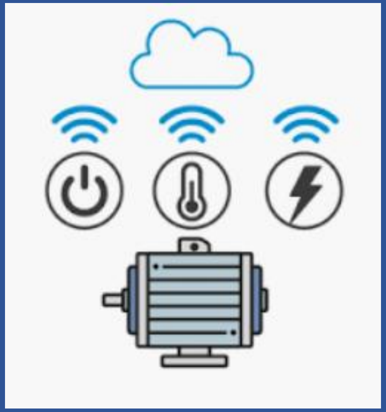
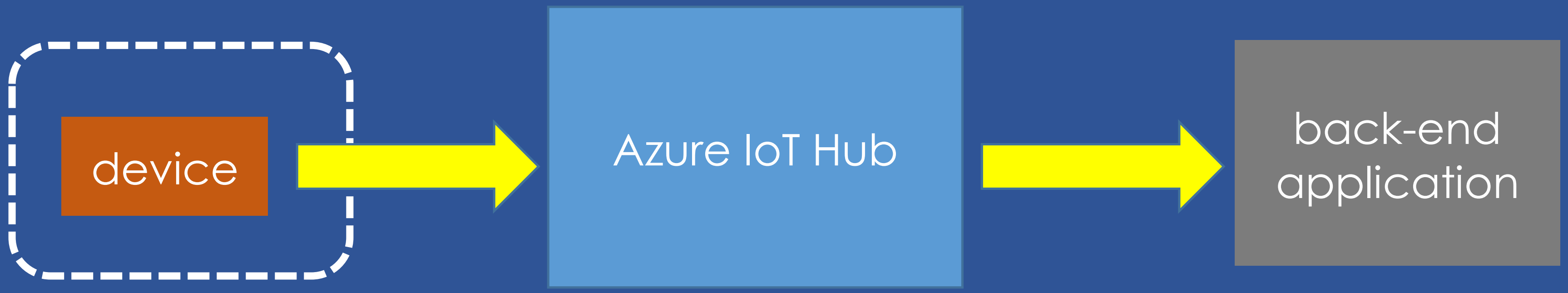
Device to Cloud

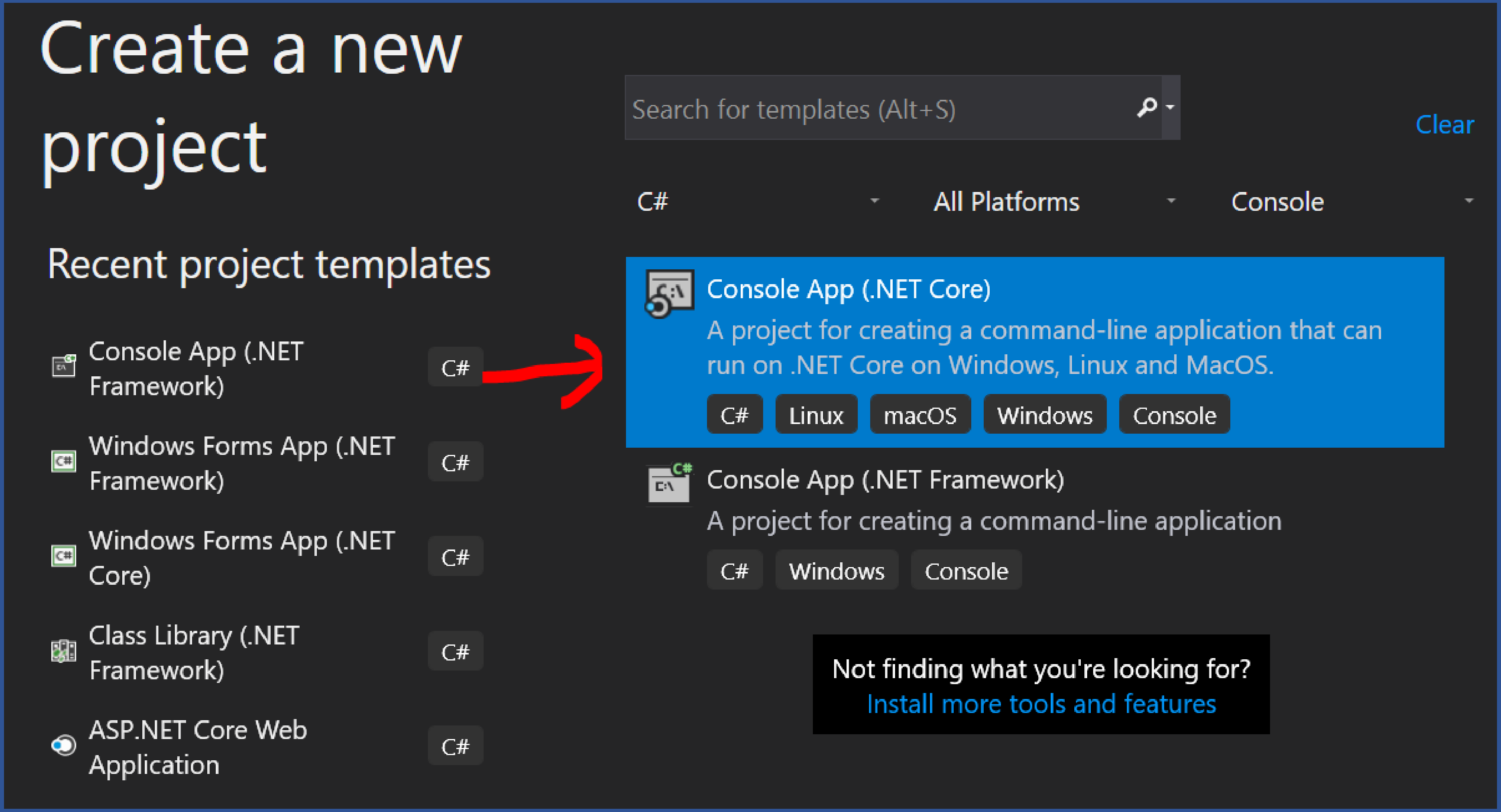


What to do?

Create Device Simulator to send telemetry data to Azure IoT Hub



Open Visual Studio / Create C# Console App .NET Core / Name = c2d



Create a new project

Search for templates (Alt+S)

Clear

C# | All Platforms | Console

Recent project templates

- Console App (.NET Framework) C#
- Windows Forms App (.NET Framework) C#
- Windows Forms App (.NET Core) C#
- Class Library (.NET Framework) C#
- ASP.NET Core Web Application C#

Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C# Linux macOS Windows Console

Console App (.NET Framework)

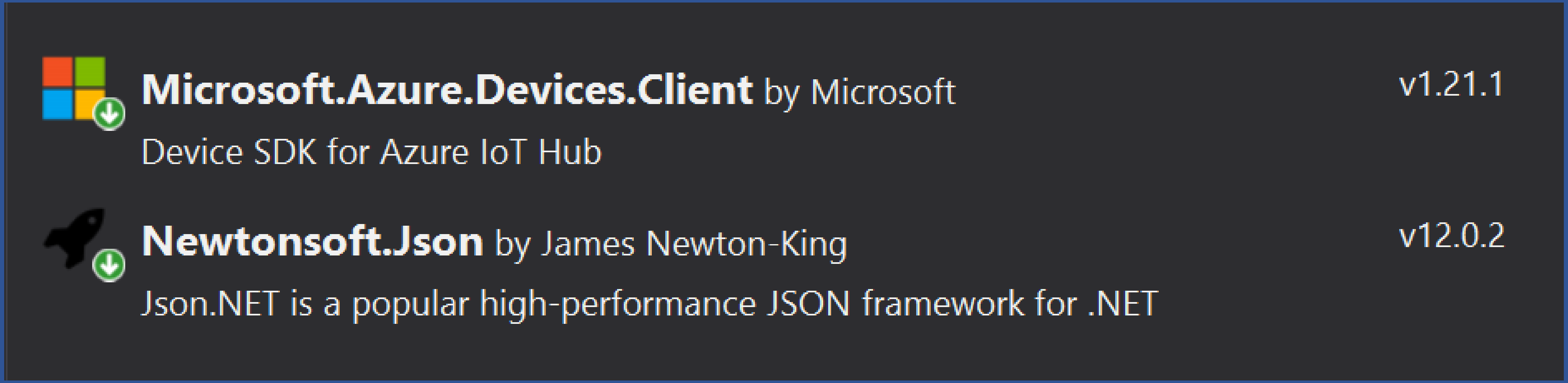
A project for creating a command-line application

C# Windows Console

Not finding what you're looking for?

Install more tools and features

NuGet 2 packages



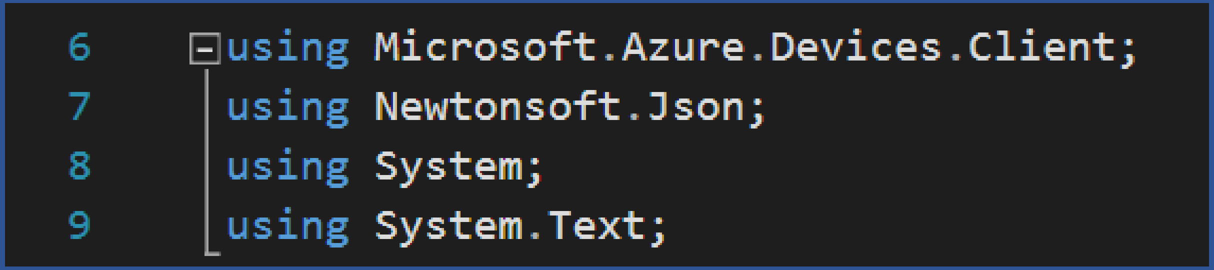
Microsoft.Azure.Devices.Client by Microsoft v1.21.1

Device SDK for Azure IoT Hub

Newtonsoft.Json by James Newton-King v12.0.2

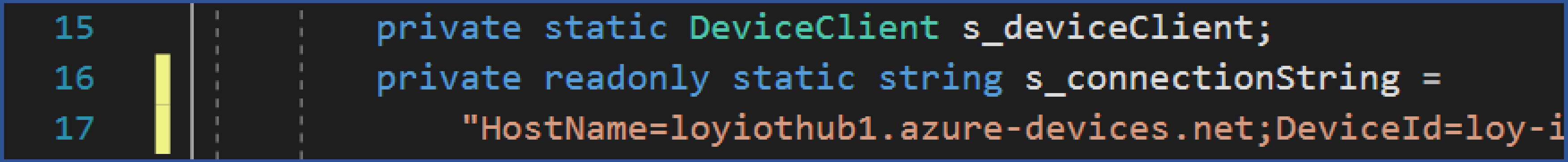
Json.NET is a popular high-performance JSON framework for .NET

Add name space



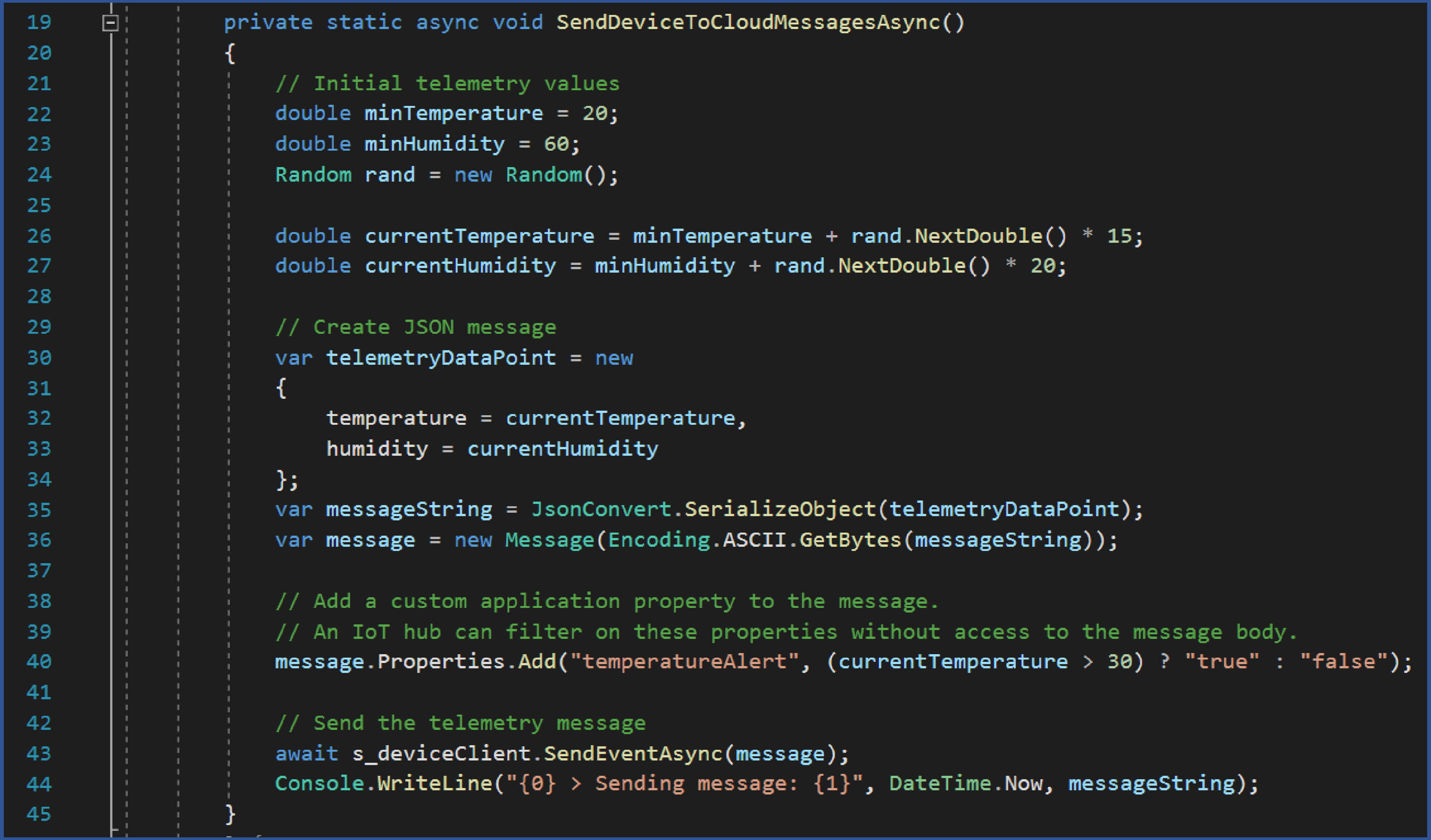
```
using Microsoft.Azure.Devices.Client;
using Newtonsoft.Json;
using System;
using System.Text;
```

Add 2 Class fields



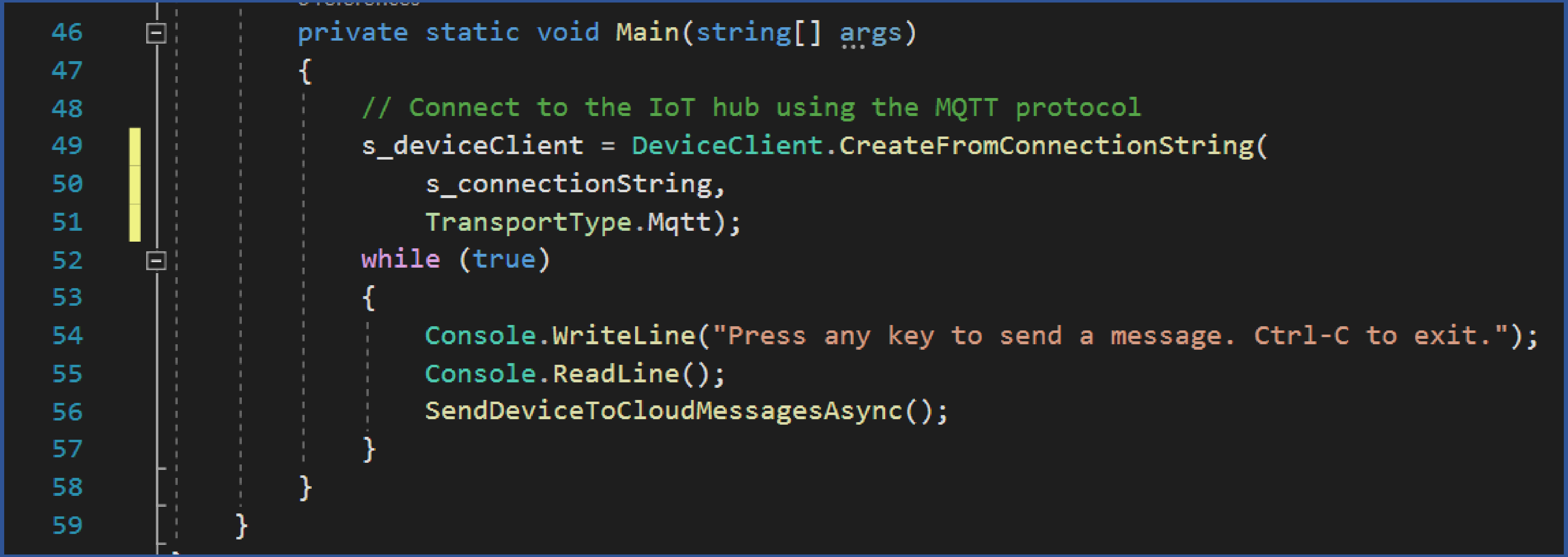
```
        private static DeviceClient s_deviceClient;
        private readonly static string s_connectionString =
            "HostName=loyiothub1.azure-devices.net;DeviceId=loy-i
```

Add method SendDeviceToCloudMessagesAsync()



```
private static async void SendDeviceToCloudMessagesAsync()
{
    // Initial telemetry values
    double minTemperature = 20;
    double minHumidity = 60;
    Random rand = new Random();

    double currentTemperature = minTemperature + rand.NextDouble() * 15;
    double currentHumidity = minHumidity + rand.NextDouble() * 20;

    // Create JSON message
    var telemetryDataPoint = new
    {
        temperature = currentTemperature,
        humidity = currentHumidity
    };
    var messageString = JsonConvert.SerializeObject(telemetryDataPoint);
    var message = new Message(Encoding.ASCII.GetBytes(messageString));

    // Add a custom application property to the message.
    // An IoT hub can filter on these properties without access to the message body.
    message.Properties.Add("temperatureAlert", (currentTemperature > 30) ? "true" : "false");

    // Send the telemetry message
    await s_deviceClient.SendEventAsync(message);
    Console.WriteLine("{0} > Sending message: {1}", DateTime.Now, messageString);
}
```

Add code to Main

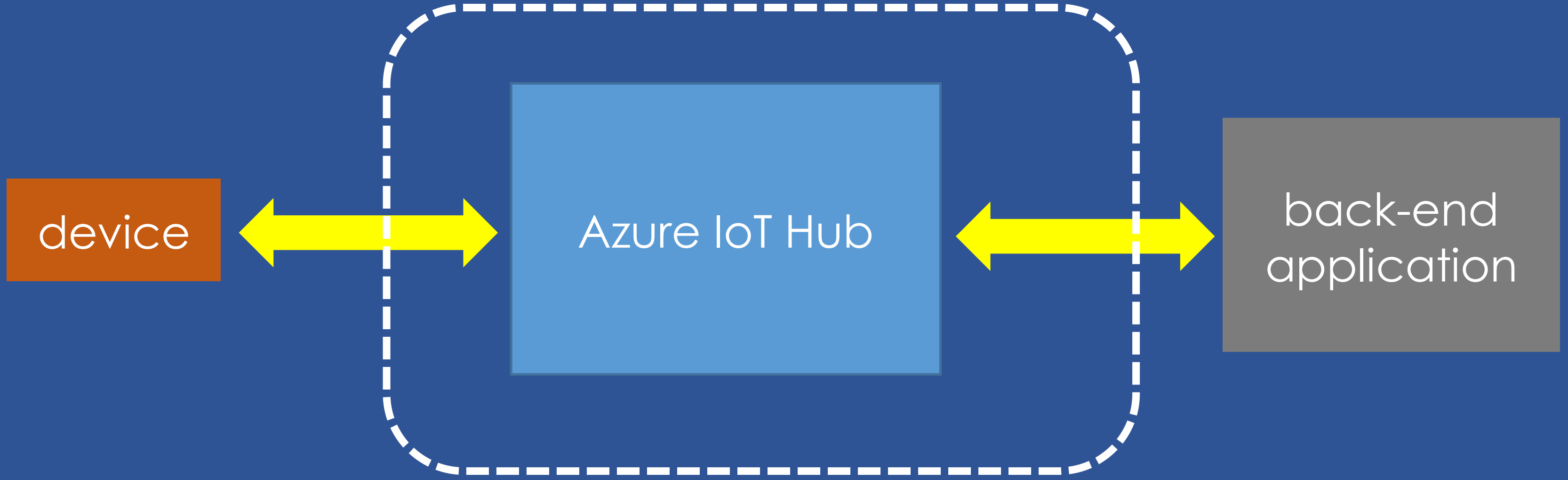


```
        private static void Main(string[] args)
        {
            // Connect to the IoT hub using the MQTT protocol
            s_deviceClient = DeviceClient.CreateFromConnectionString(
                s_connectionString,
                TransportType.Mqtt);
            while (true)
            {
                Console.WriteLine("Press any key to send a message. Ctrl-C to exit.");
                Console.ReadLine();
                SendDeviceToCloudMessagesAsync();
            }
        }
    }
```

Azure IoT Hub Device Explorer

The device explorer is a tool uses the Azure IoT service libraries to perform various functions on IoT Hub.

- Create a new device
- Delete a device
- Monitor data from device
- Send back-end message to device

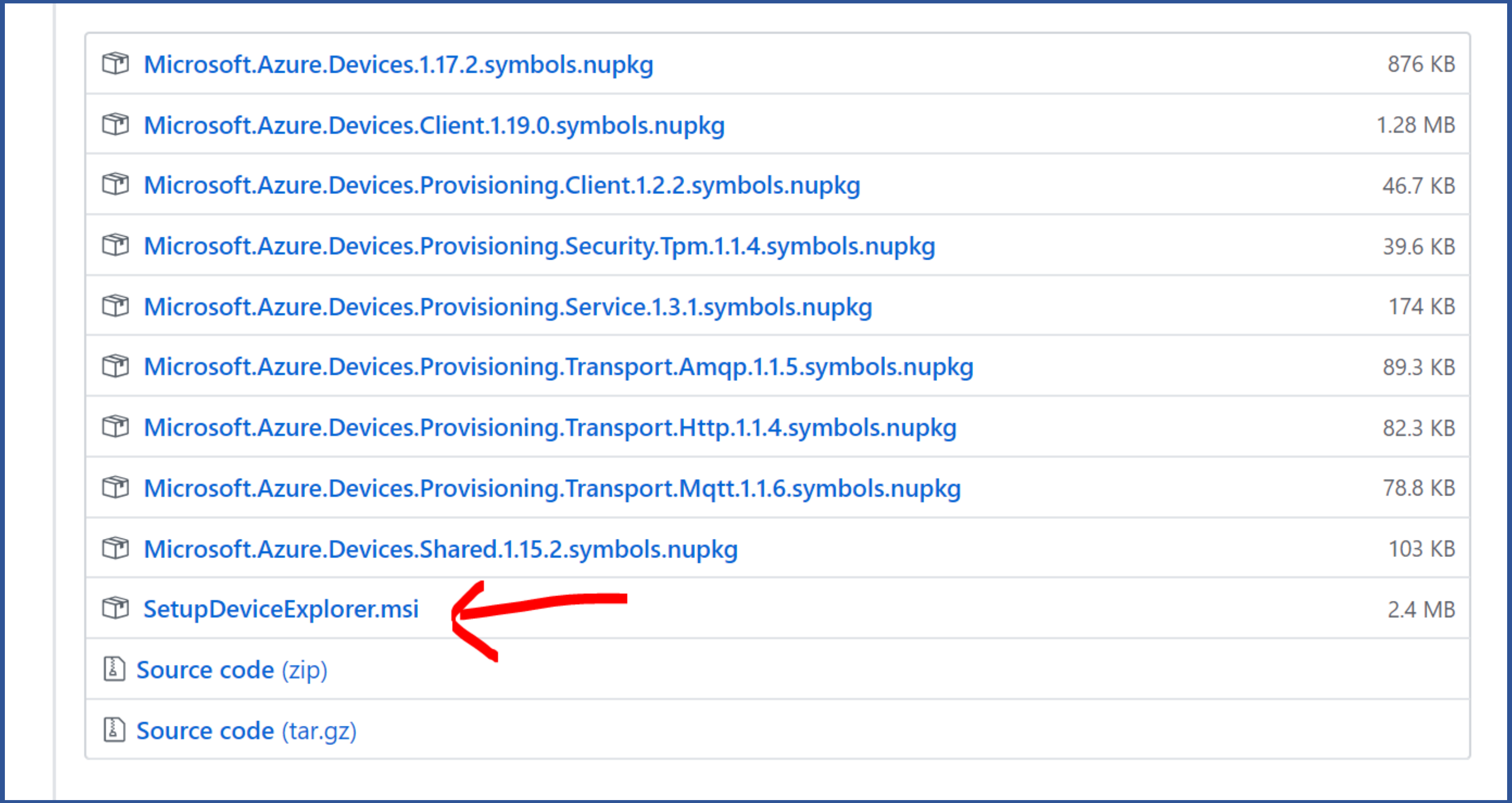


Download and setup Device Explorer

To view telemetry data you need a back-end application

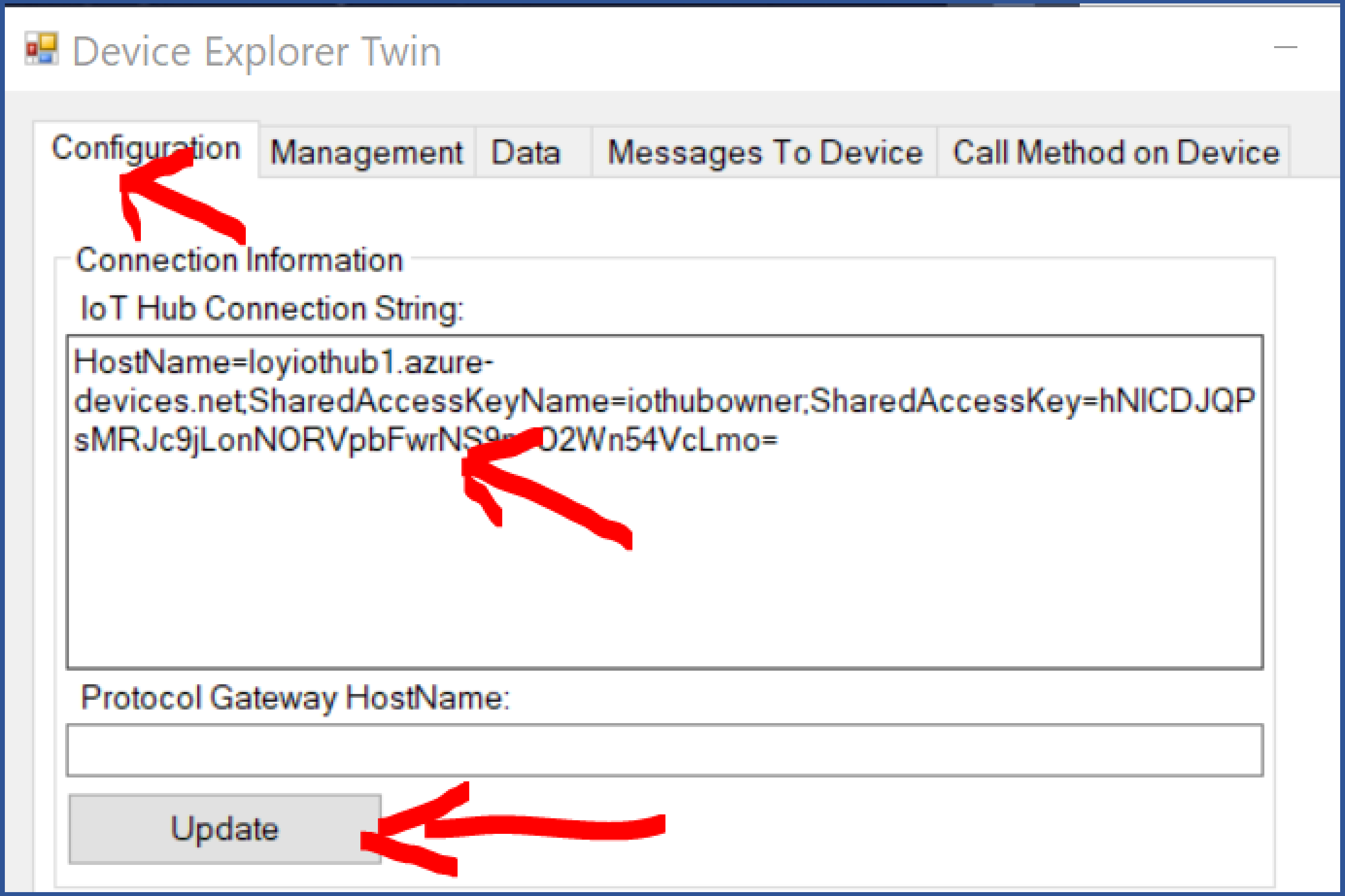
Download link

https://github.com/Azure/azure-iot-sdk-csharp/releases/tag/2019-1-4



Microsoft.Azure.Devices.1.17.2.symbols.nupkg	876 KB
Microsoft.Azure.Devices.Client.1.19.0.symbols.nupkg	1.28 MB
Microsoft.Azure.Devices.Provisioning.Client.1.2.2.symbols.nupkg	46.7 KB
Microsoft.Azure.Devices.Provisioning.Security.Tpm.1.1.4.symbols.nupkg	39.6 KB
Microsoft.Azure.Devices.Provisioning.Service.1.3.1.symbols.nupkg	174 KB
Microsoft.Azure.Devices.Provisioning.Transport.Amqp.1.1.5.symbols.nupkg	89.3 KB
Microsoft.Azure.Devices.Provisioning.Transport.Http.1.1.4.symbols.nupkg	82.3 KB
Microsoft.Azure.Devices.Provisioning.Transport.Mqtt.1.1.6.symbols.nupkg	78.8 KB
Microsoft.Azure.Devices.Shared.1.15.2.symbols.nupkg	103 KB
SetupDeviceExplorer.msi	2.4 MB
Source code (zip)	
Source code (tar.gz)	

Configuration using IoT Hub Connection String

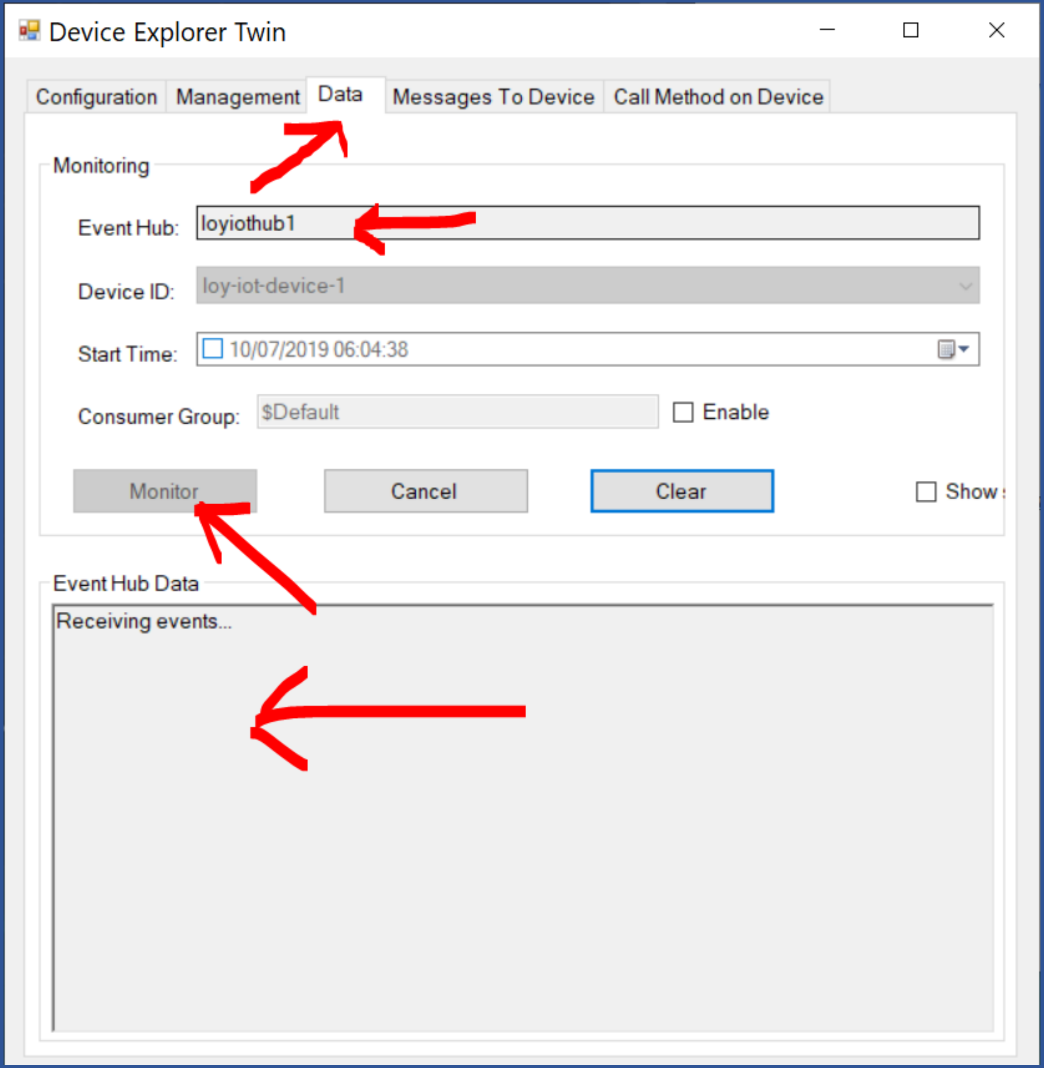


IoT Hub Connection String:

```
HostName=loyiothub1.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=hNICDJQP
sMRJc9jLonNORVpbFwrNS9r[illegible]D2Wn54VcLmo=
```

Protocol Gateway HostName:

Send Device to Could message and monitor using Device Explorer



Monitoring

Event Hub: loyiothub1

Device ID: loy-iot-device-1

Start Time: □ 10/07/2019 06:04:38

Consumer Group: $Default □ Enable

□ Show

Event Hub Data

Receiving events...

What's next?