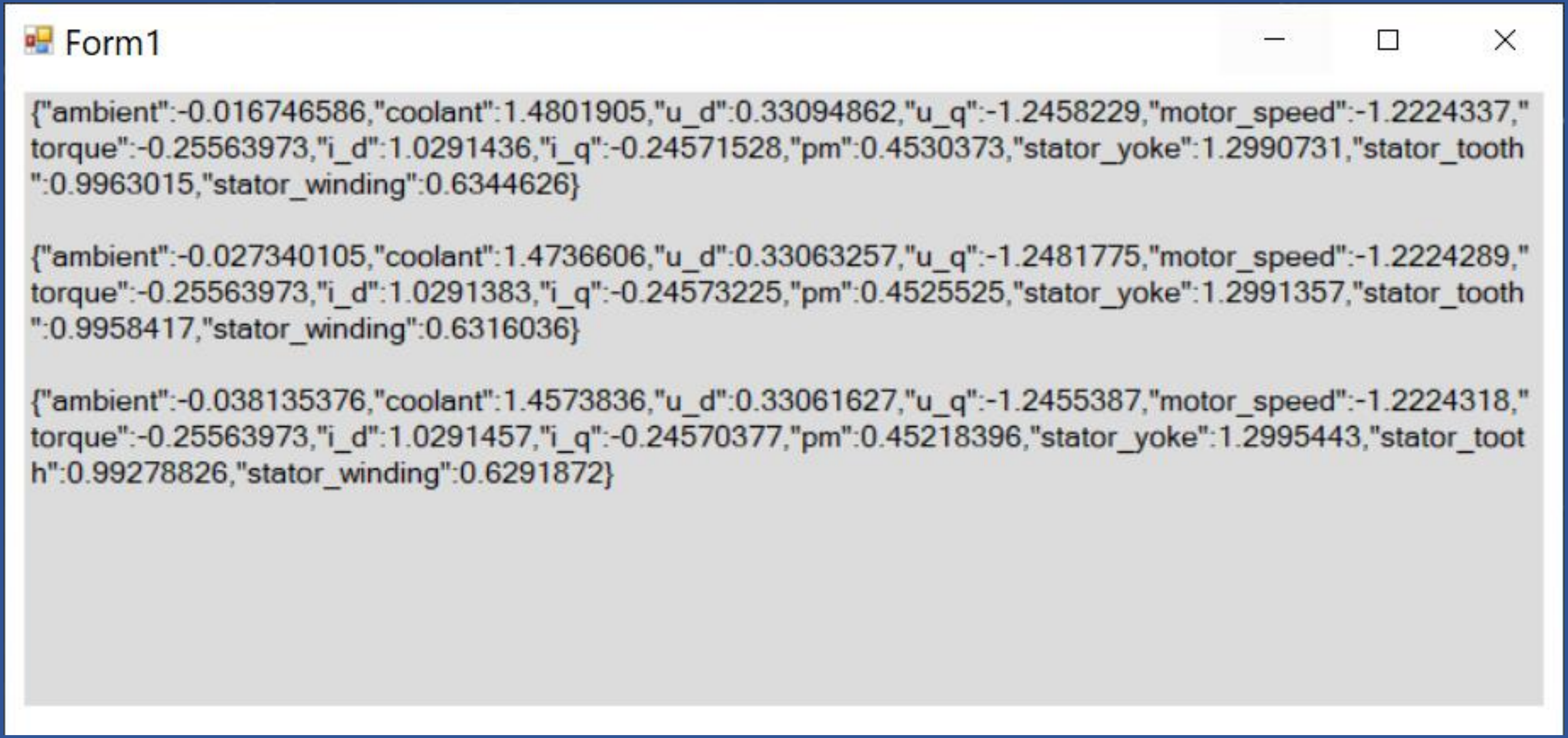
WinForm Backend D2C

Create WinForm show D2C message

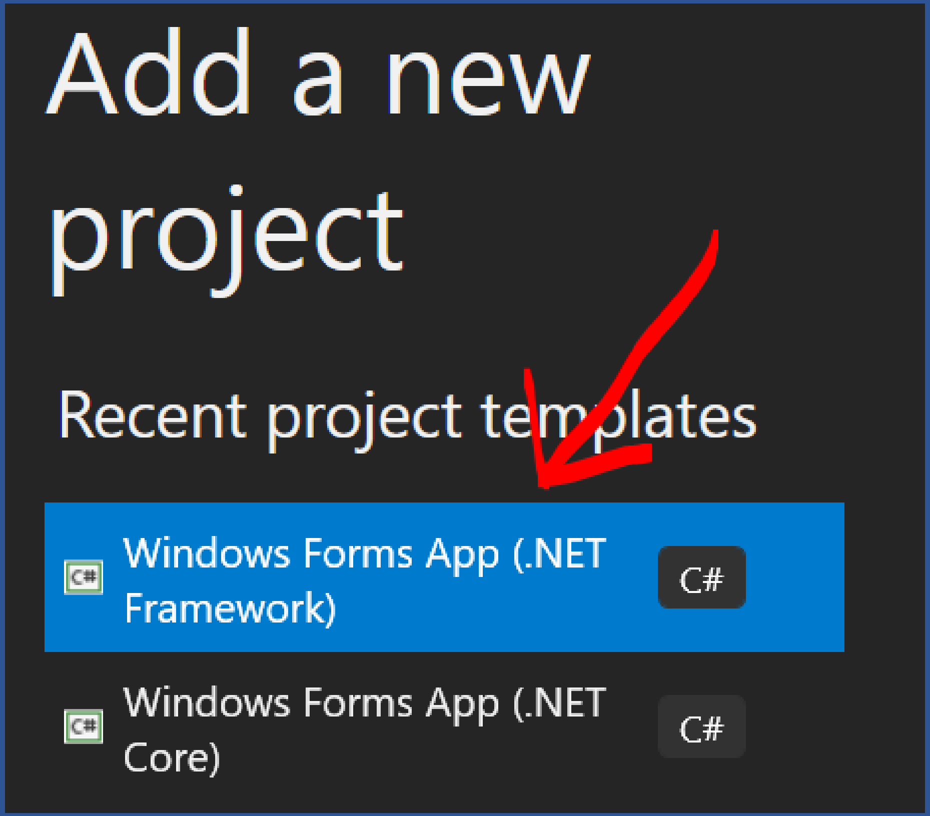


```
{"ambient":-0.016746586,"coolant":1.4801905,"u_d":0.33094862,"u_q":-1.2458229,"motor_speed":-1.2224337,"
torque":-0.25563973,"i_d":1.0291436,"i_q":-0.24571528,"pm":0.4530373,"stator_yoke":1.2990731,"stator_tooth
":0.9963015,"stator_winding":0.6344626}

{"ambient":-0.027340105,"coolant":1.4736606,"u_d":0.33063257,"u_q":-1.2481775,"motor_speed":-1.2224289,"
torque":-0.25563973,"i_d":1.0291383,"i_q":-0.24573225,"pm":0.4525525,"stator_yoke":1.2991357,"stator_tooth
":0.9958417,"stator_winding":0.6316036}

{"ambient":-0.038135376,"coolant":1.4573836,"u_d":0.33061627,"u_q":-1.2455387,"motor_speed":-1.2224318,"
torque":-0.25563973,"i_d":1.0291457,"i_q":-0.24570377,"pm":0.45218396,"stator_yoke":1.2995443,"stator_toot
h":0.99278826,"stator_winding":0.6291872}
```

Open Visual Studio / Create C# .NET Framework WinForm



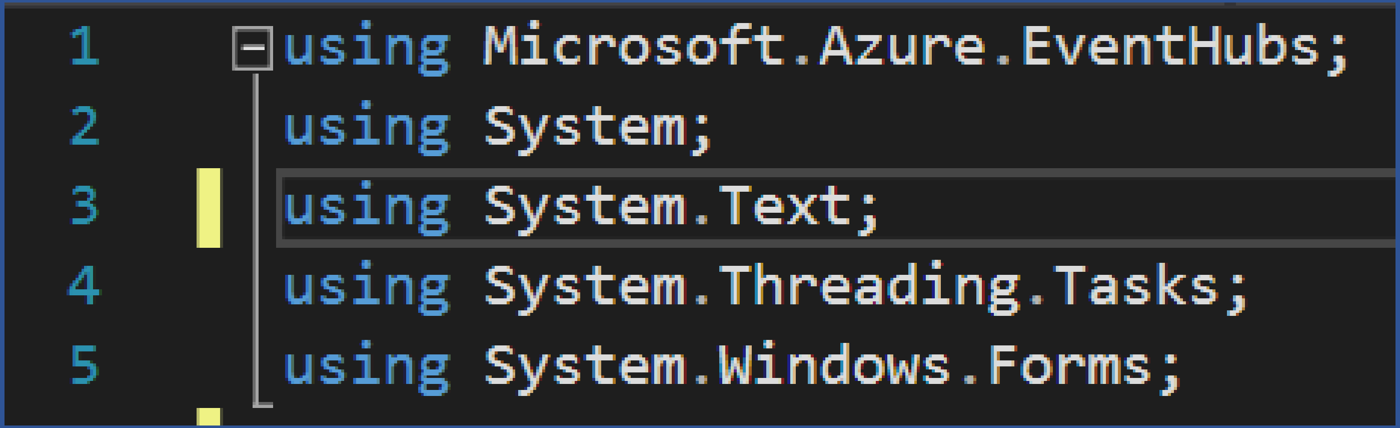
Add a new project

Recent project templates

Windows Forms App (.NET Framework) C#

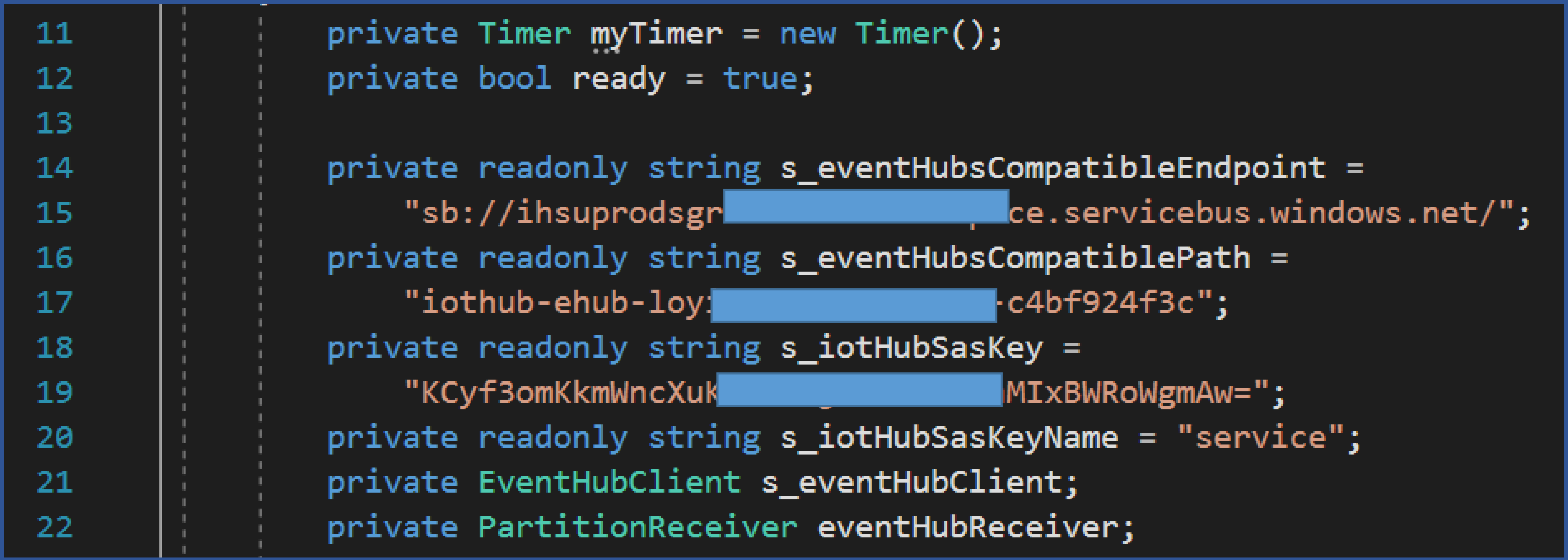
Windows Forms App (.NET Core) C#

Add Namespace



```
using Microsoft.Azure.EventHubs;
using System;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

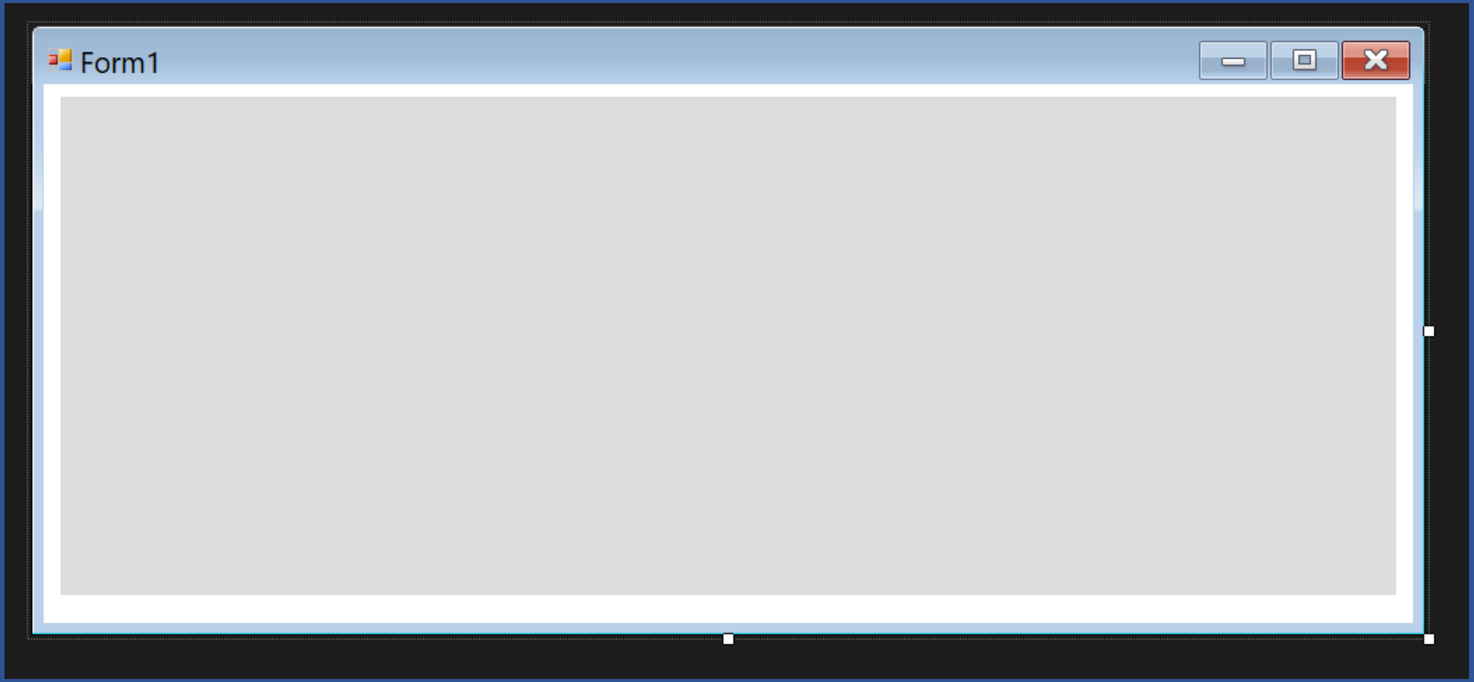
Add class members



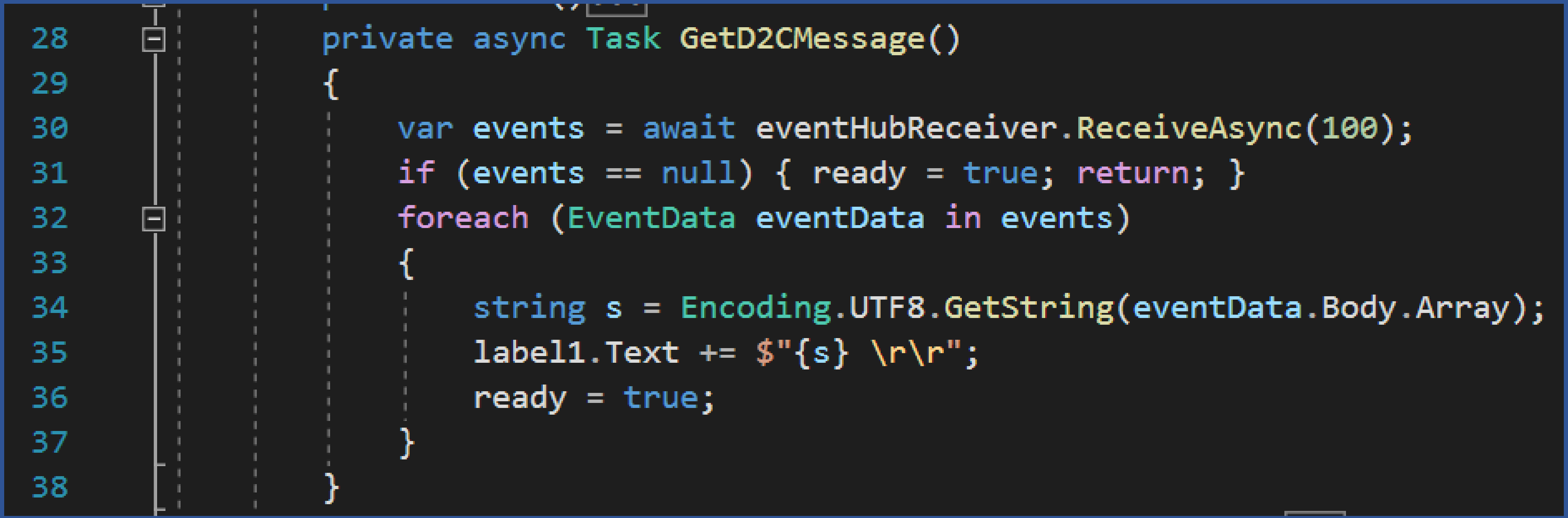
```
        private Timer myTimer = new Timer();
        private bool ready = true;

        private readonly string s_eventHubsCompatibleEndpoint =
            "sb://ihsuprodsgr[redacted]ce.servicebus.windows.net/";
        private readonly string s_eventHubsCompatiblePath =
            "iothub-ehub-loy[redacted]-c4bf924f3c";
        private readonly string s_iotHubSasKey =
            "KCyf3omKkmWncXuk[redacted]MIxBWRoWgmAw=";
        private readonly string s_iotHubSasKeyName = "service";
        private EventHubClient s_eventHubClient;
        private PartitionReceiver eventHubReceiver;
```

Add a Label to Form1

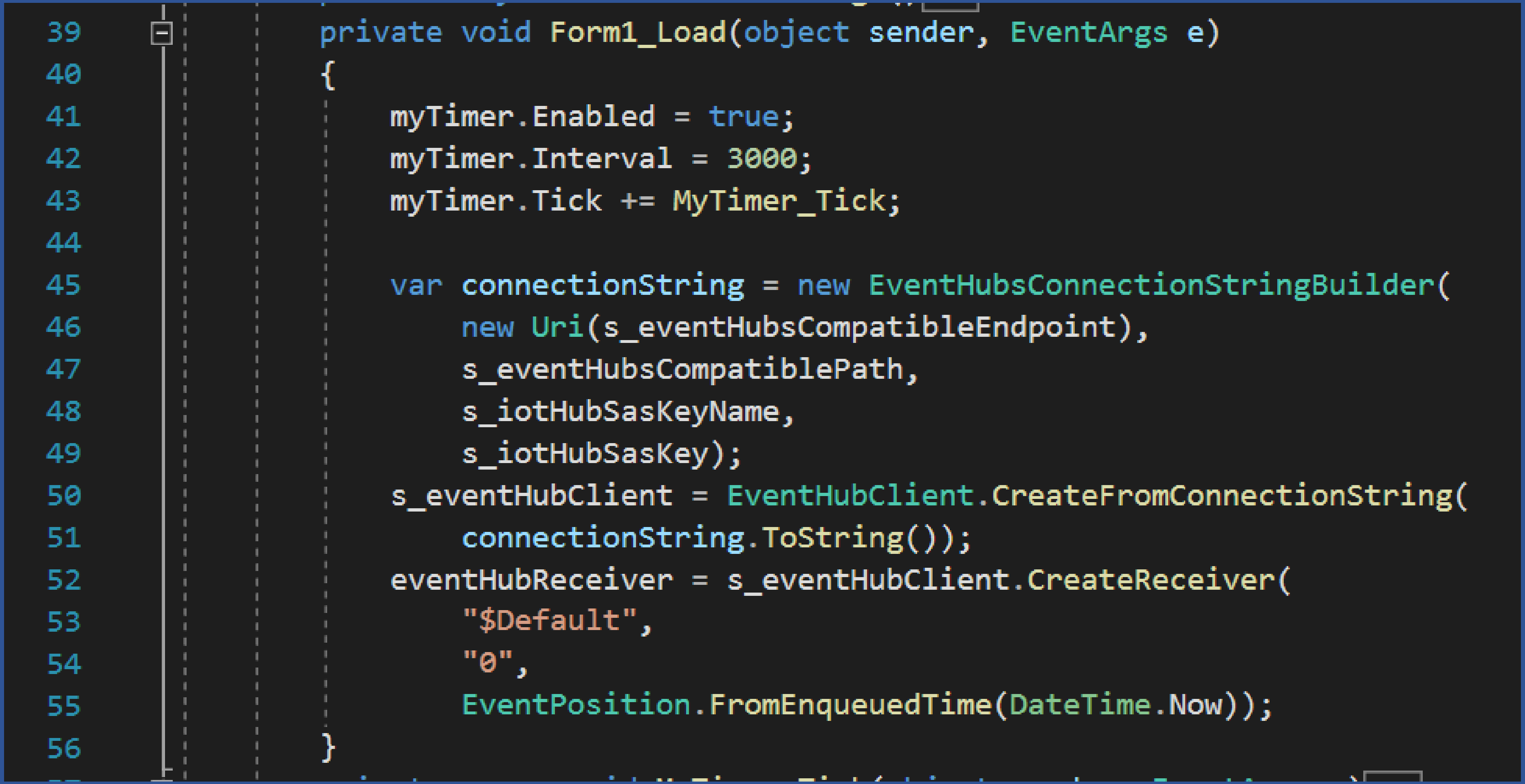


Add method GetD2CMessage()



```
private async Task GetD2CMessage()
{
    var events = await eventHubReceiver.ReceiveAsync(100);
    if (events == null) { ready = true; return; }
    foreach (EventData eventData in events)
    {
        string s = Encoding.UTF8.GetString(eventData.Body.Array);
        label1.Text += $"{s} \r\r";
        ready = true;
    }
}
```

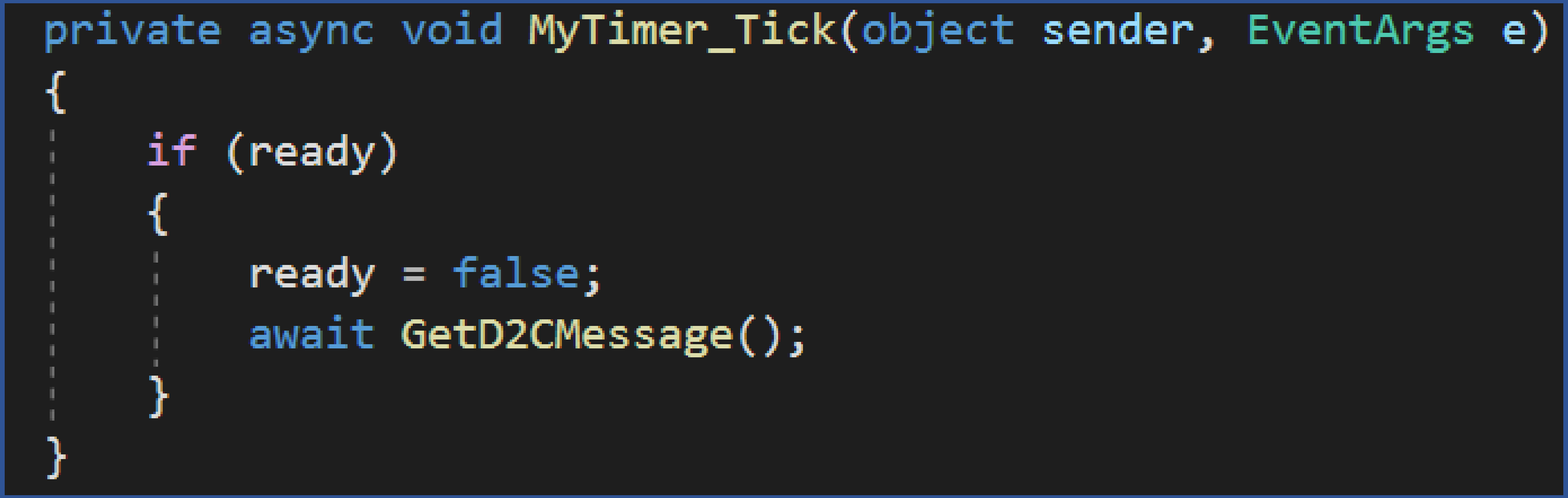
Add code to Form1_Load



```
private void Form1_Load(object sender, EventArgs e)
{
    myTimer.Enabled = true;
    myTimer.Interval = 3000;
    myTimer.Tick += MyTimer_Tick;

    var connectionString = new EventHubsConnectionStringBuilder(
        new Uri(s_eventHubsCompatibleEndpoint),
        s_eventHubsCompatiblePath,
        s_iotHubSasKeyName,
        s_iotHubSasKey);
    s_eventHubClient = EventHubClient.CreateFromConnectionString(
        connectionString.ToString());
    eventHubReceiver = s_eventHubClient.CreateReceiver(
        "$Default",
        "0",
        EventPosition.FromEnqueuedTime(DateTime.Now));
}
```

Add code to MyTimer_Tick



```
private async void MyTimer_Tick(object sender, EventArgs e)
{
    if (ready)
    {
        ready = false;
        await GetD2CMessage();
    }
}
```

What's next?