Server-to-Server (S2S) Authentication with Dynamics 365

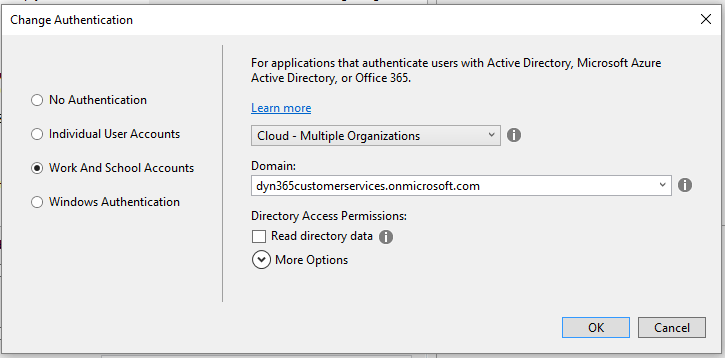
# Introduction

With server-to-server (S2S) authentication, external services like web applications can seamlessly and securely connect to Dynamics 365 in a controllable manner. Typical scenarios for this service are third-party applications and internal customer applications that need to integrate with Dynamics 365 data.

## Requirements

* Visual Studio with web developer tools installed
* A December 2016 update for Microsoft Dynamics 365 (online) tenant associated with your Azure Active Directory (Azure AD) tenant.

## Setup

* Create an MVC web application
* Using Visual Studio, create a new MVC web application and register it with your Azure AD tenant.
* Open Visual Studio, Click New Project and select .NET Framework 4.6.1 and the ASP.NET Web Application template.
* Click OK, and in the New ASP.NET project dialog select MVC.
* Click the Change Authentication button and select Work and School Accounts. 
* Select Cloud – Multiple Organizations and click OK

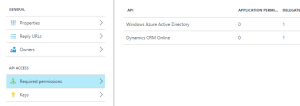
if you get the following error “Error: The user account ‘user@domain.onmicrosoft.com’ doesn’t have the required permissions to access the domain ‘domain.onmicrosoft.com’.” make sure your visual studio is singed in with the same credentials you are trying to use to connect to Azure AD.

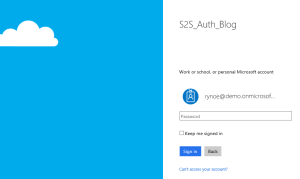
 if you navigate to [https://portal.azure.com](https://portal.azure.com/) under App Registrations you will see the app created by Visual Studio.

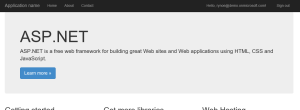
You will need to give you app access to your Dynamics 365 data.

Under API Access area click Required permissions

Click Add -> select API -> Dynamics CRM Online and click select

Click Done to save permission changes

Run your MVC application and you will be prompted to login using your Online credentials.



Open you Visual Studio solution that you created in the previous post.

Web.config

Add the following to the appSettings section

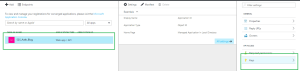
<add key="ida:OrganizationHostName" value="https://{0}.crm4.dynamics.com"/>

<add key="owin:appStartup" value="<your app namespace>.Startup" />

<add key="ida:ClientSecret" value="KEY HERE" />

Login to Azure AD navigate to App Registrations.

Select the application you created, then navigate to Keys.



Copy the newly generated code and paste it into the ida:ClientSecret value in your web.config file.

Set Home Controller access as anonymous

Navigate to \Controllers\HomeController.cs

Adding the [AllowAnonymous]  attribute to the action will give the use the ability to access the page anonymously without Authentication.

[AllowAnonymous]

public ActionResult Index()

{

return View();

}

Create Contact List link

Navigate to ../Views/Shared/\_Layout.cshtml and add the following code to display the WhoAmI link on our web application.

We will only display this link if the user has logged in and has been authenticated.

<ul class="nav navbar-nav">

<li>@Html.ActionLink("Home", "Index", "Home")</li>

<li>@Html.ActionLink("About", "About", "Home")</li>

<li>@Html.ActionLink("Contact", "Contact", "Home")</li>

@if (Request.IsAuthenticated)

{

<li>@Html.ActionLink("WhoAmI", "Index", "CrmContactList")</li>

}

</ul>

Create CrmController

Right click the Controllers folder and select Add > Controller…

Select MVC 5 Controller – Empty

Click Add

Add the following properties to the CrmController

private string clientId = ConfigurationManager.AppSettings["ida:ClientId"];

private string authority = ConfigurationManager.AppSettings["ida:AADInstance"] + "common";

private string aadInstance = ConfigurationManager.AppSettings["ida:AADInstance"];

private string OrganizationHostName = ConfigurationManager.AppSettings["ida:OrganizationHostName"];

private string appKey = ConfigurationManager.AppSettings["ida:ClientSecret"];

Create a new method called GetServiceUrl in the CrmController. This method takes the CRM Organisation name and returns the CRM organisations service URL.

private Uri GetServiceUrl(string organizationName)

{

var organizationUrl = new Uri(

string.Format(OrganizationHostName, organizationName));

return new Uri(organizationUrl +

@"/xrmservices/2011/organization.svc/web?SdkClientVersion=8.2");

}

Add the following code to your Index method

public ActionResult Index()

{

string tenantID = ClaimsPrincipal.Current.FindFirst(

"http://schemas.microsoft.com/identity/claims/tenantid").Value;

// Clean organization name from user logged

string organizationName = User.Identity.Name.Substring(

User.Identity.Name.IndexOf('@') + 1,

User.Identity.Name.IndexOf('.') - (User.Identity.Name.IndexOf('@') + 1));

//string crmResourceId = "https://[orgname].crm.microsoftonline.com";

var resource = string.Format(OrganizationHostName, organizationName);

// Request a token using application credentials

ClientCredential clientcred = new ClientCredential(clientId, appKey);

AuthenticationContext authenticationContext =

new AuthenticationContext(aadInstance + tenantID);

AuthenticationResult authenticationResult =

authenticationContext.AcquireToken(resource, clientcred);

var requestedToken = authenticationResult.AccessToken;

// Create CRM Connection using the token

using (var sdkService = new OrganizationWebProxyClient(

GetServiceUrl(organizationName), false))

{

// Query CRM

sdkService.HeaderToken = requestedToken;

OrganizationRequest request = new OrganizationRequest()

{

RequestName = "WhoAmI"

};

OrganizationResponse response = sdkService.Execute(request);

return View((object)string.Join(",", response.Results.ToList()));

}

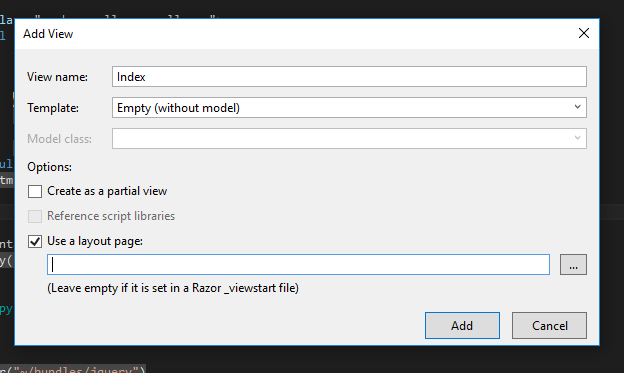
}

If you get an error “AuthenticationContext does not contain AcquireToken” make sure that you have added version 2.xx.x of the Microsoft.IdentityModel.Clients.ActiveDirectory assembly reference to your project

Create Crm View

Right click the Crm folder and select Add > View…

In the Add View dialog, set the following values:



Click Add

Update your view to represent the following code.

@model string

@{

ViewBag.Title = "CRM SDK Contact List";

}

<h2>@ViewBag.Title.</h2>

<p>Connected and executed sdk command.</p>

<p>Value: @Model</p>

Press F5 and test you application.