**VSTS Authentication using oAuth 2.0**

Authenticate your web app's users to access the REST APIs so that your app doesn't have to keep asking for their usernames and passwords. VSTS uses the [OAuth 2.0 protocol](http://oauth.net/2/) to authorize your app for a user and generate an access token. Use this token when you call the REST APIs from your app.

First, you'll register your web app and get an app ID from VSTS. Using that app ID, you'll send your users to VSTS to authorize your app to access their accounts there. Once they've done that, you'll use that authorization to get an access token for that user. When you call VSTS APIs on behalf of that user, you'll use that user's access token. Access tokens expire, so you'll also need to refresh the access token if it's expired.

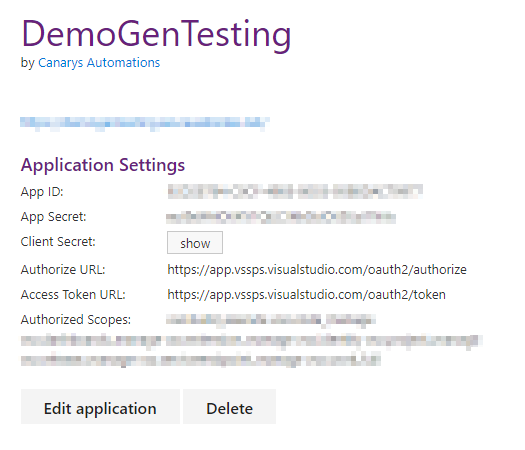
**Register your app**

Go to <https://app.vsaex.visualstudio.com/app/register> to register you app with VSTS.

Make sure you select the [scopes](https://docs.microsoft.com/en-in/vsts/integrate/get-started/authentication/oauth#scopes) that your application needs, and then use the exact same scopes when you [authorize your app](https://docs.microsoft.com/en-in/vsts/integrate/get-started/authentication/oauth#authorize-your-app). If you registered your app using the preview APIs, you'll want to re-register because the scopes that you used are now deprecated.

When VSTS presents the authorization approval page to your user, it will use your company name, and app name and descriptions, along with the URLs for your company's web site, your app's website, and your terms of service and privacy statements, like this.

While registering make sure that you have given the callback URL to which page of your application needs to get Access Token.



Once you registered successfully you will get the App ID, App Secret, Client Secret and Scopes you selected, save those details in your web.config file in the AppSetting section OR Store it as protect member of the class for future use. Using these your application will be authenticated.

<appSettings>

<add key="ClientId" value="YOUR\_CLIENT\_ID" />

<add key="ClientSecret" value="YOUR\_CLIENT\_SECRET" />

<add key="RedirectUri" value="YOUR\_REDIRECT\_URL" />

</appSettings>

**Structure of .Net or .Net Core project**

Once you create Asp.Net or Asp.Net Core application (MVC), think that you have create three view pages from Home Controller. Viz,

1. Sign In
2. Verify
3. Callback

Sign In page is the start page of your application where user will choose the Authentication mode as VSTS Authentication.

public ActionResult SignIn(){

return View();

}

Once user clicks on ***Sign*** button, app will be redirected to second page to verify the scopes you selected against your Client ID, Client Secret and Redirect URL.

URL format:

{authorizeURL}?client\_id={CLIENT\_ID}}&response\_type=Assertion&state=User1&scope={SELECTED\_SCOPES}&redirect\_uri={YOUR\_REDIRECT-URL}

public ActionResult Verify()

{

string url = "https://app.vssps.visualstudio.com/oauth2/authorize?client\_id={0}&response\_type=Assertion&state=User1&scope=vso.build\_execute%20vso.code\_manage%20vso.dashboards\_manage%20vso.extension\_manage%20vso.identity%20vso.project\_manage%20vso.release\_manage%20vso.serviceendpoint\_manage%20vso.work\_full&redirect\_uri={1}";

string redirectUrl = System.Configuration.ConfigurationManager.AppSettings["RedirectUri"];

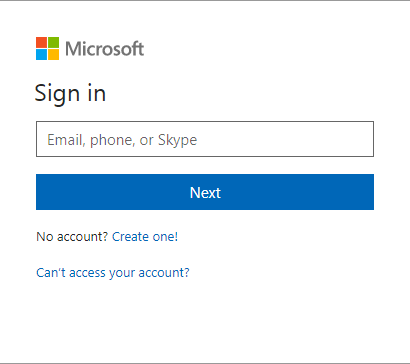
string clientId = System.Configuration.ConfigurationManager.AppSettings["ClientId"];

url = string.Format(url, clientId, redirectUrl);

return Redirect(url);

}

If the user is not logged in to any of the Microsoft Accounts or the Organization Account, this will give Sign in prompt where user must provide his credentials to get authenticated to access the application.



Once user signed in successfully, Microsoft will provide a page where it will show the list of Scope you have selected while registering your application. Based on your scope selection your application can successfully call the APIs with in that scope. User can give the permission by pressing **Accept** button. If user clicks **Deny** button, app should redirect user to Sign in page.

Upon clicking Accept Button Application will be redirected to third page, this is the page where you will get access token and call the REST API for further app transactions.

For more information please go through [Authorize access to REST APIs with OAuth 2.0](https://docs.microsoft.com/en-in/vsts/integrate/get-started/authentication/oauth)

