**How to implement OPID/Identity for Web API to Web API**

**With JWT**

Structure:

Web App MVC

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_Web API

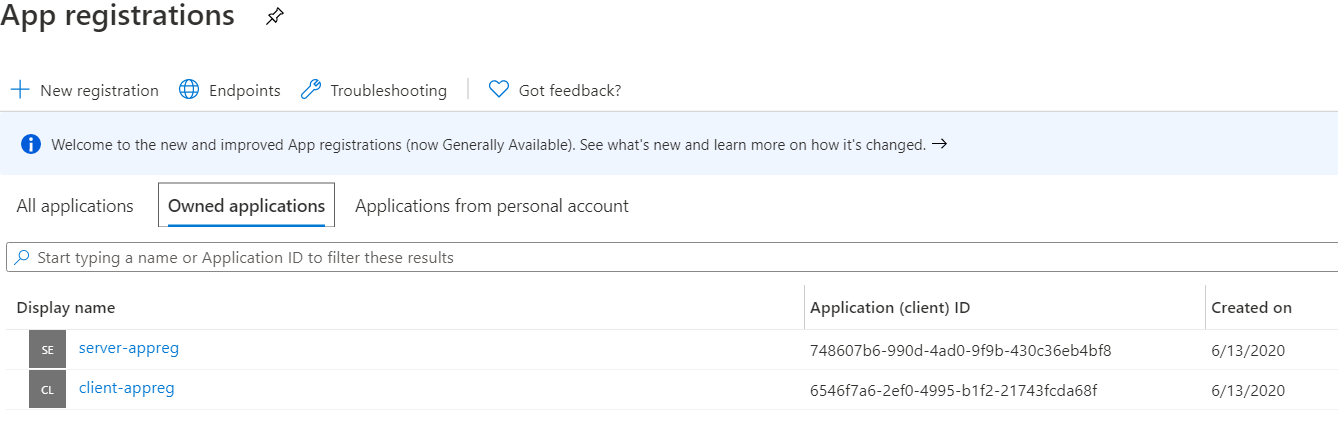
[Here both are secured by [Authorize] attribute using JWT Token from Azure AD]

For code sample refer at the bottom pf the page.

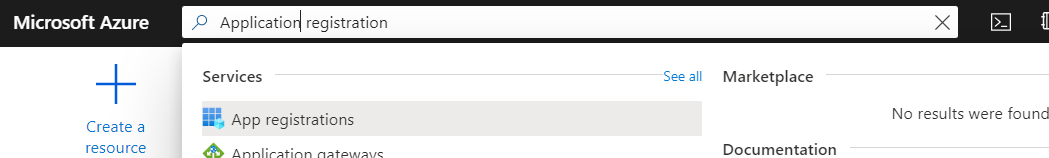
Steps:

1: Do the App Registration as below (1 for Client MVC/API and 1 for Server API)

**SERVER API REGISTRATION**

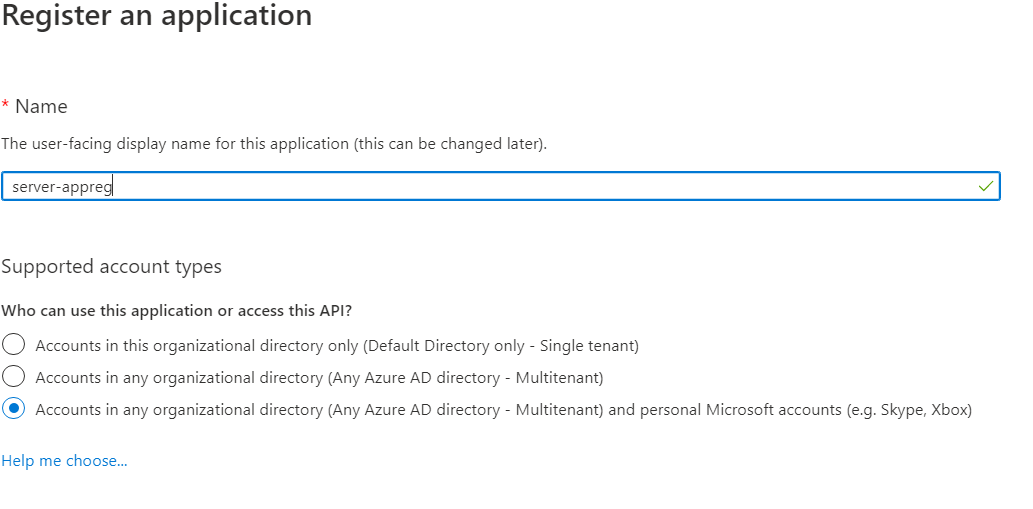


For this go to Azure Portal and Search by:



2: Click App Registration and then “New Registration” in the UI

3: Name is ‘server-appreg’ and choose

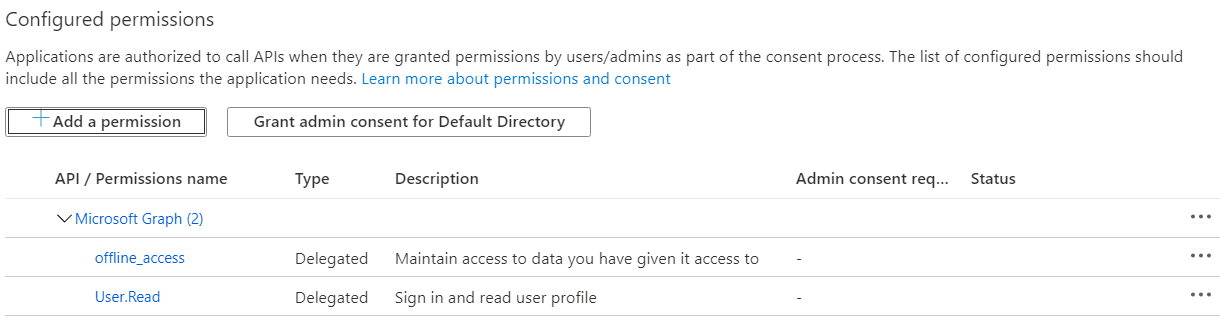


Click Register

4: Go to the OverView menu of the registration newly created and look @ the ClientID and TenantID

>> **API Permissions Menu**

1: Click Add Permission > Select Tab “Microsoft API” > Click Microsoft Graph > Delegated Permission > (offline\_access) and (User.Read) Access select from the option to make it like:

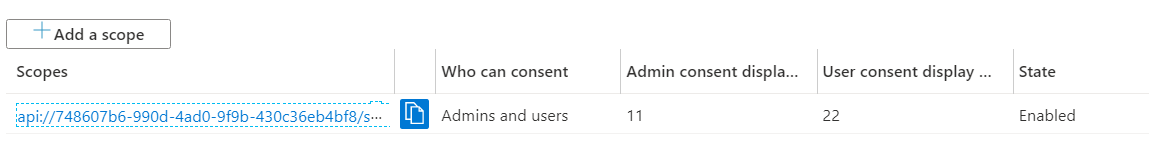


>> **Expose an API Menu**

1: Click “Add a Scope”

2: Give any scopename like ‘service\_scope’ and enter/chose Consent data with “Admin & User” can consent.

3: Add the Scope to get something like:



**>> Manifest**

Click and just have a look of Element below that indicates OAuth2 is being used.

"accessTokenAcceptedVersion": 2,

**CLIENT API REGISTRATION**

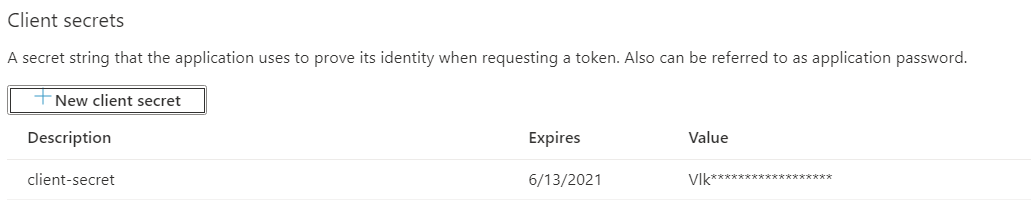
1: Create Another App Registration by Clicking “New Registration” and name it “client-appreg”

2: Note the Client ID and Tenant ID

**>> Certificates and Secret Menu**

1: Click “New client secret”

2: Name it “client-secret” , put 1 or 2 yrs as per the need to get it like



3: Copy immediately secret value

**>> Authentication**

1: Add a Platfrom and chose Web

2: put the url <https://localhost:44321/signin-oidc> and click configure

3: Similarly set the Logout URL as : <https://localhost:44321/signout-callback-oidc>

4: In the Implicit Grant, select [ID Token]

**>> API Permission**

1: Make sure you have User.Read and offline\_access are turned on, if not add them by clicking “Add Permission” > MS API > MS Graph > Delegated Permission

Finally the required Items are:

For Client App

"Domain": "[Enter the domain of your tenant, e.g. contoso.onmicrosoft.com]",

"TenantId": "get the client app clientid value from above",

"ClientId": " get the client app tenant value from above",

"ClientSecret": "get it from Client App Registration > Cert & Secrets menu ‘’"

"TodoListScope": "api://this is server-appreg client id/service\_scope",

[TodoListScope] – this scope can be found in Server-appreg >> Expose an API Menu as

api://748607b6-990d-4ad0-9f9b-430c36eb4bf8/service\_scope

For Server API App

"Domain": "[Enter the domain of your tenant, e.g. contoso.onmicrosoft.com]",

"TenantId": "53b0d682-d636-43fa-9d5f-842efbdb7a88",

"ClientId": "748607b6-990d-4ad0-9f9b-430c36eb4bf8"

Final Client Configuration in c#: appsettings.json

|  |
| --- |
| {    "AzureAd": {      "Instance": "https://login.microsoftonline.com/",      "Domain": "[Enter the domain of your tenant, e.g. contoso.onmicrosoft.com]",      "**TenantId**": "53b0d682-d636-43fa-9d5f-842efbdb7a88",      "**ClientId**": "6546f7a6-2ef0-4995-b1f2-21743fcda68f",      "**CallbackPath**": "/**signin-oidc**",      "**SignedOutCallbackPath** ": "/signout-callback-oidc",      // To call an API      "ClientSecret": "VlktQfI~po2jR1VUWfAv2t793n-.Xzg\_65"    },    "TodoList": {      "TodoListScope": "api://748607b6-990d-4ad0-9f9b-430c36eb4bf8/service\_scope",      "TodoListBaseAddress": "https://localhost:44351"    },    "Logging": {      "LogLevel": {          "Default": "Information",          "Microsoft": "Warning",          "Microsoft.Hosting.Lifetime": "Information"      }    },    "AllowedHosts": "\*"  } |

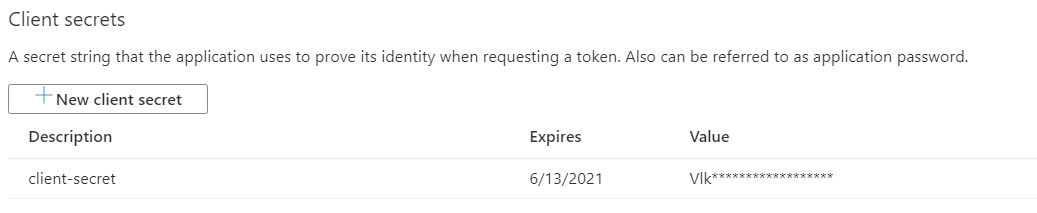
Server: appsettings.json

|  |
| --- |
| {    "AzureAd": {      "Instance": "https://login.microsoftonline.com/",      "Domain": "[Enter the domain of your tenant, e.g. contoso.onmicrosoft.com]",      "**TenantId**": "53b0d682-d636-43fa-9d5f-842efbdb7a88",      "**ClientId**": "748607b6-990d-4ad0-9f9b-430c36eb4bf8"    },    "Kestrel": {      "Endpoints": {        "Http": {          "Url": "http://localhost:44351"        }      }    },    "Logging": {      "LogLevel": {        "Default": "Warning"      }    },    "AllowedHosts": "\*"  } |

Quick Reference:

1: How to get **ClientSecre**t Scope id for Service API/Downstream API

1. Go to Application registration
2. Click the ‘server=appreg’
3. Select Menu “Certificates & Secrets”



The secret value to be added in appsettings.json in client app

2: How to get **ClientScope**

1. Go to Application Registration >> ‘server-appreg’
2. Goto Menu “Expose an API”
3. Get the Scope value

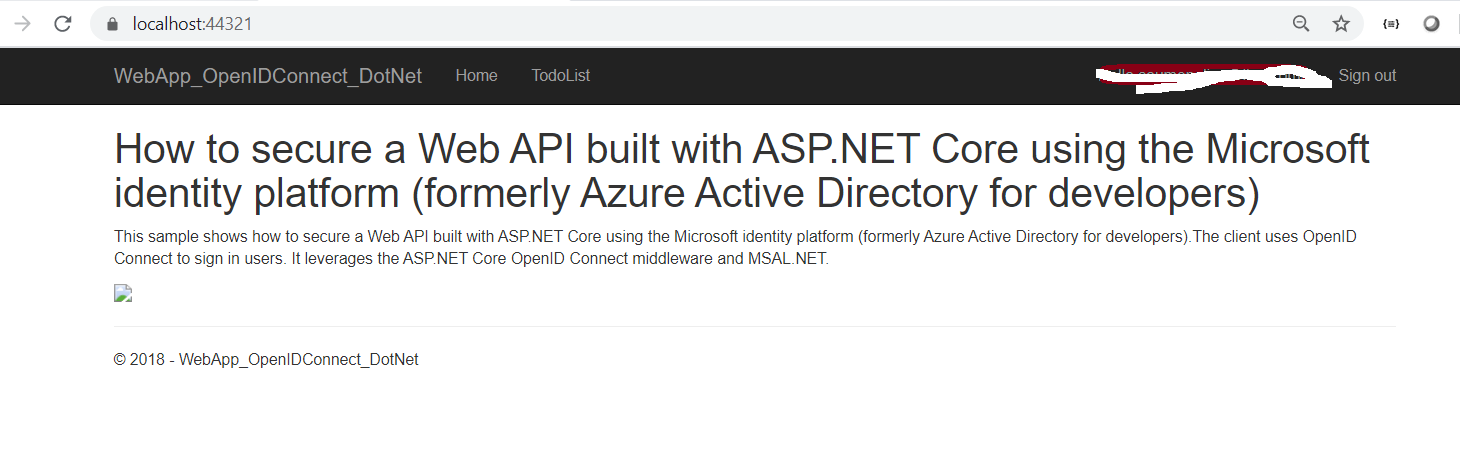
Also in this case it is TodoListScope in json

SOURCE CODE:

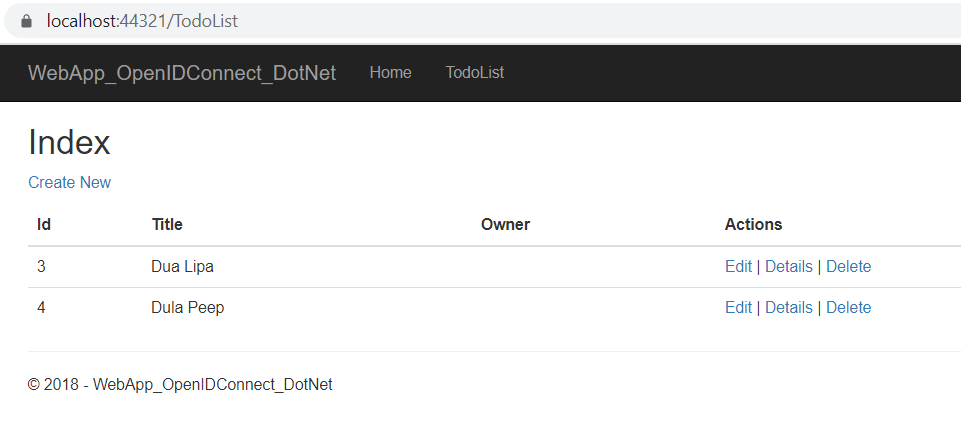
|  |  |
| --- | --- |
|  | <PackageReference Include="Microsoft.Graph" Version="1.16.0" />  <PackageReference Include="Microsoft.Identity.Web" Version="0.1.3-preview" />  <PackageReference Include="Microsoft.Identity.Web.UI" Version="0.1.3-preview" />  Install-Package Microsoft.Identity.Web -Version 0.1.4-preview  Install-Package Microsoft.Identity.Web.UI -Version 0.1.0-preview |

This lab has been created referring Microsoft Labs found inside Readme file in the .zip, where you have complete git url from Microsoft to test, practice.

Screen for Client App:



Click TodoList for (This TodoList is JWT Authenticated)



For Authorization Feature, please refer the doc: Azure AD\_RBAC\_WithJWT.docx in the git