* Microsoft.IdentityModel.Clients.ActiveDirectory
* WindowsAzure.Storage

using Microsoft.IdentityModel.Clients.ActiveDirectory;

using Microsoft.WindowsAzure.Storage.Auth;

using System.Globalization;

using Microsoft.WindowsAzure.Storage.Blob;

using System.IO;

static void Main(string[] args)

{

Task<AuthenticationResult> t = GetUserOAuthToken();

t.Wait();

string accessToken = t.Result.AccessToken;

Console.WriteLine( "ACCESS TOKEN \n\n" + accessToken);

Console.WriteLine("\n\n Please any key to display content of the blob");

Console.ReadKey();

// Use the access token to create the storage credentials.

TokenCredential tokenCredential = new TokenCredential(accessToken);

StorageCredentials storageCredentials = new StorageCredentials(tokenCredential);

// Create a block blob using those credentials

CloudBlockBlob blob = new CloudBlockBlob(new Uri("https://rudrateststorageaccount.blob.core.windows.net/testcontainer/SalesOrder.json"), storageCredentials);

using (var stream = blob.OpenRead())

{

using (StreamReader reader = new StreamReader(stream))

{

while (!reader.EndOfStream)

{

Console.WriteLine(reader.ReadLine());

}

}

}

Console.WriteLine("\n\n Please any key to terminate the program");

Console.ReadKey();

}

static async Task<AuthenticationResult> GetUserOAuthToken()

{

const string ResourceId = "<https://storage.azure.com/>";

const string AuthInstance = "[https://login.microsoftonline.com/{0}/](https://login.microsoftonline.com/%7b0%7d/)";

const string TenantId = "**<TenantID>**"; // Tenant or directory ID

// Construct the authority string from the Azure AD OAuth endpoint and the tenant ID.

string authority = string.Format(CultureInfo.InvariantCulture, AuthInstance, TenantId);

AuthenticationContext authContext = new AuthenticationContext(authority);

ClientCredential cc = new ClientCredential("**<CLIENT ID>**", "**<CLIENT SECRET>**");

// Acquire an access token from Azure AD.

AuthenticationResult result = await authContext.AcquireTokenAsync(ResourceId, cc);

return result;

}