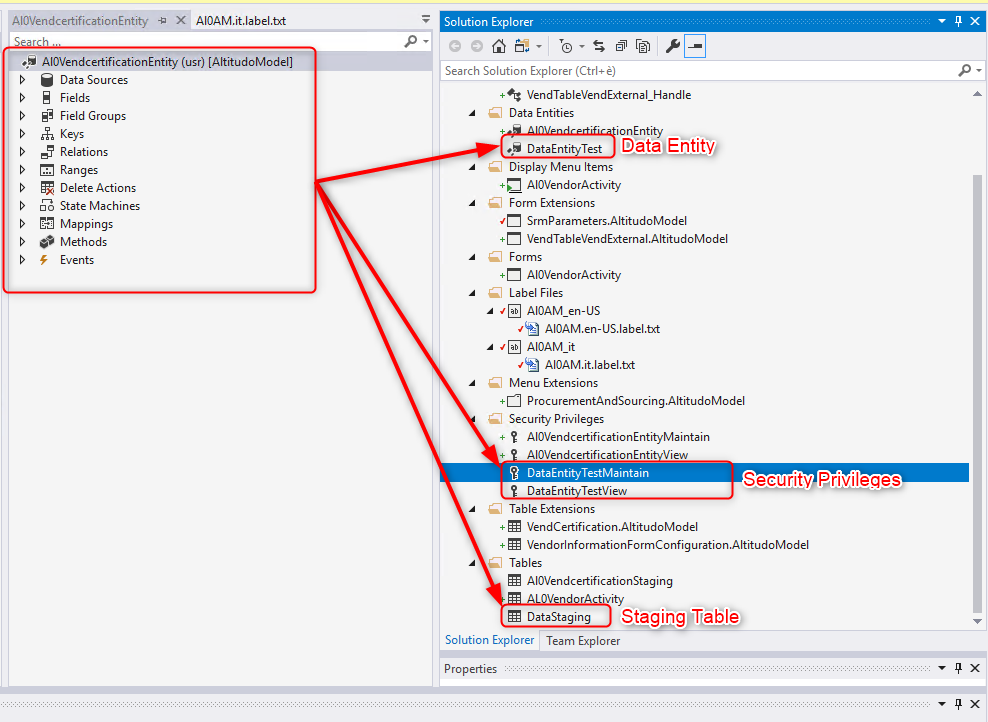
If you set “Is Read only” to No the data entity will be valid only for Exporting data and not for import. Otherwise set value to Yes to be able to import data.



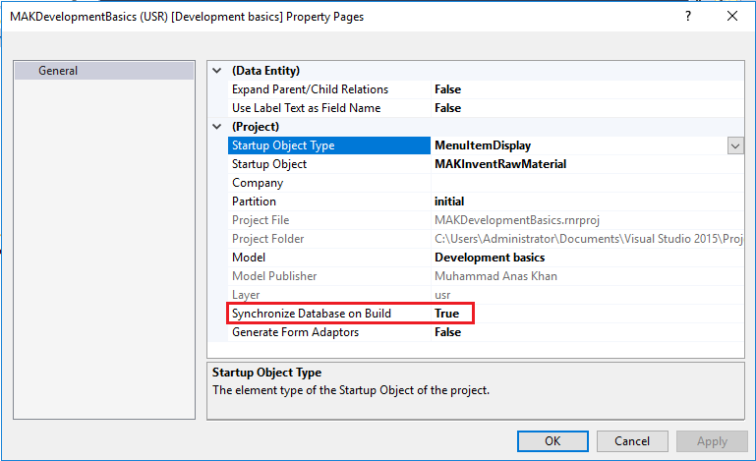
Staging table must be generated. You can do it by right clicking Data Entity and click on “Regenerate staging table”



Step 4:- The following highlighted development artifacts must be created.

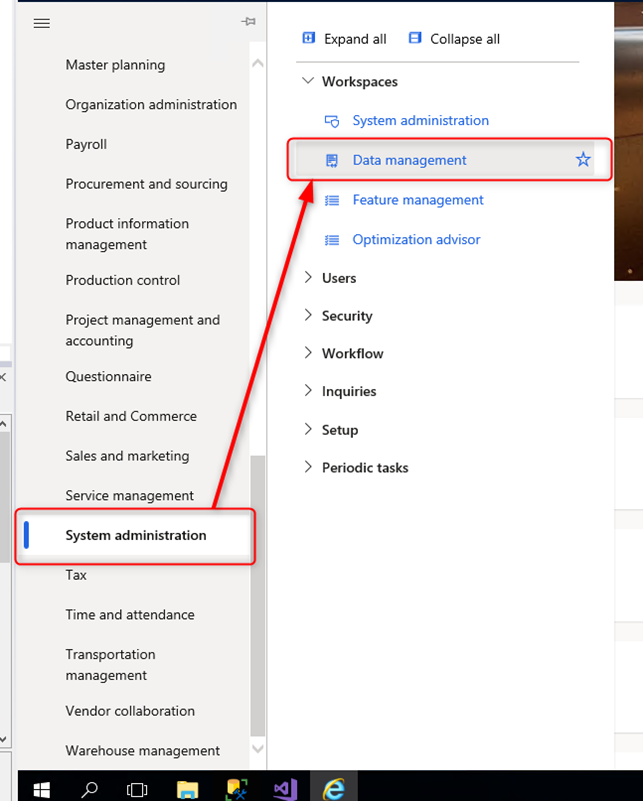


Step 5:- Open project properties and set **Synchronize Database on Build** to **True** to synchronize the newly created table and data entity with database

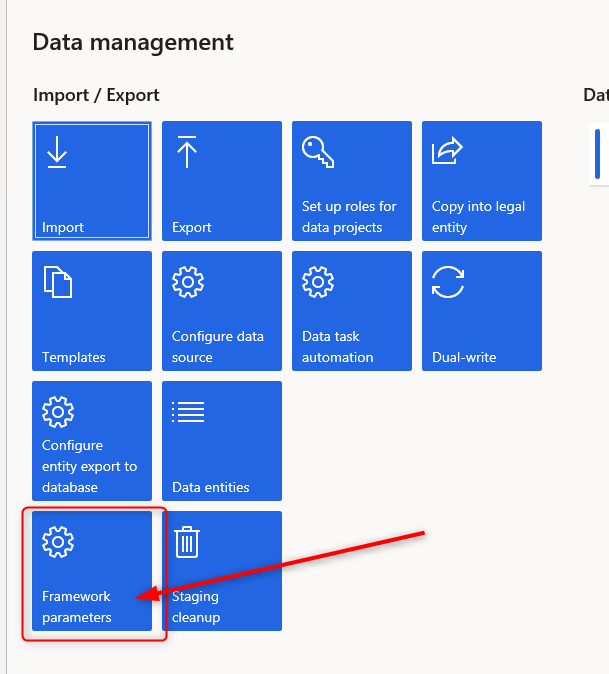


Step 6:- Now you have to Refresh data entity to make it visible in Data Entity framework in D365FO.

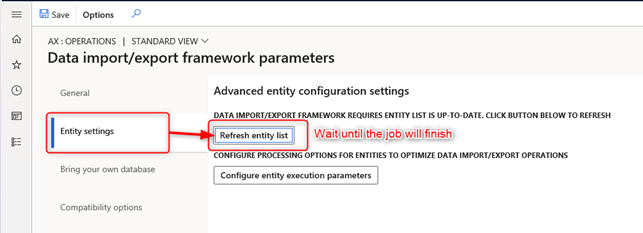
Open D365FO, go to System Administrator, Data Management



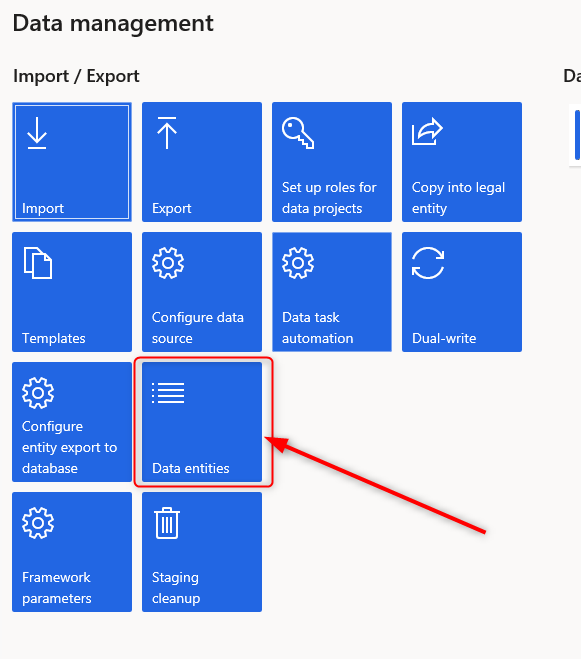
Click on Framework parameters

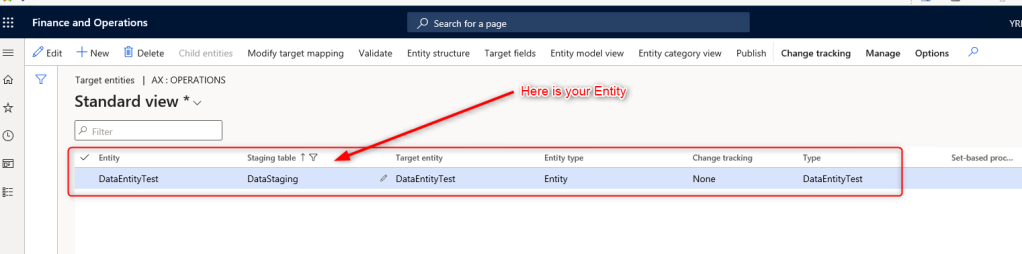


Switch to Entity settings and click on “Refresh Entity list”. Wait until job will finish. It will take some times.

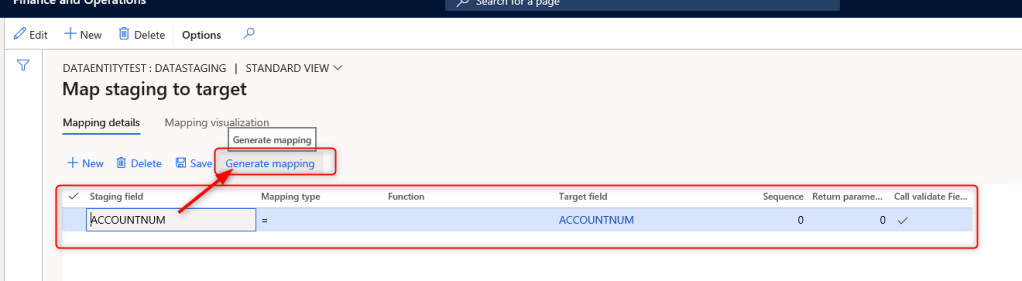


After job has finished you can search your Entity by clicking on Data Entity Tile and search the entity





Click on “Generate mapping”



If you click on “Mapping Visualization” you’ll see the mapping.

For convenience I have entered only one field (AccountNum) and Click Save.

