Operating Azure Data Lake with the .NET SDK

Updated on: 1/31/2017

# Introduction

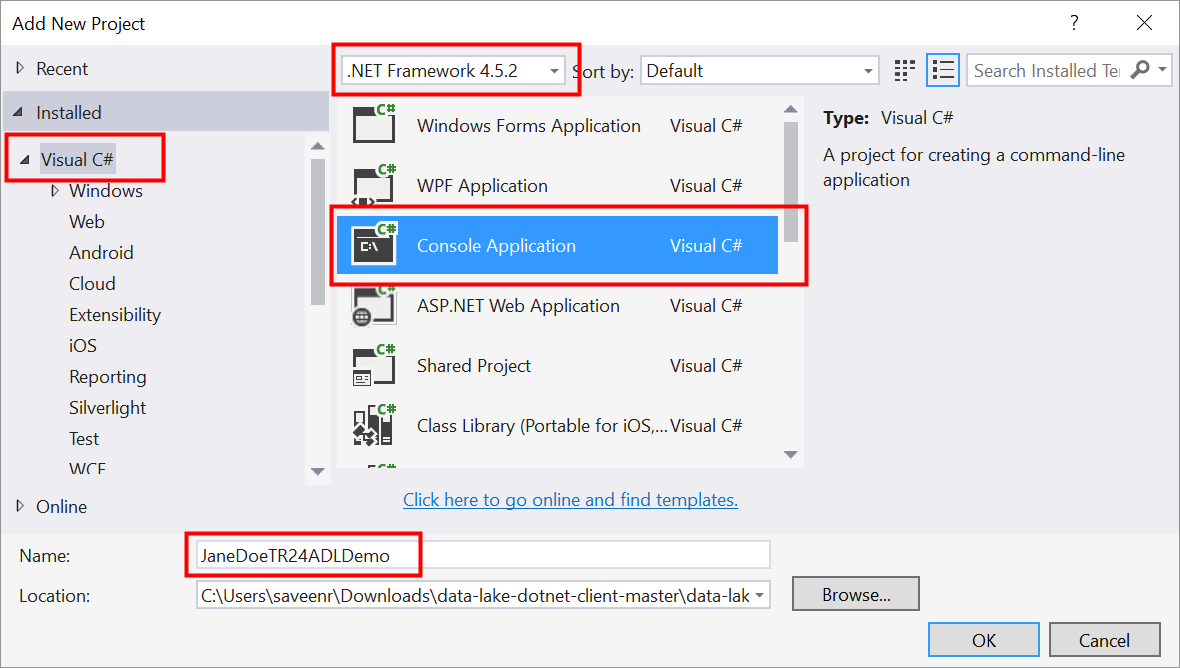
In this lab you will learn how to use the Data Lake .NET SDK to interact with Azure Data Lake in C#.

# Prerequisites

* You Need an ADLA Account
* You Need an ADLS Account

# Exercise 0: Creating the base project

* Launch Visual Studio
* Create a new C# Console Application project
  + Make sure that .NET Framework 4.5.2 is selected
  + Call your project something like **JaneDoeTR24ADLDemo**



* We have published a NuGet package that dramatically simplifies working with Azure Data Lake.
* Right-click on the project and click **Manage NuGet packages**
* From the nuget.org package provider, search for a **preview** package called **AzureDataLake.Client** and add it to your project
* This NuGet package contains all the helper code and also will automatically add the many other NuGet packages you need.
  + You can view the source code for this package here: <https://github.com/Azure-Samples/data-lake-dotnet-client>

# Exercise 1:

* Your Program.cs will look like this:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace JaneDoeTR24ADLDemo

{

class Program

{

static void Main(string[] args)

{

}

}

}

* Now make your code look like this:

using System.Collections.Generic;

using System.Linq;

namespace JaneDoeTR24ADLDemo

{

class Program

{

static void Main(string[] args)

{

string tenant = "YOURTENANT"; // change this to YOUR tenant

string adla\_account = "YOURACCOUNT"; // change this to an ADL Analytics account you have access to

string adls\_account = "YOURACCOUNT"; // change this to an ADL Store account you have access to

var auth\_session = new AzureDataLakeClient.Authentication.AuthenticatedSession("ADL\_Demo\_Client", tenant);

auth\_session.Authenticate();

var job\_client = new AzureDataLakeClient.Analytics.AnalyticsJobClient(adla\_account, auth\_session);

var fs\_client = new AzureDataLakeClient.Store.StoreFileSystemClient(adls\_account, auth\_session);

}

}

}

* Replace the values ass needed for tenant, adla\_account, adls\_account
* All the methods you need are available from job\_client and fs\_client
* Run this code.
  + It won’t print anything.
  + If it runs without failure then it indicates that the authentication works correctly.

# Exercise 2: List jobs and submit a job

In this exercise you will retrieve a list of all the jobs that have run on your ADLA account. You will then submit a new job and check the job status programmatically.

* List all jobs that have run on your ADLA account.
  + Use job\_client.GetJobs
  + Notice that job\_client.GetJobs requires an options object of type AzureDataLakeClient.Analytics.GetJobsOptions. This will prove useful soon.
* Get only the first top 10 jobs
  + Use the Top property of the GetJobsOptions
* Get only the failed jobs
  + Use the Top property of the GetJobsOptions
  + The options object has a filter property, use it to filter to the Result property
* Submit a job
  + Create any U-SQL Scipt you want
  + Use job\_client.SubmitJob
  + Notice that job\_client. SubmitJob requires an options object of type AzureDataLakeClient.Analytics.SubmitJobOptions.
* Get the status of the job that you submitted
* reference the job ID you gave in the previous step
* Use the method \_jobclient.GetJob( )

# Exercise 2: List files and download a file

In this exercise you will retrieve a list of all the files in an output folder. You will also download the output of the job you created in Exercise 1.

List all files in the /Samples/Output/ folder using the **ListFilesPaged** method.

var root = new AzureDataLakeClient.Store.FsPath("/");

var lfo = new AzureDataLakeClient.Store.ListFilesOptions();

foreach (var page in fs\_client.ListFilesPaged(root,lfo))

{

foreach (var fileitemn in page.FileItems)

{

Console.WriteLine("path={0} filename={1}",page.Path,fileitemn.PathSuffix);

}

}

Download the output of the job. Use the fs\_client.**Download** method.

* The file to download is /Samples/Output/UserName/SearchLog\_TestOutput.tsv.