

# How to create a .NET 8 C# console application to send messages from D2C (from Device to Azure IoT Hub)

In this example we are going to create an application to simulate the Device.

From our application(Device) we will send a message to the Azure IoT Hub

For additional info about Azure IoT Hub Getting Start tutorials see the references:

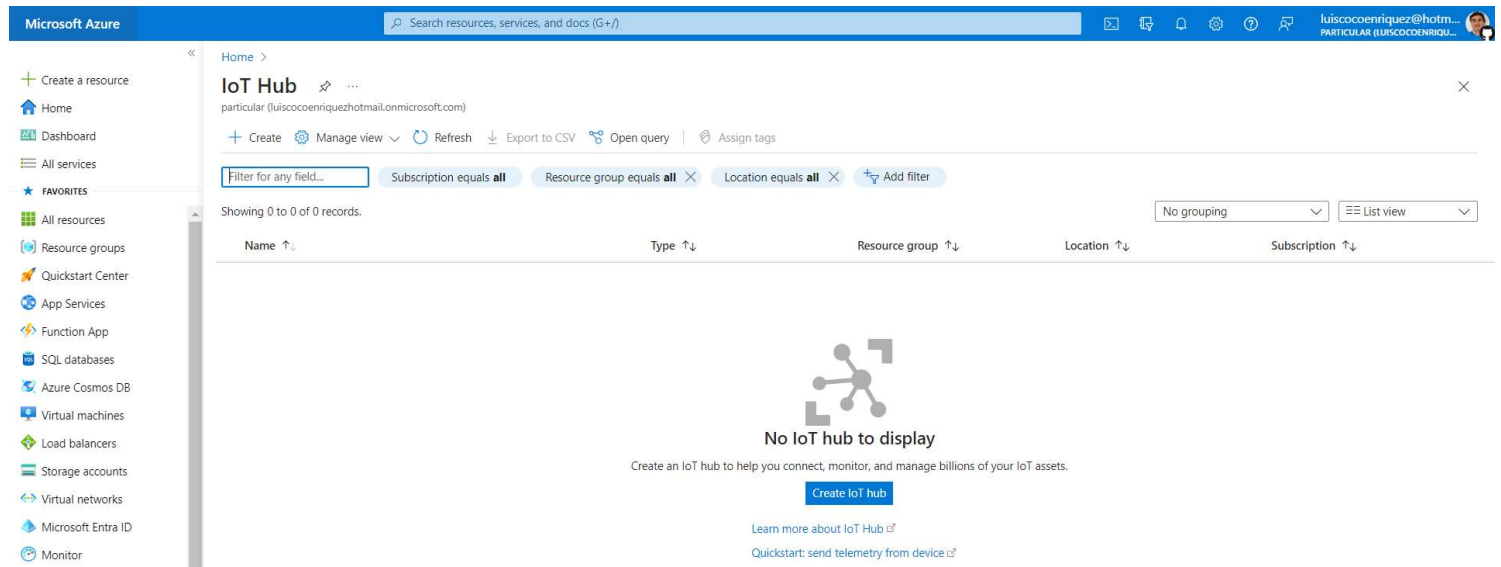
<https://github.com/Azure/azure-iot-explorer>

<https://github.com/Azure/azure-iot-sdk-csharp>

<https://learn.microsoft.com/en-us/azure/iot-develop/quickstart-send-telemetry-iot-hub?source=recommendations&pivots=programming-language-csharp>

## 0. Prerequisites

Create an Azure IoT Hub



**Microsoft Azure**

Search resources, services, and docs (G+/)

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Microsoft Entra ID

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Home > IoT Hub >

# IoT hub

Microsoft

BasicsNetworkingManagementAdd-onsTagsReview + create

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription \* ⓘAzure subscription 1

Resource group \* ⓘ(New) myRG

Create new

Instance details

IoT hub name \* ⓘmyIoTHubname1974

Region \* ⓘWest Europe

Tier \*Free

Free trial explores the app with live data. Trials cannot scale or be upgraded later.

Compare tiers

Daily message limit \* ⓘ8,000 (\$0/month)

Review + create

< Previous

Next: Networking >

**Microsoft Azure**

Search resources, services, and docs (G+/)

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Home > IoT Hub >

# IoT hub

Microsoft

BasicsNetworkingManagementAdd-onsTagsReview + create

You can connect to your IoT hub either publicly via its public hostname or privately using a private endpoint. [Learn more](#)

Connectivity configuration \*

☒ Public access

☐ Private access (Recommended)

You can change this or configure another connectivity method after this resource has been created. [Learn more](#)

**Microsoft Azure**

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Microsoft Entra ID

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Help + support

Home > IoT Hub >

# IoT hub

Microsoft

BasicsNetworking**Management**Add-onsTagsReview + create

**Role-based access control**

Change the permission model to Azure role-based access control (RBAC) only, or to a combination of shared access policies and RBAC. [Learn more](#)

Permission model

☐ RBAC only

☒ Shared access policy + RBAC

To manage the elements within an instance, a user needs access to IoT Hub data APIs. Select the suggested role below to grant yourself full access to the APIs. You can also use Access Control (IAM) to choose appropriate roles later. [Learn more](#)

Assign me

☒ IoT Hub Data Contributor role ⓘ

**Scale**

Device-to-cloud partitions \* ⓘ

**Preview mode**

⚠

Turning preview mode on means this IoT hub can't be used for production environments. Some existing features may not work. This setting can't be changed later. [Learn more](#)

Try new features like MQTT 5, TLS maximum fragment length negotiation, and ECC server cert. [Learn more](#)

Preview mode

☐ On

☒ Off

Review + create

< Previous: Networking

Next: Add-ons >

**Microsoft Azure**

Search resources, services, and docs (G+)

«

Home > IoT Hub >

# IoT hub

Microsoft

BasicsNetworkingManagementAdd-onsTagsReview + create

Pricing

IoT hub

\$0 USD  
per month  
[Change basics](#)

Add-ons total

[Change add-ons](#)

Basics

Subscription

Resource group

IoT hub name

Region

Disaster recovery enabled

Tier

Daily message limit

Azure subscription 1

myRG

myIoTHubname1974

West Europe

Yes

Free

8,000 (\$0/month)

Networking

Connectivity configuration

Private endpoint connections

Allow public network access

Public access

None

Enabled

Create

< Previous: Tags

Next >

[Automation options](#)

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Microsoft Entra ID

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Microsoft Azure

Search resources, services, and docs (G+/)

Home > myloTHubname1974-127175524 | Overview

Deployment

Search

Overview

Inputs

Outputs

Template

Delete Cancel Redeploy Download Refresh

**Your deployment is complete**

Deployment name: myloTHubname1974-127175524 Start time: 12/7/2023, 5:55:26 PM  
 Subscription: Azure subscription 1 Correlation ID: 7c988035-5190-4f8e-9f6e-2c27d245ad4a  
 Resource group: myRG

**Deployment details**

**Next steps**

Add and configure IoT Devices Recommended  
 Configure routing rules for device messaging Recommended

[Go to resource](#)

**Give feedback**

Tell us about your experience with deployment

Microsoft Azure

Search resources, services, and docs (G+/)

Home > myloTHubname1974-127175524 | Overview

myloTHubname1974 IoT Hub

Search

Move Delete Refresh Feedback

**Overview**

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

**Device management**

Devices

IoT Edge

Configurations + Deployments

Updates

Queries

**Hub settings**

Built-in endpoints

Message routing

File upload

Failover

Pricing and scale

**Essentials**

Resource group (move): myRG Hostname: myloTHubname1974.azure-devices.net  
 Status: Active Tier: Free  
 Location: West Europe Daily message limit: 8,000  
 Service region: West Europe  
 Subscription (move): Azure subscription 1  
 Tags (edit): Add tags  
[See more](#)

**Usage** Get started

Show data for last: 1 Hour 6 Hours 12 Hours 1 Day 7 Days 30 Days

**IoT Hub Usage**

- Messages used today: 0
- Daily messages quota: 8000
- IoT Devices: 0

**Number of messages used**

100  
90  
80  
70  
60  
50  
40  
30  
20

Create a Device inside the previously created Azure IoT Hub

Microsoft Azure

Search resources, services, and docs (G+/)

Home > myloTHubname1974

myloTHubname1974 IoT Hub

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

**Device management**

Devices

IoT Edge

View, create, delete, and update devices in your IoT Hub. [Learn more](#)

[Add Device](#) Edit columns Refresh Assign tags Delete

Find devices using a query

enter device ID Types: All Add filter

Device ID	Type	Status	Last status update	Authentication type	C2D messages que...	Tags
There are no devices to display.						



**Microsoft Azure**

Search resources, services

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Microsoft Entra ID

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Home > myIoTHubname1974 | Devices >

Create a device

Find Certified for Azure IoT devices in the Device Catalog

Device ID \* ⓘ  
myDevice

☐ IoT Edge Device

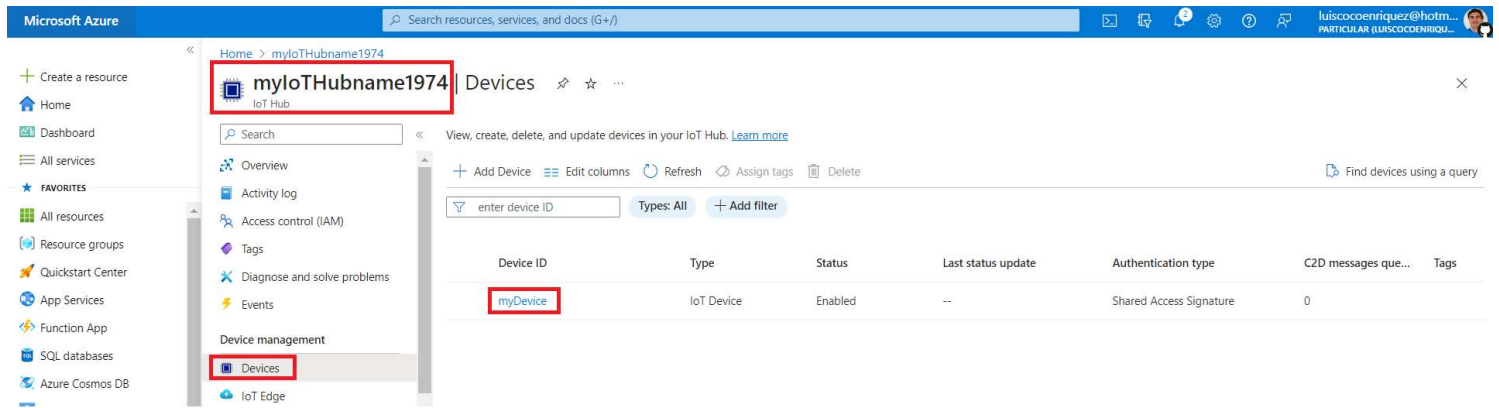
Authentication type ⓘ  
**Symmetric key** X.509 Self-Signed X.509 CA Signed

Auto-generate keys ⓘ  
☒

Connect this device to an IoT hub ⓘ  
**Enable** Disable

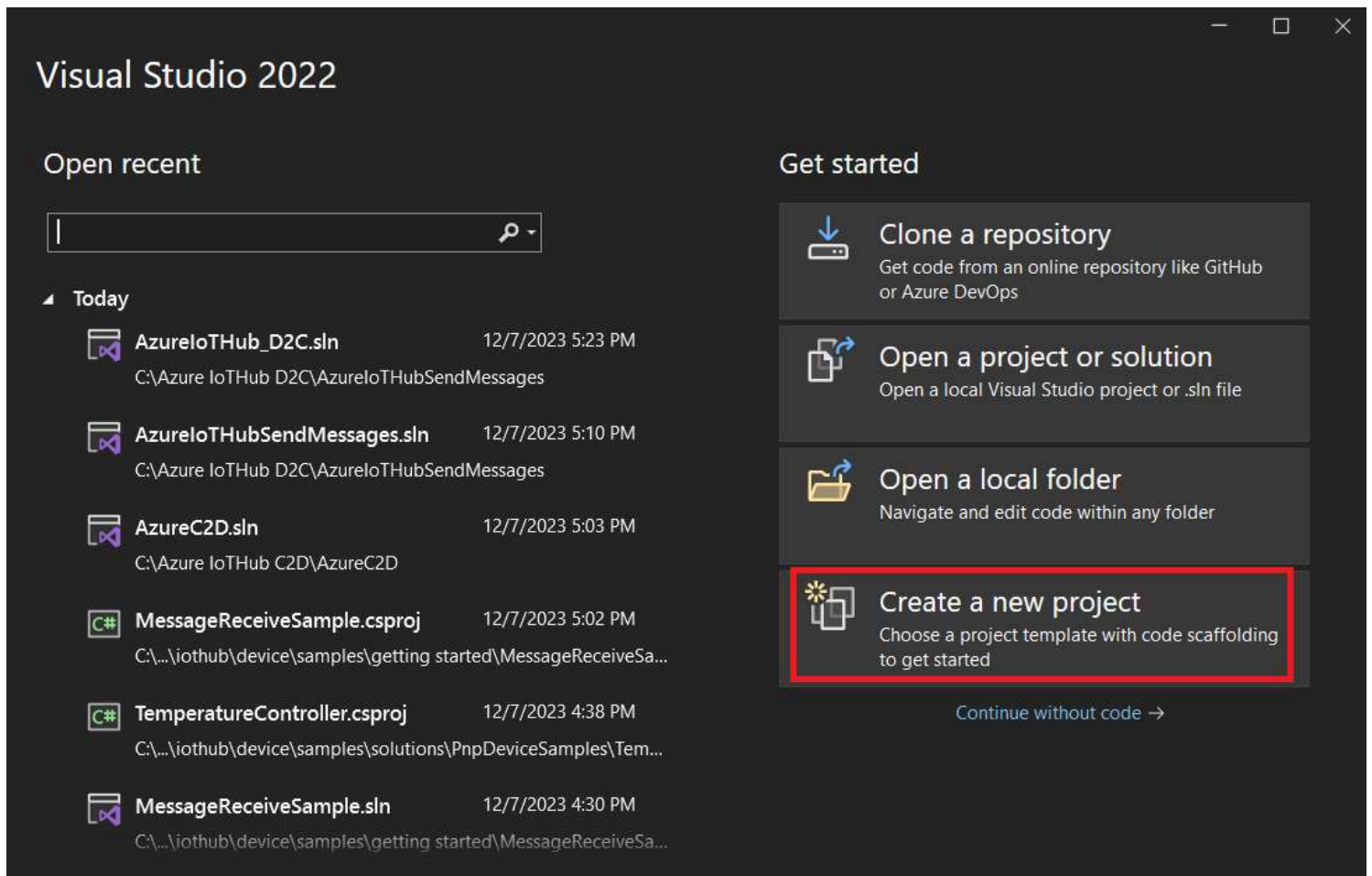
Parent device ⓘ  
**No parent device**  
[Set a parent device](#)

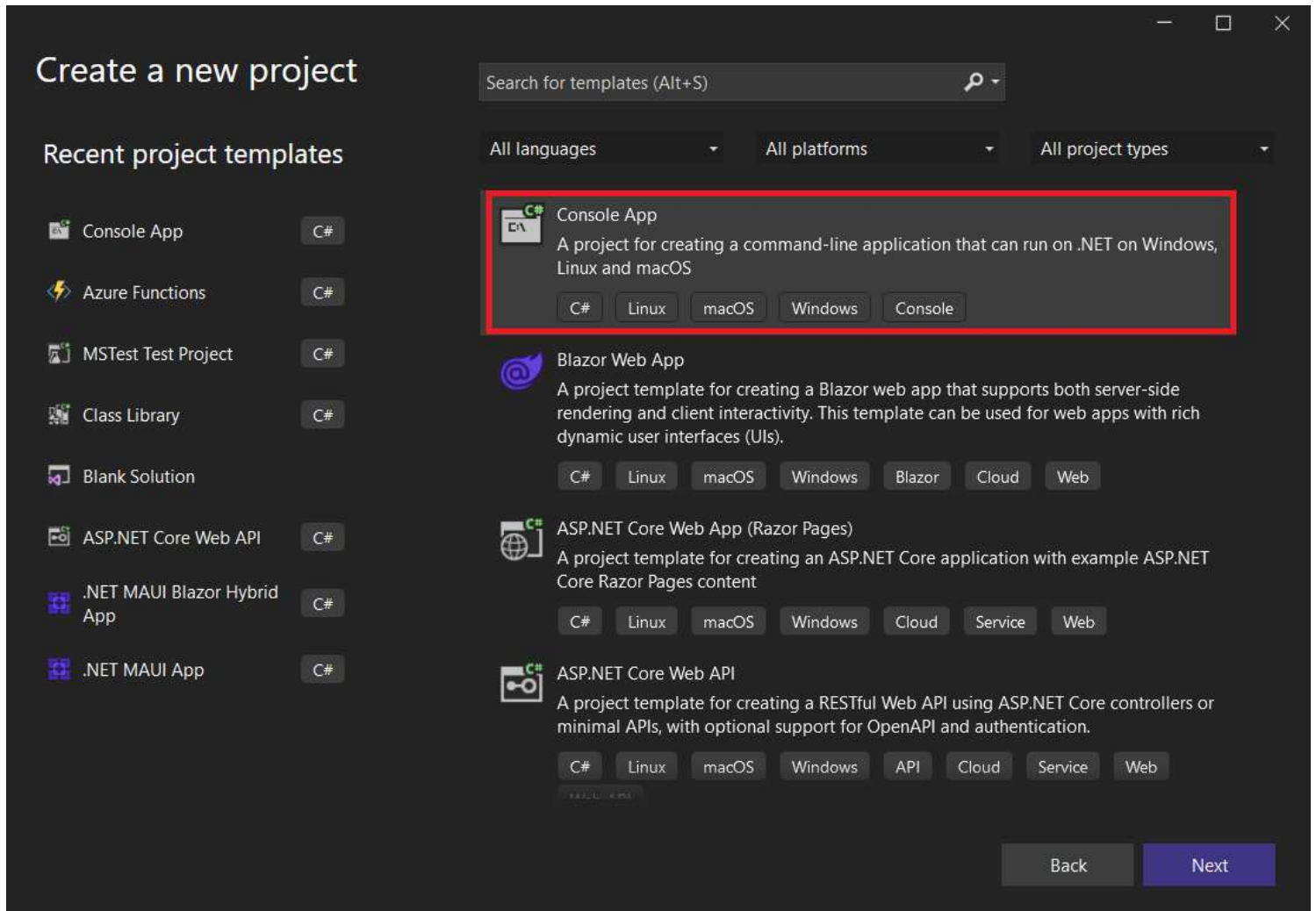
**Save**



# 1. Create a .NET 8 C# console application in Visual Studio 2022 Community Edition

Run Visual Studio 2022 and follow these steps







# Configure your new project

## Console App

C#

Linux

macOS

Windows

Console

Project name

AzureIoT Hub\_D2C

Location

C:\Azure IoT Hub D2C\

...

Solution name ⓘ

AzureIoT Hub\_D2C

☒ Place solution and project in the same directory

Project will be created in "C:\Azure IoT Hub D2C\AzureIoT Hub\_D2C\"

Back

Next

## Additional information

### Console App

C#


Linux

macOS

Windows

Console

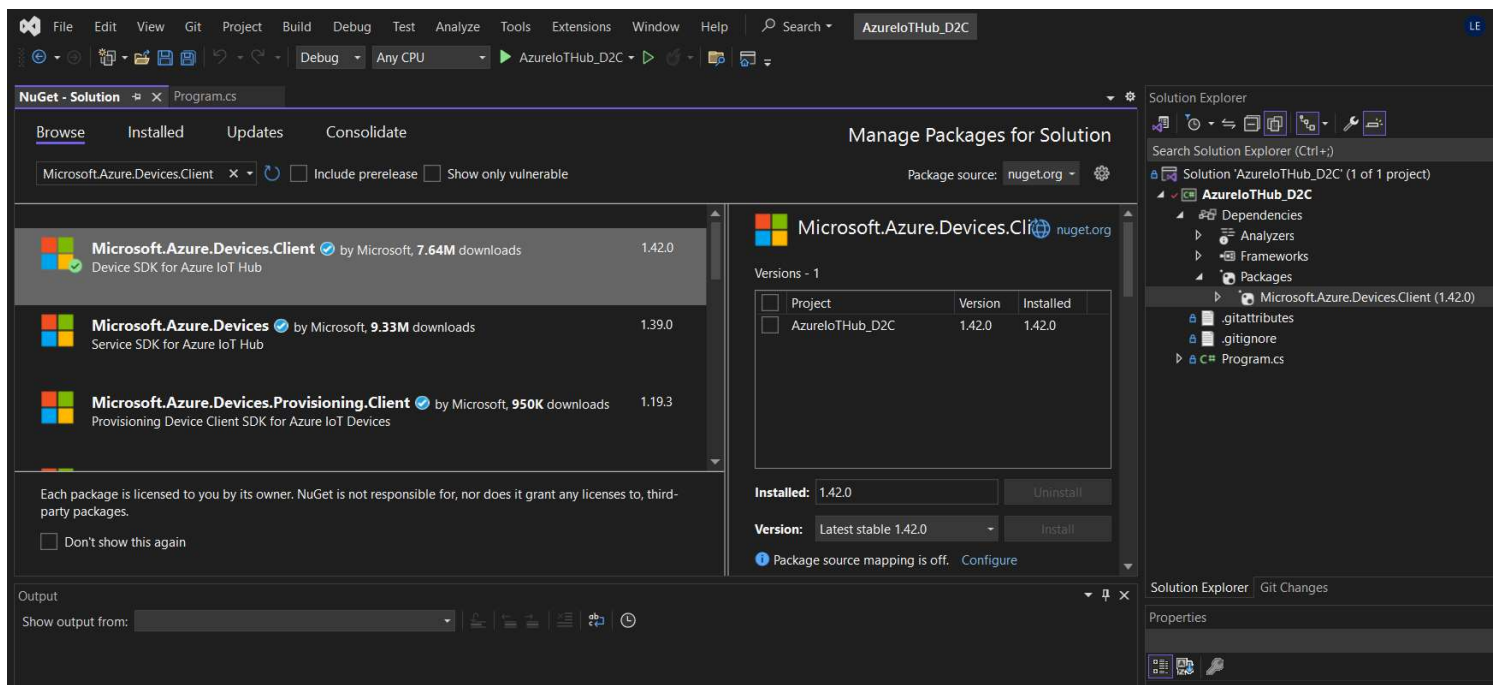
### Framework

☐ Do not use top-level statements ☐ Enable native AOT publish 

Back

Create

## Load with Nuget the library: Microsoft.Azure.Devices.Client



The screenshot shows the Visual Studio IDE with the NuGet Package Manager window open. The 'Browse' tab is active, displaying a list of packages. The package 'Microsoft.Azure.Devices.Client' is selected, showing its details on the right. The package is version 1.42.0, published by Microsoft, and has 7.64M downloads. The 'Versions' table shows that version 1.42.0 is installed in the project 'AzureIoT\_D2C'. The 'Solution Explorer' on the right shows the project structure, including the 'Program.cs' file.

**NuGet - Solution** | Program.cs

**Manage Packages for Solution**

Package source: nuget.org

**Microsoft.Azure.Devices.Client** by Microsoft, 7.64M downloads, 1.42.0

Device SDK for Azure IoT Hub

**Microsoft.Azure.Devices** by Microsoft, 9.33M downloads, 1.39.0

Service SDK for Azure IoT Hub

**Microsoft.Azure.Devices.Provisioning.Client** by Microsoft, 950K downloads, 1.19.3

Provisioning Device Client SDK for Azure IoT Devices

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.


☐ Don't show this again

**Versions - 1**

Project	Version	Installed
AzureIoT_D2C	1.42.0	1.42.0

**Installed:** 1.42.0 **Uninstall**

**Version:** Latest stable 1.42.0 **Install**

 Package source mapping is off. **Configure**

**Solution Explorer**

Search Solution Explorer (Ctrl+;) | Solution 'AzureIoT\_D2C' (1 of 1 project)

- AzureIoT\_D2C
  - Dependencies
  - Analyzers
  - Frameworks
  - Packages
    - Microsoft.Azure.Devices.Client (1.42.0)
  - .gitattributes
  - .gitignore
  - Program.cs

**Solution Explorer** | Git Changes

**Properties**

## 2. Input the application source code

---

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

string ConnectionString = "HostName=myIoTHubname1974.azure-devices.net;DeviceId=myDevice;SharedAc

var message = "Hello from Azure IoT Hub!";
await SendMessageToIoTHubAsync(message);

async Task SendMessageToIoTHubAsync(string message)
{
    var deviceClient = DeviceClient.CreateFromConnectionString(ConnectionString, TransportType.Mq

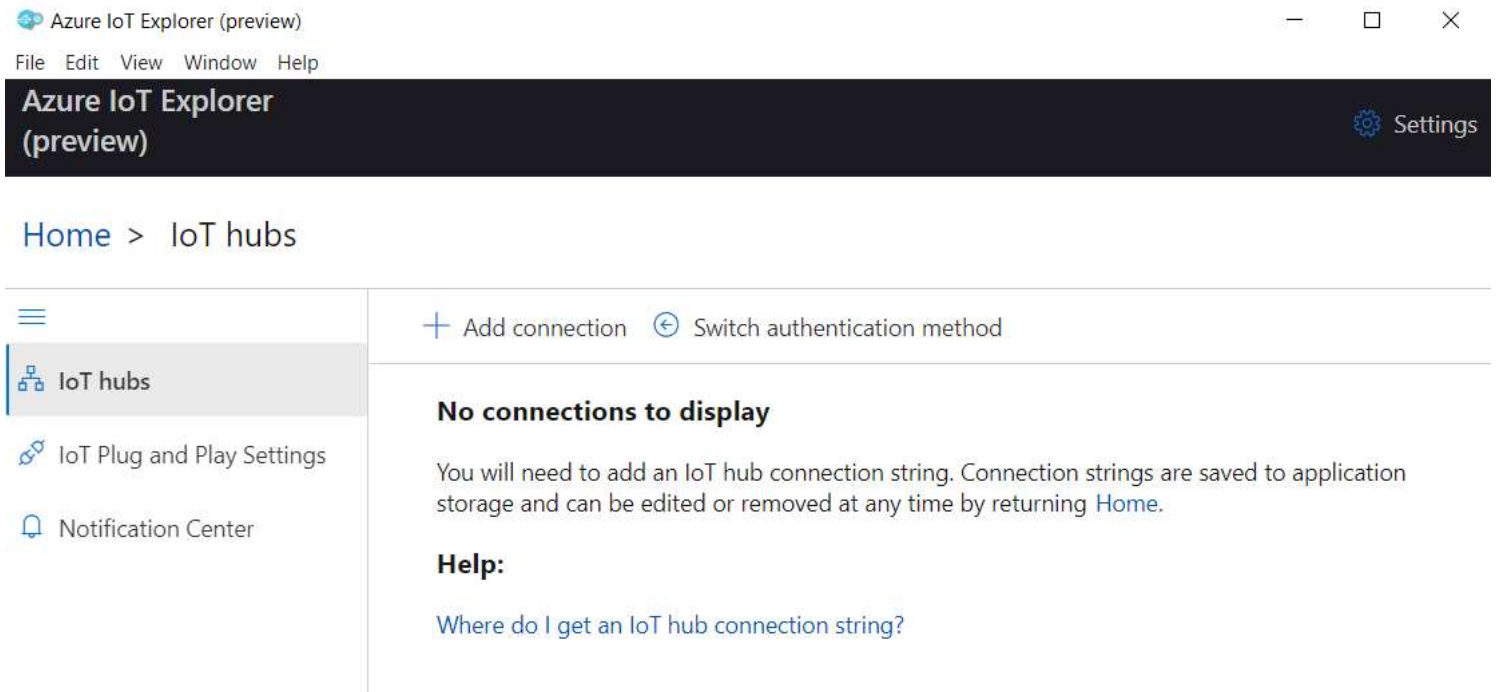
    var payload = new Message(Encoding.UTF8.GetBytes(message));

    try
    {
        await deviceClient.SendEventAsync(payload);
        Console.WriteLine($"Message sent: {message}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error sending message: {ex.Message}");
    }
    finally
    {
        await deviceClient.CloseAsync();
    }
}
```

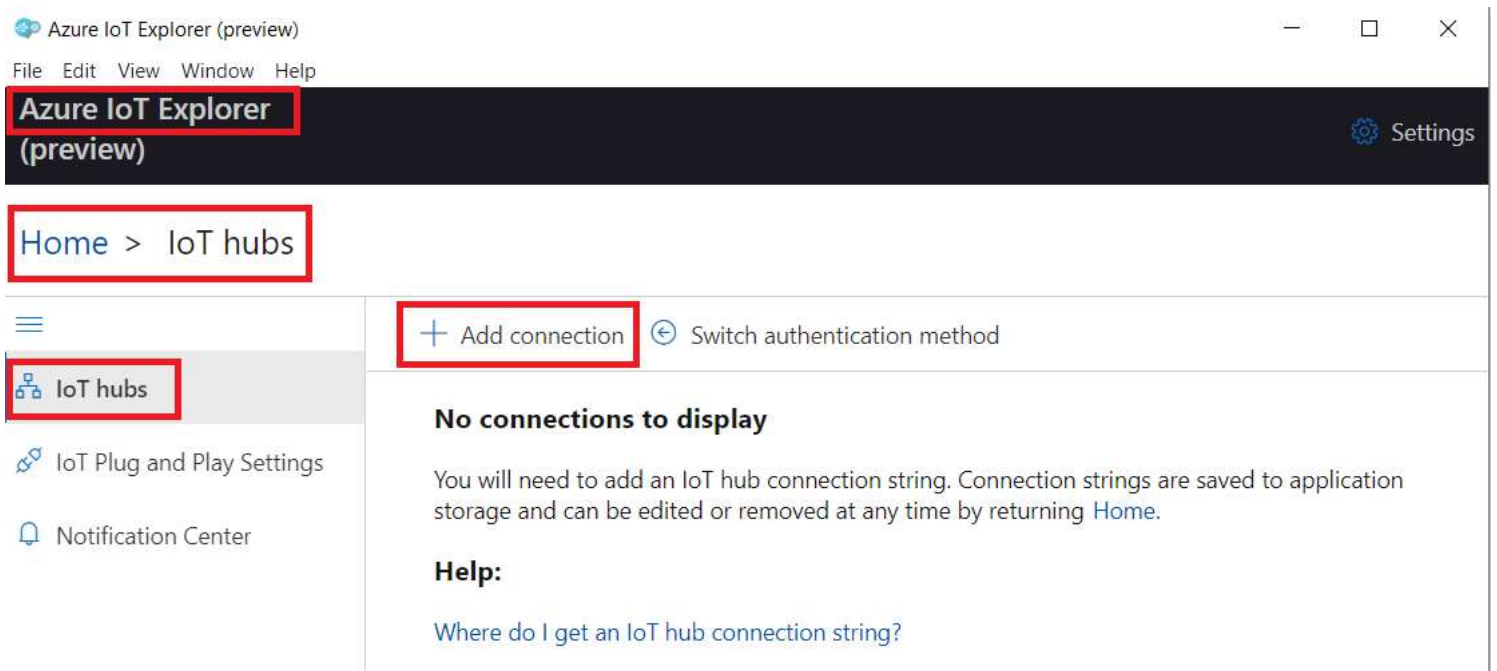
## 3. Build and Run the application and Verify it with Azure.IoT.Explorer

---

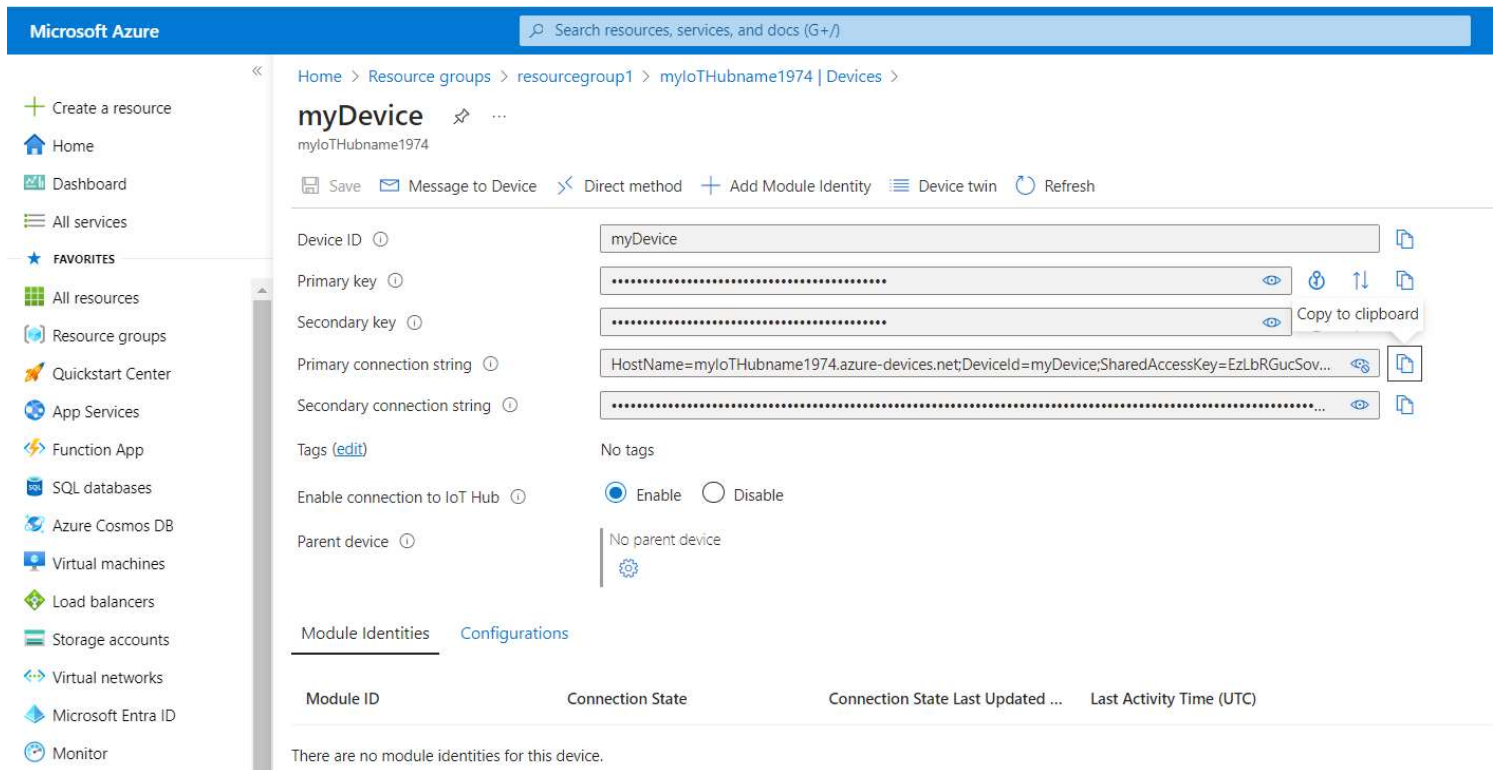
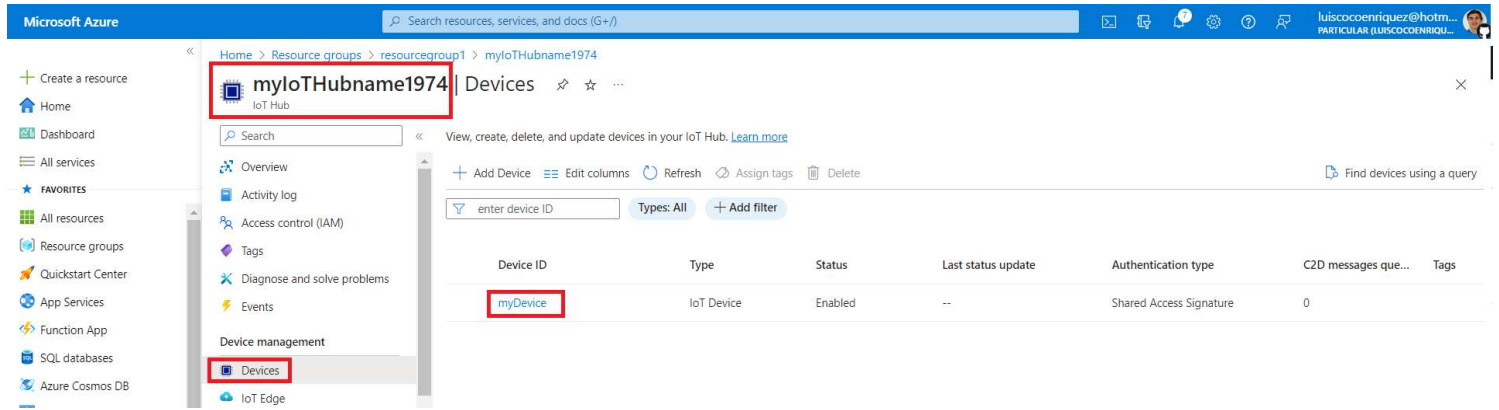
Download and install the Azure IoT Explorer: <https://github.com/Azure/azure-iot-explorer>



Add a new Azure IoT Hub connection in the Azure IoT Explorer



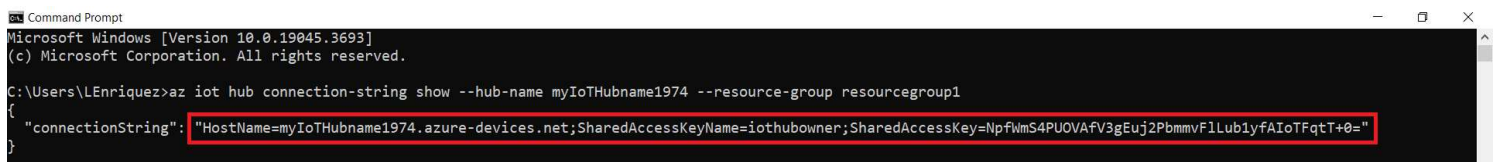
Navigate to the Azure IoT Hub and copy the connection string



Or get the connection string running this command

```
az iot hub connection-string show --hub-name myIoTHubname1974 --resource-group resourcegroup1
```

Do not forget to remove the quotes "" from the connection string



Add and configure a new connection to the Azure IoT Hub



Azure IoT Explorer (preview)

File Edit View Window Help

Home > IoT hubs

+ Add connection Switch authentication method

No connections to display

You will need to add an IoT hub connection string. Connection strings are saved to application storage and can be edited or removed at any time by returning Home.

Help:

Where do I get an IoT hub connection string?

Where do I get an IoT hub connection string?  
Please do not save your hub connection string to any unsafe locations

Connection string \*

```
HostName=myIoTHubname1974.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=NpfWmS4PUOVAfV3gEuj2PbmmvFLub1yfAlotFqT+0=
```

Host name

myIoTHubname1974.azure-devices.net

Shared access policy name

iothubowner

Shared access policy key

.....

Save Cancel

Save the connection and we can see out device inside out Azure IoT Hub. Click on the Device Id

Azure IoT Explorer (preview)

File Edit View Window Help

Home > myIoTHubname1974 > Devices

+ New Refresh Delete

Query by device ID...

Add query parameter

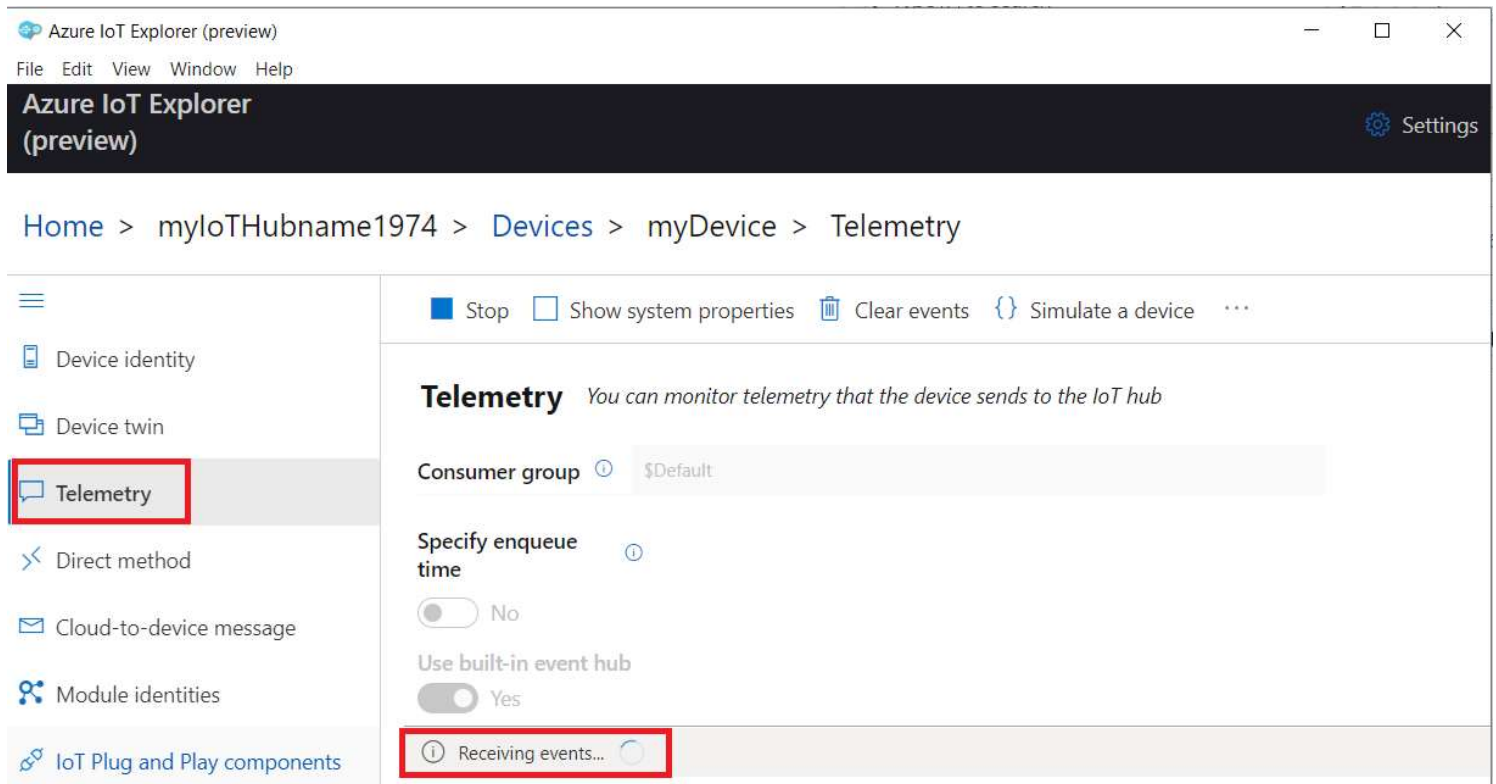
Device ID	Status	Connection...	Authentic...	Last status...	IoT Plug and ...	Edge device
myDevice	Enabled	Disconnected	Sas	--		

Now we can see the device option in the left menu

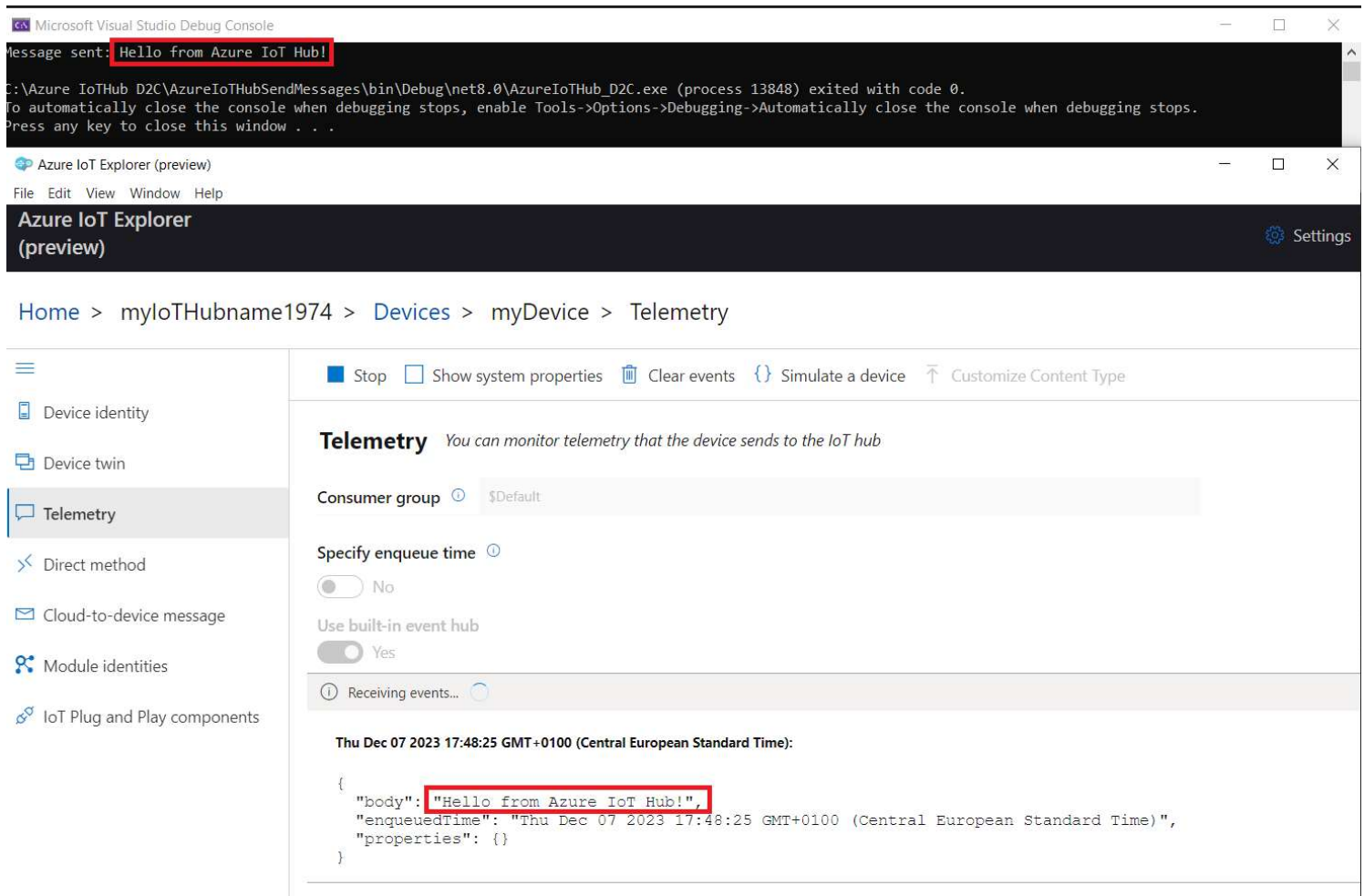
The screenshot shows the Azure IoT Explorer (preview) interface. The breadcrumb navigation at the top reads: Home > myIoTHubname1974 > Devices > myDevice > Device identity. The left sidebar contains a menu with options: Device identity (selected), Device twin, Telemetry, Direct method, Cloud-to-device message, Module identities, and IoT Plug and Play components. The main content area is titled "Device identity" and includes fields for Device ID (myDevice), Primary key, Secondary key, Primary connection string (HostName=myIoTHubname1974.azure-devices.net;DeviceId=myDevice;SharedAccessKey=EzLbRGucSovGeSzk8WcflvDuTqk752tpRAIoTO9Zbfk=), and Secondary connection string. There is a checkbox for "Connection string with SAS token" and a toggle switch for "Connect this device to IoT hub" which is currently set to "Enable".

We are going to test sending a message from the device to the cloud (D2C), for this purpose we select the **Telemetry** option in the left menu.

The screenshot shows the Azure IoT Explorer (preview) interface with the breadcrumb navigation highlighted by a red box: Home > myIoTHubname1974 > Devices > myDevice > Telemetry. The left sidebar menu also has the "Telemetry" option highlighted with a red box. The main content area is titled "Telemetry" with the subtitle "You can monitor telemetry that the device sends to the IoT hub". At the top of the main area, there is a toolbar with buttons: Start (highlighted with a red box), Show system properties, Clear events, and Simulate a device. Below the toolbar, the "Consumer group" is set to "\$Default". The "Specify enqueue time" section has a toggle switch set to "No". The "Use built-in event hub" section has a toggle switch set to "Yes".



It is the time to start the C# console application to send a message from our application(our simulated device) to the Azure IoT Hub



```
Message sent: Hello from Azure IoT Hub!
```

```
C:\Azure_IoTHub_D2C\AzureIoTHubSendMessage\bin\Debug\net8.0\AzureIoTHub_D2C.exe (process 13848) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

Thu Dec 07 2023 17:48:25 GMT+0100 (Central European Standard Time):

```
{  
  "body": "Hello from Azure IoT Hub!",  
  "enqueueTime": "Thu Dec 07 2023 17:48:25 GMT+0100 (Central European Standard Time)",  
  "properties": {}  
}
```