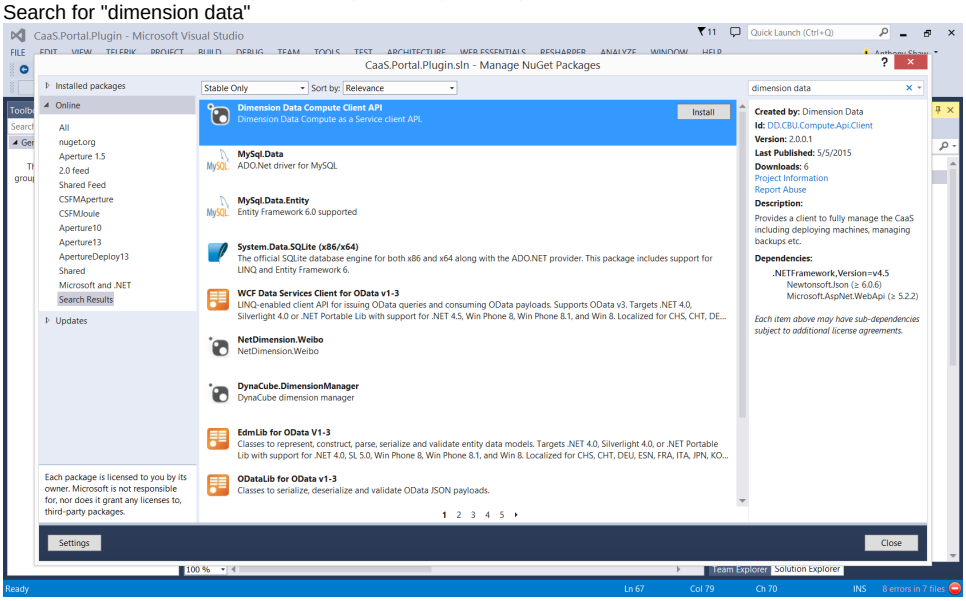
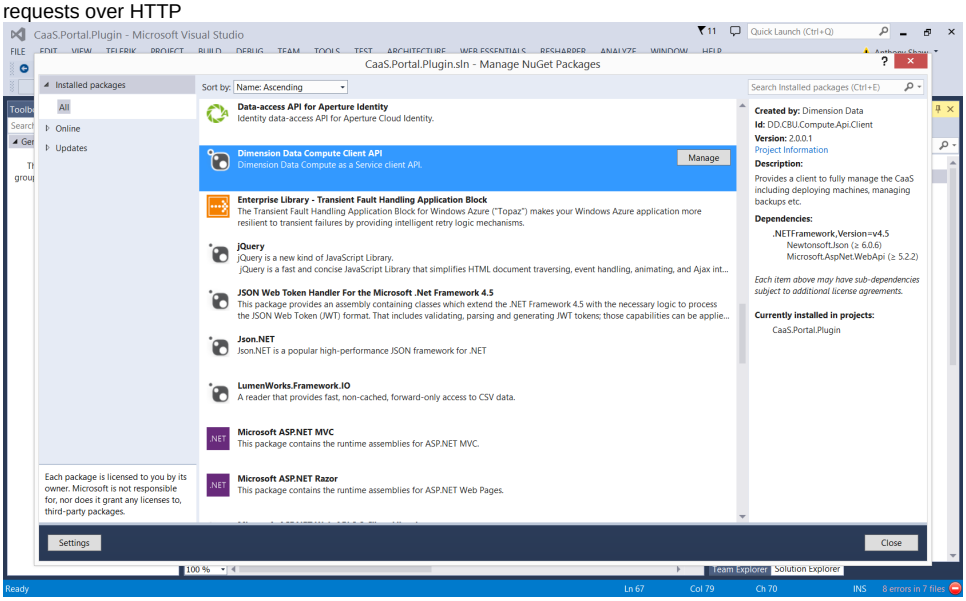
Getting started

Adding the package to a Visual Studio project

1. From within Visual Studio, open the Nuget Package Manager
2. Search for "dimension data"



3. Click Install
4. The package will install its dependencies, the Newtonsoft.Json for serializing JSON and Microsoft's ASP.NET WebApi client for sending XML requests over HTTP



6.

Connecting to the API and returning data

The following example shows an example class using the .NET 4.5 async/await patterns. All of the API call methods within the ComputeApiClient are async.

```
using System.Linq;
using System.Net;
using System.Threading.Tasks;

namespace DD.Cloud.Examples.ApiClient
{
        using DD.CBU.Compute.Api.Client;
        using DD.CBU.Compute.Api.Client.Interfaces;
        using DD.CBU.Compute.Api.Contracts.Directory;

        public class TestClient
        {
                private IComputeApiClient _apiClient;
                private ICredentials _credentials = new NetworkCredential("myuser", "mypassword");
                private KnownApiRegion _targetRegion = KnownApiRegion.Australia_AU;
                private KnownApiVendor _targetVendor = KnownApiVendor.DimensionData;

                public TestClient()
                {
                        _apiClient = ComputeApiClient.GetComputeApiClient(_targetVendor, _targetRegion,
_credentials);
                }

                public async Task<string[]> GetServerNames()
                {
                        // Login
                        IUser myUser = await _apiClient.LoginAsync();
            // Get all of the servers
            var servers = await _apiClient.ServerManagement.Server.GetServersPaginated();
            // Return all of the server names
            return servers.items.Select(server => server.name).ToArray();
                }
        }
}
```

Note : the library use async/await patterns, hence if running in a Console application, the main needs to wait for the tasks to finish before it exists.

Running in a Console Application

```
using System;
using System.Threading;
using System.Threading.Tasks;

namespace DD.Cloud.Examples.ApiClient
{
    static class Program
    {
        static void Main(string[] args)
        {
                        // Console apps don't have a SynchronizationContext by default.
                        // You'll want to create one, or calling .Wait() like this will deadlock.
                        //
                        SynchronizationContext.SetSynchronizationContext(
                                new SynchronizationContext()
                        );

                        MainAsync().Wait();
            Console.ReadKey();
        }
        static async Task<string[]> MainAsync()
        {
            try
            {
                TestClient cli = new TestClient();
                return await cli.GetServerNames();
            }
            catch (Exception ex)
            {
                // Handle Exception
                return new string[] {};
            }
        }
    }
}
```