## 環境準備

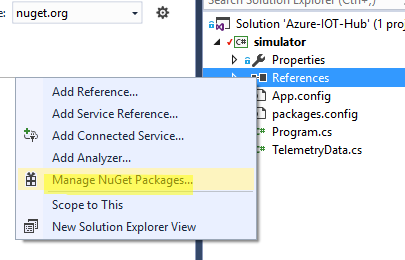
* Visual Studio 2015
* Azure SDK：<http://go.microsoft.com/fwlink/?linkid=518003&clcid=0x404>
* Azure Powershell 1.0.3：<https://www.powershellgallery.com/packages/Azure/1.0.3>
* 已經建立Azure IOT Hub

## 說明

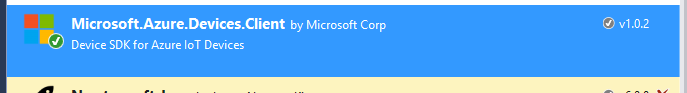
* 在這個Lab我們會建立一個C# Console城市模擬裝置傳遞訊息到IOT Hub
* 實際情況下您的Device應該要透過某些Trigger或是如Lab中展示的方式收集相關資料並傳遞到IOT Hub

## 步驟

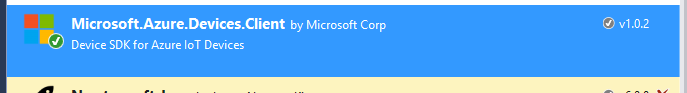
* 開啟Visual Studio
* 建立Console專案
* 加入IOTHub相關Nuget Package



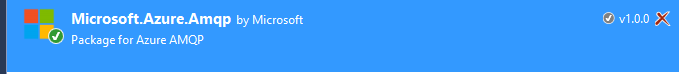
* Microsoft.Azure.Devices.Client



* Microsoft.Azure.Devices



* Microsoft.Azure.Amqp



* 建立新的Class檔案，並命名為TelemetryData.cs
* 加入以下的程式碼

namespace simulator

{

//流水號,timestamp,類別,主機號,UID,DC/AC,ADSL/3G,Msg

public class TelemetryData

{

static Random random = new System.Random();

public string SeqNo { get; set; }

public DateTime Timestamp { get; set; }

public TelemetryTypes Type { get; set; }

public string DeviceId { get; set; }

public string UID { get; set; }

public DCAC DCorAC { get; set; }

public string ADSLor3G { get; set; }

public string Message { get; set; }

public static TelemetryData Random(string seqNo, string msg)

{

var ret = new TelemetryData()

{

SeqNo = seqNo,

Timestamp = DateTime.UtcNow,

Type = (TelemetryTypes)random.Next(0, 2),

DeviceId = Guid.NewGuid().ToString(),

UID = "UID-" + Guid.NewGuid().ToString(),

DCorAC = (DCAC)random.Next(0, 1),

ADSLor3G = random.Next(100) >= 50 ? "ADSL" : "3G",

Message = msg

};

return ret;

}

}

public enum TelemetryTypes

{

A= 0,B= 1,C = 2

}

public enum DCAC

{

DC = 0,AC = 1

}

}

* 回到Program.cs
* 在Program類別下，加入以下變數宣告

static RegistryManager registryManager;

static string connectionString = "{IOT Hub Connection String}";

static string iotHubUri = "{IOT Hub Domain Name}";

static string deviceId = null;

static int maxTemperature = 0;

static int minTemperature = 1;

static string deviceKey = null;

static DeviceClient deviceClient = null;

static Random random = new Random();

* 加入以下的Method已建立模擬訊息

static string GenerateMessage(int seq, string message)

{

var msg = TelemetryData.Random(string.Format("{0}{1}",DateTime.UtcNow.ToString("yyyymmdd"),seq.ToString("0000000")),message);

return JsonConvert.SerializeObject(msg);

}

* 加入以下的程式碼建立Helper Method

static void Error(string msg)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine(msg);

Console.ResetColor();

}

static void Success(string msg)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine(msg);

Console.ResetColor();

}

static void Log(string msg)

{

Console.ResetColor();

Console.WriteLine(msg);

}

static void Wait(string msg = null)

{

if (string.IsNullOrEmpty(msg))

{

Log("Press [ENTER] to continue...");

}

else

{

Log(msg);

}

Console.ReadLine();

}

* 在Main()中加入以下的程式碼；這裡會先檢查傳入參數是否小於三個，並初始化參數；接著會透過ConnectionString建立RegistryManager物件，後續會以此物件註冊Device。

if (args.Length < 3)

{

Error("Usage: sumulator {deviceid} {min} {max}");

Wait();

return;

}

deviceId = args[0];

minTemperature = int.Parse(args[1]);

maxTemperature = int.Parse(args[2]);

registryManager = RegistryManager.CreateFromConnectionString(connectionString);

* 接著，註冊Device

private async static Task AddDeviceAsync()

{

Device device;

try

{

device = await registryManager.AddDeviceAsync(new Device(deviceId));

}

catch (DeviceAlreadyExistsException)

{

device = await registryManager.GetDeviceAsync(deviceId);

}

deviceKey = device.Authentication.SymmetricKey.PrimaryKey;

Log($"device id {deviceId} : {deviceKey}");

}

* 註冊成功後，會得到一組Device Key；IOT Hub會以此Device Key建立各Device獨立的連線字串
* 加入Unregister的method；RemoveDeviceAsync()會將Device從IOT Hub的註冊資料中移除，移除之後該Device就無法再連線到IOTHub

private async static Task RemoveDeviceAsync()

{

var device = await registryManager.GetDeviceAsync(deviceId);

await registryManager.RemoveDeviceAsync(device);

}

* 接著實作Device-to-Cloud訊息

private static async void SendDeviceToCloudMessagesAsync()

{

int i = 0;

while (true)

{

i++;

string telemetry = GenerateMessage(i, $"message:{i}");

await deviceClient.SendEventAsync(new Microsoft.Azure.Devices.Client.Message(Encoding.UTF8.GetBytes(telemetry)));

Console.WriteLine("{0} > Sending message: {1}", DateTime.Now, telemetry);

Thread.Sleep(3000);

}

}

* 實作接收Cloud-to-Device訊息loop

private async static void ReceiveCommandAsync()

{

while (true)

{

var cmd = await deviceClient.ReceiveAsync();

if (cmd != null)

{

Success(Encoding.UTF8.GetString(cmd.GetBytes()));

}

Thread.Sleep(1000);

}

}

* 將以下程式碼加入Main()

AddDeviceAsync().Wait();

//deviceKey = "OTUJwGXWV6mweq/CUSlqaEnackTI6SYXBYM3U75HbKg=";

#if false

//AMQP (default)

deviceClient = DeviceClient.Create(iotHubUri, new DeviceAuthenticationWithRegistrySymmetricKey(deviceId, deviceKey));

#else

//HTTPS

deviceClient = DeviceClient.Create(iotHubUri, new DeviceAuthenticationWithRegistrySymmetricKey(deviceId, deviceKey),

Microsoft.Azure.Devices.Client.TransportType.Http1);

#endif

SendDeviceToCloudMessagesAsync();

ReceiveCommandAsync();

Wait("Press [ENTER] to exit...");

RemoveDeviceAsync().Wait();

* 完整的Main()看起來應該要像這樣子

static void Main(string[] args)

{

#region added

if (args.Length < 3)

{

Error("Usage: sumulator {deviceid} {min} {max}");

Wait();

return;

}

deviceId = args[0];

minTemperature = int.Parse(args[1]);

maxTemperature = int.Parse(args[2]);

registryManager = RegistryManager.CreateFromConnectionString(connectionString);

#endregion

AddDeviceAsync().Wait();

//deviceKey = "OTUJwGXWV6mweq/CUSlqaEnackTI6SYXBYM3U75HbKg=";

#if false

//AMQP (default)

deviceClient = DeviceClient.Create(iotHubUri, new DeviceAuthenticationWithRegistrySymmetricKey(deviceId, deviceKey));

#else

//HTTPS

deviceClient = DeviceClient.Create(iotHubUri, new DeviceAuthenticationWithRegistrySymmetricKey(deviceId, deviceKey),

Microsoft.Azure.Devices.Client.TransportType.Http1);

#endif

SendDeviceToCloudMessagesAsync();

ReceiveCommandAsync();

Wait("Press [ENTER] to exit...");

RemoveDeviceAsync().Wait();

}