Operating Azure Data Lake with the .NET SDK

Updated on: 1/13/2017

Introduction

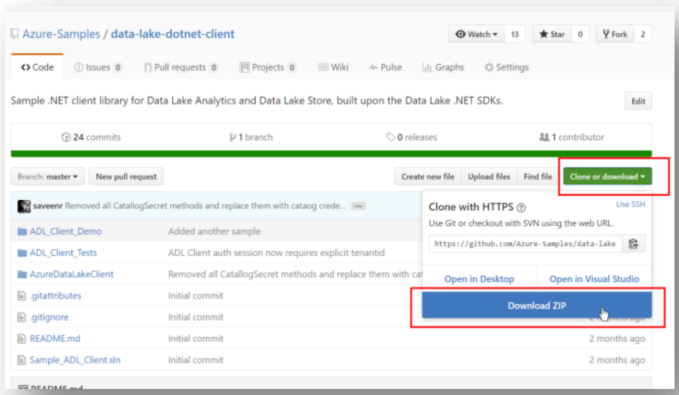
In this lab you will learn how to use the Data Lake .NET SDK to interact with Azure Data Lake in C#.

Prerequisites

- You Need an ADLA Account
- You Need an ADLS Account

Exercise 0: Download the Sample Client

- Open a browser to https://github.com/Azure-Samples/data-lake-dotnet-client
- Download the ZIP file containing the code



Sample .NET client library for Data Lake Analytics and Data Lake Store, built upon the Data Lake .NET SDKs.

saveenr Removed all CatallogSecret methods and replace them with cataog crede...

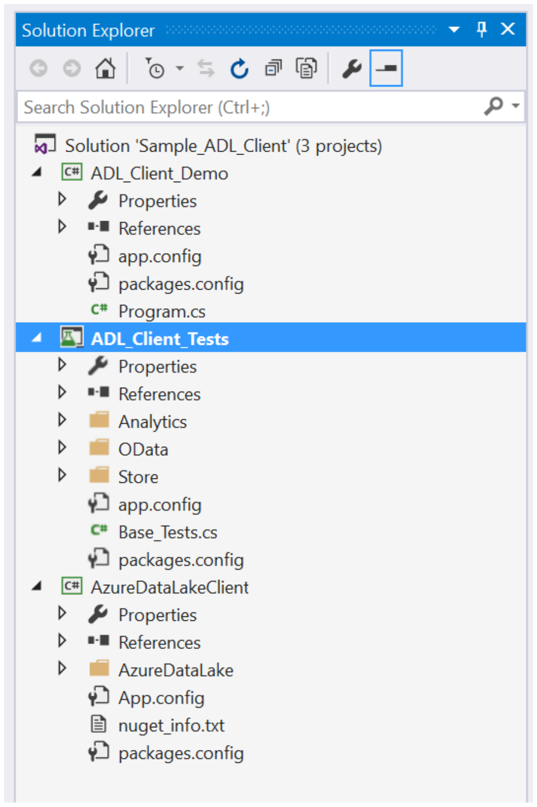
Use Git or checkout with SVN using the web URL.

https://github.com/Azure-Samples/data-lake

ADL_Client_Demo	Added another sample	
ADL_Client_Tests	ADL Client auth session now requires explicit tenantid	
AzureDataLakeClient	Removed all CatallogSecret methods and replace them with cat	
.gitattributes	Initial commit	
.gitignore	Initial commit	
README.md	Initial commit	2 months ago
Sample_ADL_Client.sln	Initial commit	2 months ago

- Extract the contents of the ZIP file
- Launch Visual Studio 2015
- Inside the folder with the extract files Open the solution called "Sample_ADL_Client.sln"

The solution Explorer window should look like this:



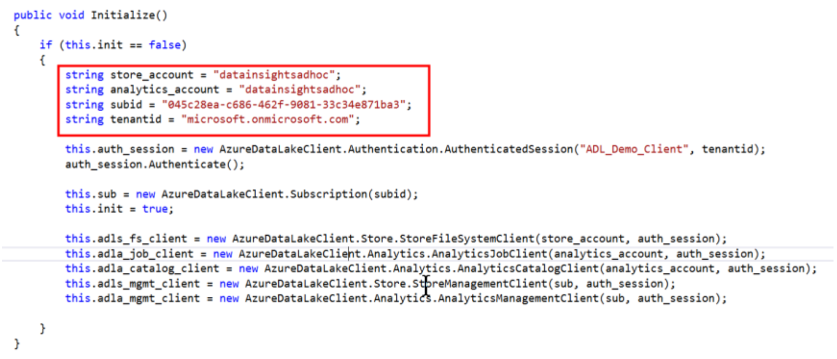
```
Solution 'Sample_ADL_Client' (3 projects)
  ADL_Client_Demo
    Properties
    References
    app.config
    packages.config
    Program.cs
  ADL_Client_Tests
    Properties
    References
    Analytics
    OData
    Store
    app.config
    Base_Tests.cs
    packages.config
  AzureDataLakeClient
    Properties
    References
    AzureDataLake
    App.config
    nuget_info.txt
    packages.config
```

Testing the code

The solution comes with some built-in unit tests. Running these tests helps ensure that you will be able to complete this lab.

IN the ADL_Client_Tests project, open the Base_Tests.cs file

In the Initialize() method, replace the values highlted below:



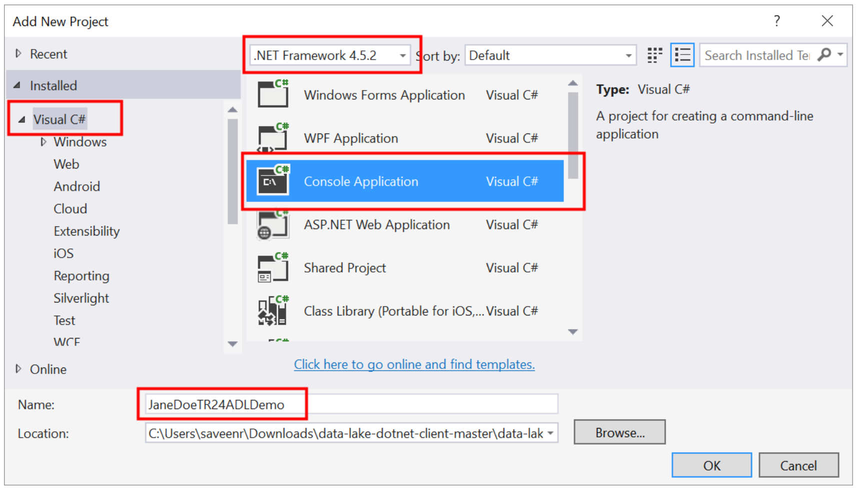
```
public void Initialize()
{
    if (this.init == false)
    {
        string store_account = "datainsightsadhoc";
        string analytics_account = "datainsightsadhoc";
        string subid = "045c28ea-c686-462f-9081-33c34e871ba3";
        string tenantid = "microsoft.onmicrosoft.com";

        this.auth_session = new AzureDataLakeClient.Authentication.AuthenticatedSession("ADL_Demo_Client", tenantid);
        auth_session.Authenticate();

        this.sub = new AzureDataLakeClient.Subscription(subid);
        this.init = true;

        this.adls_fs_client = new AzureDataLakeClient.Store.StoreFileSystemClient(store_account, auth_session);
        this.adla_job_client = new AzureDataLakeClient.Analytics.AnalyticsJobClient(analytics_account, auth_session);
        this.adla_catalog_client = new AzureDataLakeClient.Analytics.AnalyticsCatalogClient(analytics_account, auth_session);
        this.adls_mgmt_client = new AzureDataLakeClient.Store.StoreManagementClient(sub, auth_session);
        this.adla_mgmt_client = new AzureDataLakeClient.Analytics.AnalyticsManagementClient(sub, auth_session);

    }
}
```

Make sure theTest Explorer window is visible Test > Windows > Test Explorer

The click **Tests > Run > All Tests**

You might be prompted to log-in at this time

Exercise 0: Creating the base project

- Go to the Solution Explorer
- Right click on the solution lick > Add > New Project > Visual C# > Console Application
  - Make sure that .NET Framework 4.5.2 is selected
  - Call your project Something like JaneDoeTR24ADLDemo
- 

Add New Project

.NET Framework 4.5.2 — Sort by: Default — Search Installed Templates

Recent
Installed
Visual C#
Windows, Web, Android, Cloud, Extensibility, iOS, Reporting, Silverlight, Test, WCF
Online

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
ASP.NET Web Application	Visual C#
Shared Project	Visual C#
Class Library (Portable for iOS,...	Visual C#

Type: Visual C#
A project for creating a command-line application

Click here to go online and find templates.

Name: JaneDoeTR24ADLDemo
Location: C:\Users\saveenr\Downloads\data-lake-dotnet-client-master\data-lak

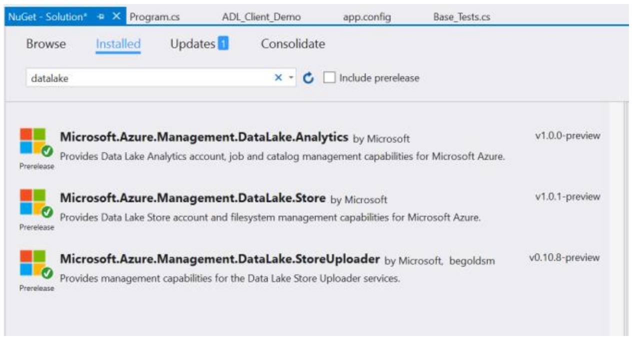
Browse... OK Cancel

Getting the NuGet packages

Right click on the solution and click ManageNuGetPackages for Solution

Make sure you are looking at Installed packages

Filter the package list by “datalake”



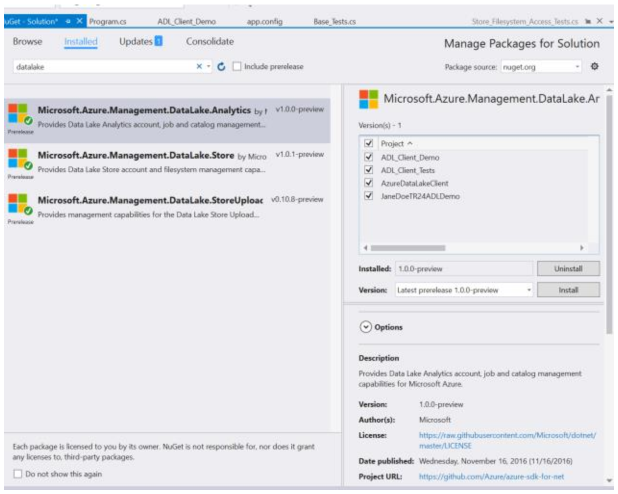
NuGet - Solution* | Program.cs | ADL_Client_Demo | app.config | Base_Tests.cs

Browse | Installed | Updates 1 | Consolidate

datalake — Include prerelease

Microsoft.Azure.Management.DataLake.Analytics by Microsoft Provides Data Lake Analytics account, job and catalog management capabilities for Microsoft Azure.	v1.0.0-preview
Microsoft.Azure.Management.DataLake.Store by Microsoft Provides Data Lake Store account and filesystem management capabilities for Microsoft Azure.	v1.0.1-preview
Microsoft.Azure.Management.DataLake.StoreUploader by Microsoft, begoldsm Provides management capabilities for the Data Lake Store Uploader services.	v0.10.8-preview

Click on each one



datalake

nuget.org

Microsoft.Azure.Management.DataLake.Analytics by I v1.0.0-preview
Provides Data Lake Analytics account, job and catalog management...

Microsoft.Azure.Management.DataLake.Store by Micro v1.0.1-preview
Provides Data Lake Store account and filesystem management capa...

Microsoft.Azure.Management.DataLake.StoreUploac v0.10.8-preview
Provides management capabilities for the Data Lake Store Upload...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Microsoft.Azure.Management.DataLake.Ar

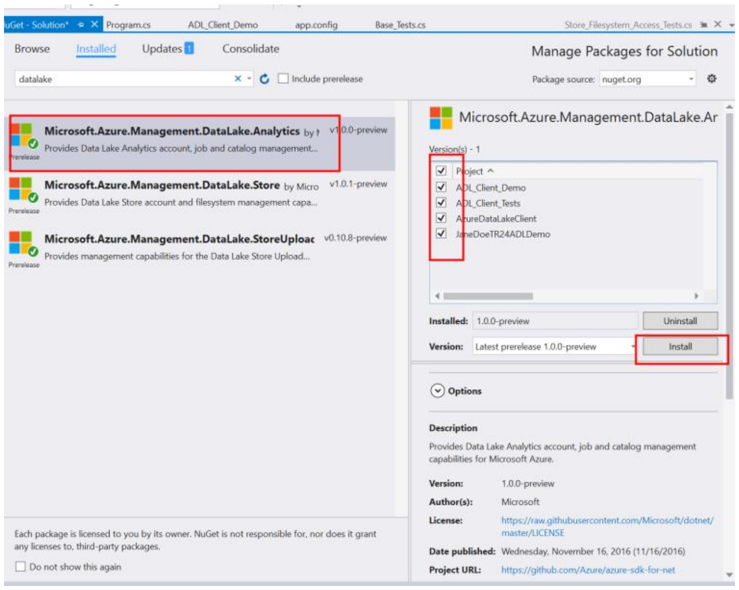
Version(s) - 1

☑ Project
☑ ADL_Client_Demo
☑ ADL_Client_Tests
☑ AzureDataLakeClient
☑ JaneDoeTR24ADLDemo

Installed: 1.0.0-preview
Version: Latest prerelease 1.0.0-preview

Description
Provides Data Lake Analytics account, job and catalog management capabilities for Microsoft Azure.

Version: 1.0.0-preview
Author(s): Microsoft
License: https://raw.githubusercontent.com/Microsoft/dotnet/master/LICENSE
Date published: Wednesday, November 16, 2016 (11/16/2016)
Project URL: https://github.com/Azure/azure-sdk-for-net



datalake

nuget.org

Microsoft.Azure.Management.DataLake.Analytics by I v1.0.0-preview
Provides Data Lake Analytics account, job and catalog management...

Microsoft.Azure.Management.DataLake.Store by Micro v1.0.1-preview
Provides Data Lake Store account and filesystem management capa...

Microsoft.Azure.Management.DataLake.StoreUploac v0.10.8-preview
Provides management capabilities for the Data Lake Store Upload...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

Microsoft.Azure.Management.DataLake.Ar

Version(s) - 1

☑ Project
☑ ADL_Client_Demo
☑ ADL_Client_Tests
☑ AzureDataLakeClient
☑ JaneDoeTR24ADLDemo

Installed: 1.0.0-preview
Version: Latest prerelease 1.0.0-preview

Description
Provides Data Lake Analytics account, job and catalog management capabilities for Microsoft Azure.

Version: 1.0.0-preview
Author(s): Microsoft
License: https://raw.githubusercontent.com/Microsoft/dotnet/master/LICENSE
Date published: Wednesday, November 16, 2016 (11/16/2016)
Project URL: https://github.com/Azure/azure-sdk-for-net

Open Visual Studio and create a new C# Console Application.

```
Install-Package Microsoft.Azure.Management.DataLake.Analytics -Pre
Install-Package Microsoft.Azure.Management.DataLake.Store -Pre
Install-Package Microsoft.Azure.Management.DataLake.StoreUploader -Pre
```

- 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)
        {
        }
    }
}
```

- Add a Reference to AzureDataLakeClient

- 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

Exercise 1: 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 _dataLakeAnalyticsJobClient.Jobs.Get.

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.

- Use the method _dataLakeStoreFileSystemClient.FileSystem.ListFileStatus.

Download the output of the job from Part 2.

- Use the cmdlet _dataLakeStoreFileSystemClient.FileSystem.DirectOpen.
- Use the ADLS path /Samples/Output/UserName/SearchLog_TestOutput.tsv.