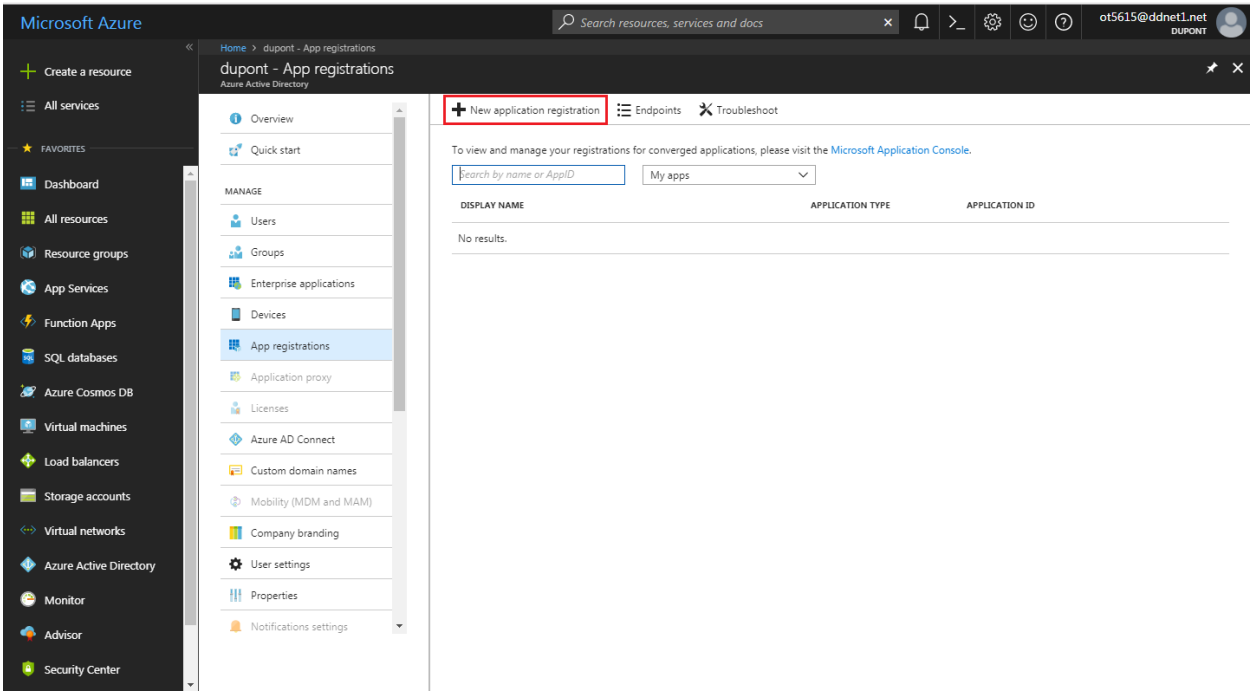
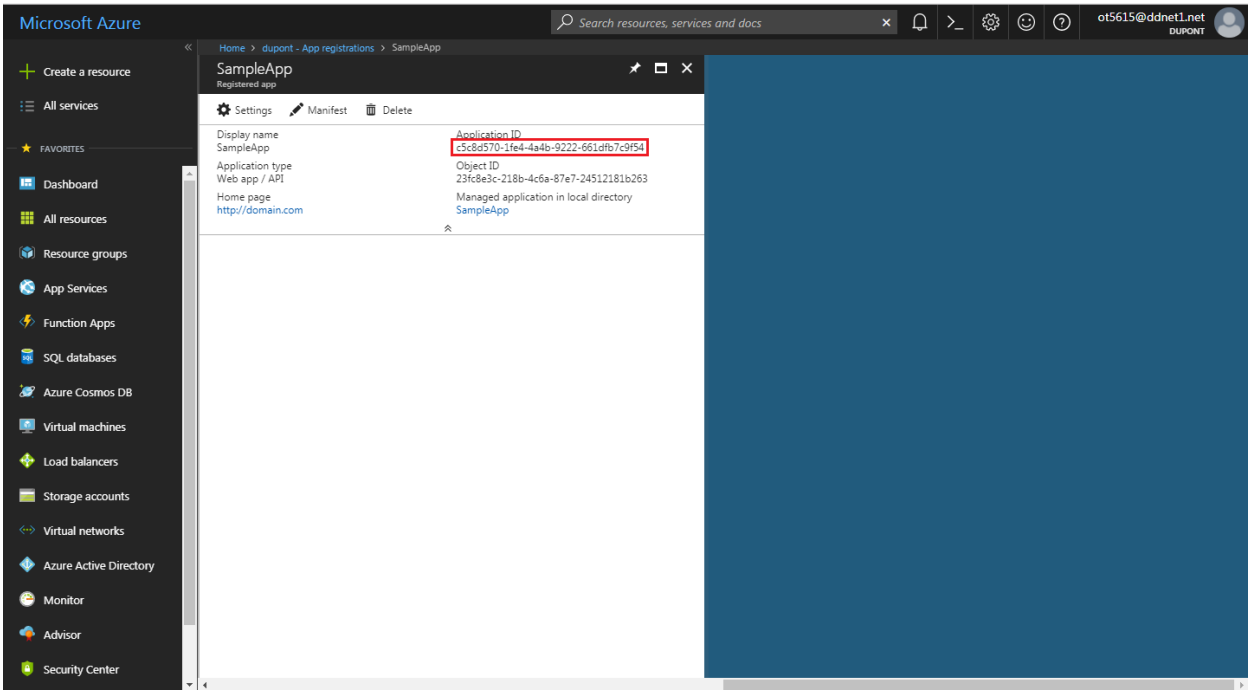
Querying the Microsoft Graph API to get User Details from Azure Active Directory

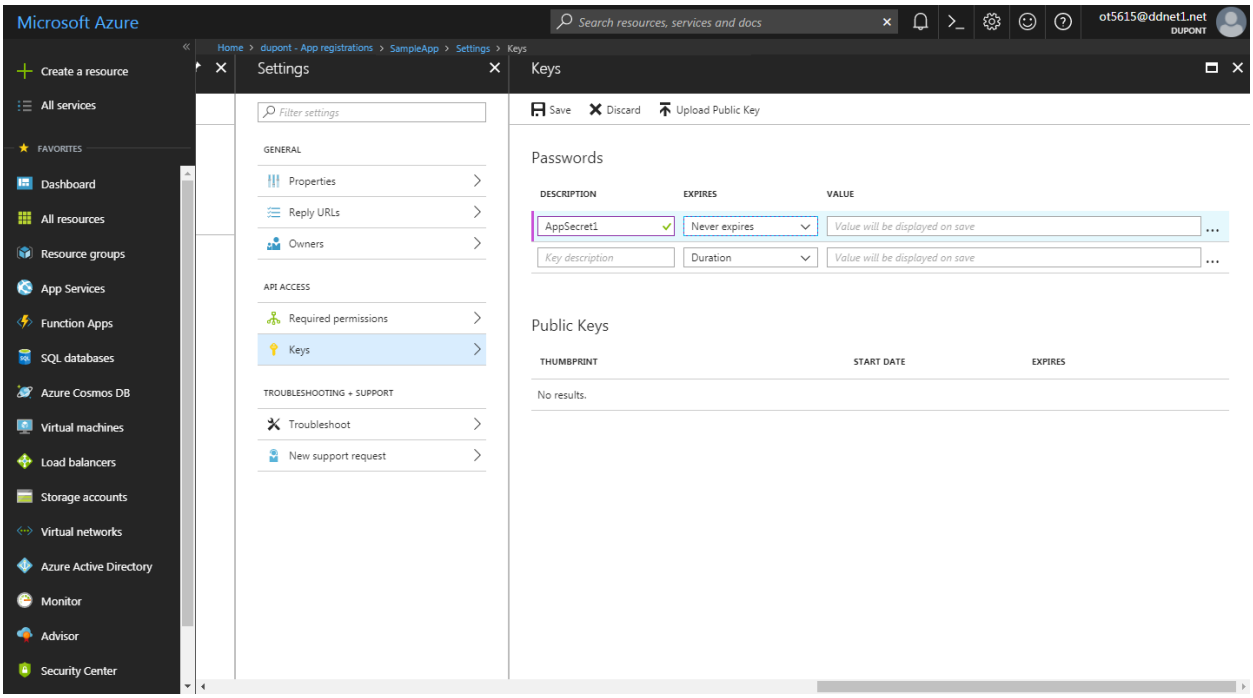
1. Register the application to azure active directory



2. Get, AppId, AppSecret and ClientId



Generate AppSecret Key



3. Write these keys in the web.config File

```
<appSettings>
    <add key="ida:ClientId" value="Your_App_Id" />
    <add key="ida:ClientSecret" value="App_Secret" />
    <add key="ida:TenantId" value="Your_Tennant_Id" />
</appSettings>
```

4. Get all the keys to .cs file

```
private static string clientId = ConfigurationManager.AppSettings["ida:ClientId"];
private static string appKey = ConfigurationManager.AppSettings["ida:ClientSecret"];
private static string tenantId = ConfigurationManager.AppSettings["ida:TenantId"];
```

5. Generate URL

```
private string GetUserUrl(string fname, string lname)
{
    // Add a filter and append to URL
    string filter = $"$filter=givenName eq '{fname}' and surname eq '{lname}'";
    return string.Format("https://graph.windows.net/{0}/users?{1}&{2}", tenantId, filter, "api-version=1.6");
}
```

6. Generate BearerAccess Token using the Keys

```
private async Task<string> GetBearerAccesToken()
{
    string result = string.Empty;

    // Get OAuth token using client credentials
    string authString = "https://login.microsoftonline.com/" + tenantId;

    AuthenticationContext authenticationContext = new AuthenticationContext(authString, false);

    // Config for OAuth client credentials
    ClientCredential clientCred = new ClientCredential(clientId, appKey);

    string resource = "https://graph.windows.net";

    AuthenticationResult authenticationResult =
                                    await authenticationContext.AcquireTokenAsync(resource, clientCred);
    result = authenticationResult.AccessToken;

    return result;
}
```

7. Query the API

```
public async Task<string> GetUsers(string fname, string lname)
{
    string result = string.Empty;

    // Get the Token for Authentication
    string accessToken = await GetBearerAccesToken();

    using (var client = new HttpClient())
    {
        using (var request = new HttpRequestMessage(HttpMethod.Get, GetUserUrl(fname, lname)))
        {
            // Add Bearer token to Authorization Header before sending the request
            request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", accessToken);

            // Sending the Request
            using (var response = await client.SendAsync(request))
            {
                // If Status code is OK(200), Read the Response
                if (response.StatusCode == HttpStatusCode.OK)
                {
                    var json = JObject.Parse(await response.Content.ReadAsStringAsync());
                    result = json.ToString();
                }
            }
        }
    }

    return result;
}
```