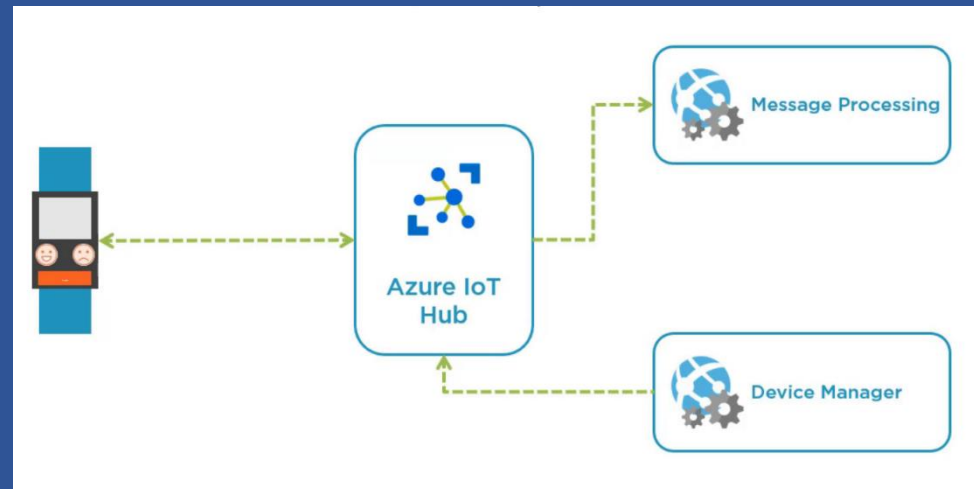
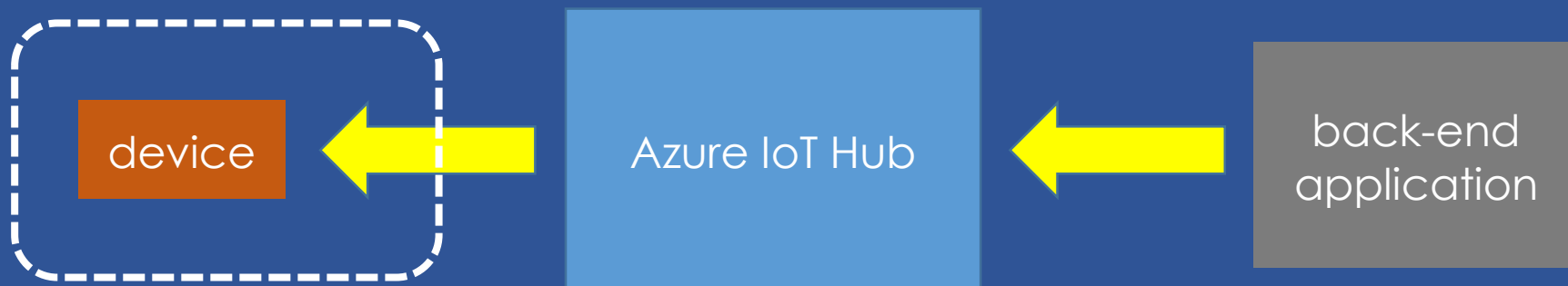


Send Cloud to Device message




What to do?

Create Device Simulator that wait for message from Azure IoT Hub - back-end application










Open Visual Studio / Create C# Console App .NET Core / Name = d2c


Create a new project

Search for templates (Alt+S)  [Clear](#)

C# All Platforms Console

Recent project templates

 Console App (.NET Framework)	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
 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 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

```
6  using Microsoft.Azure.Devices.Client;  
7  using Newtonsoft.Json;  
8  using System;  
9  using System.Text;
```

Add 2 Class fields

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

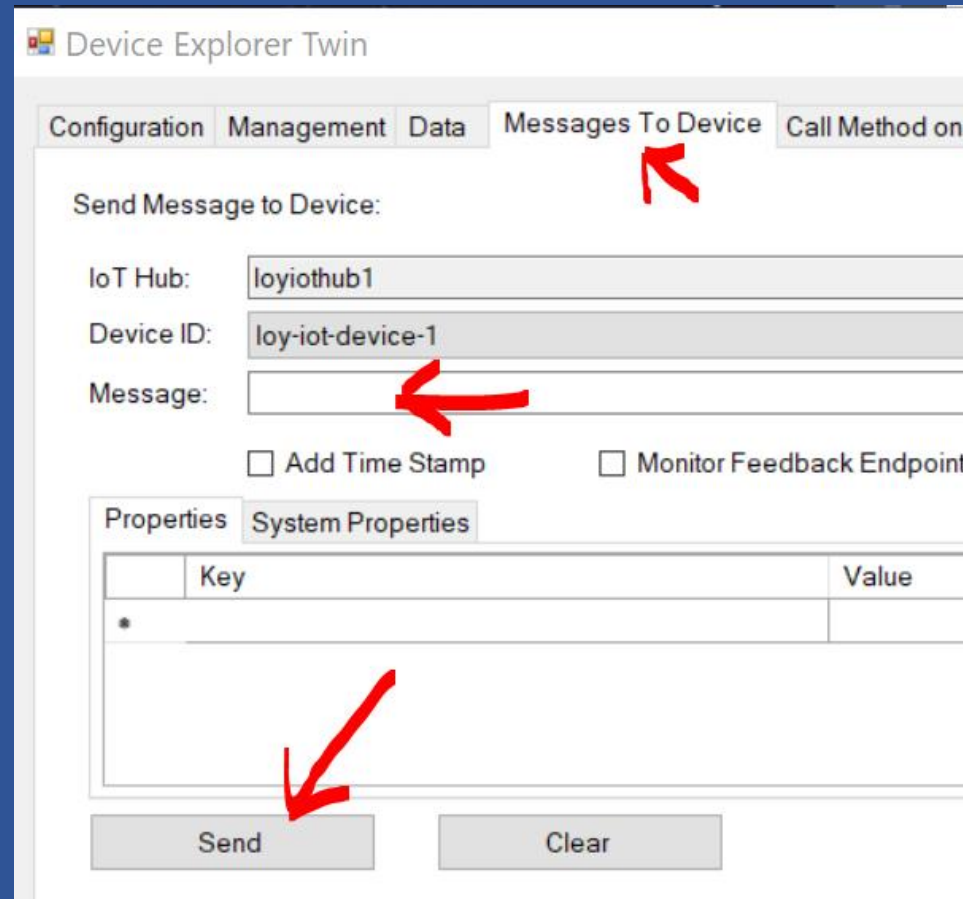
Add method ReciveC2dAsync

```
17 private static async void ReceiveC2dAsync()
18 {
19     Console.WriteLine("\nReceiving cloud to device messages from service");
20     while (true)
21     {
22         Message receivedMessage = await s_deviceClient.ReceiveAsync();
23         if (receivedMessage == null) continue;
24
25         Console.ForegroundColor = ConsoleColor.Yellow;
26         Console.WriteLine("Received message: {0}",
27             Encoding.ASCII.GetString(receivedMessage.GetBytes()));
28         Console.ResetColor();
29
30         // Notifies IoT Hub that the message has been successfully processed
31         await s_deviceClient.CompleteAsync(receivedMessage);
32     }
33 }
```

Add code to Main

```
35 private static void Main(string[] args)
36 {
37     // Connect to the IoT hub using the MQTT protocol
38     s_deviceClient = DeviceClient.CreateFromConnectionString(
39         s_connectionString,
40         TransportType.Mqtt);
41     ReceiveC2dAsync();
42     Console.ReadLine();
43 }
```

1. Run program to wait for message from cloud
2. Device Explorer / Messages to device / Send



What's next?

