Quickstart - Queclink to Azure IoT Hub

Quickly integrate Queclink devices to Azure IoT Hub

Queclink to Azure IoT Hub

This guide will walk you though integrating a Queclink device to Azure IoT Hub using an available solution template.

Pre-requisites

1. Account on Tartabit IoT Bridge.
2. Access to a Microsoft Azure subscription.
3. Queclink `GL500` or `GV500` devices.

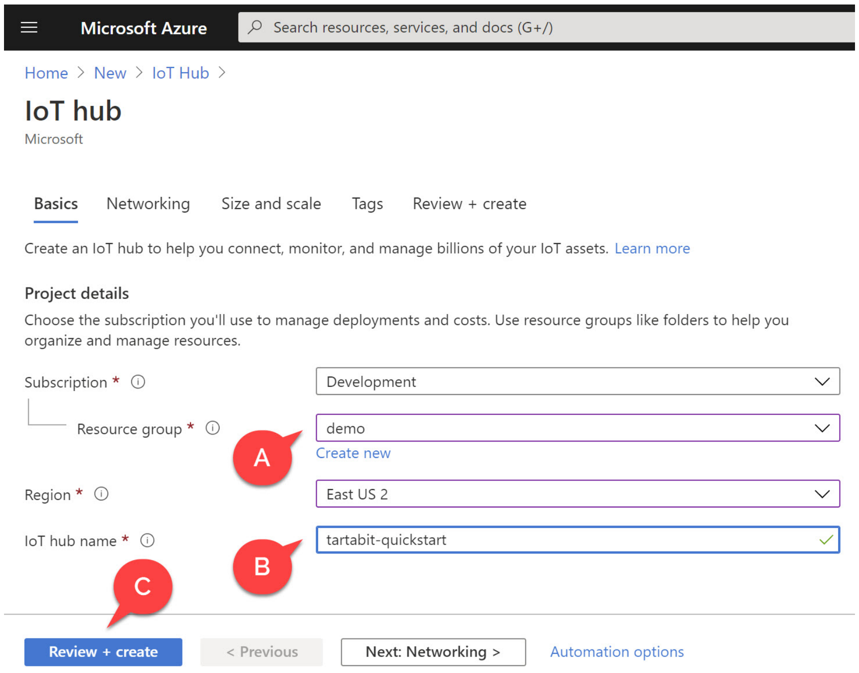
Details of this guide

- In this guide you will:
  - Create a new Azure IoT Hub.
  - Retrieve the credentials needed to connect to your Azure IoT Hub.
  - Import a solution template, this will create all of the required triggers, services, and endpoints required to connect your device.
  - Connect a Queclink device
  - Verify that the device is properly reporting into the Azure IoT Hub.

1. Create an IoT Hub in Azure

If you already have an IoT Hub, you can skip this step.

For a detailed walkthrough from Microsoft, check out the following link: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal



IoT hub
Microsoft

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. Learn more

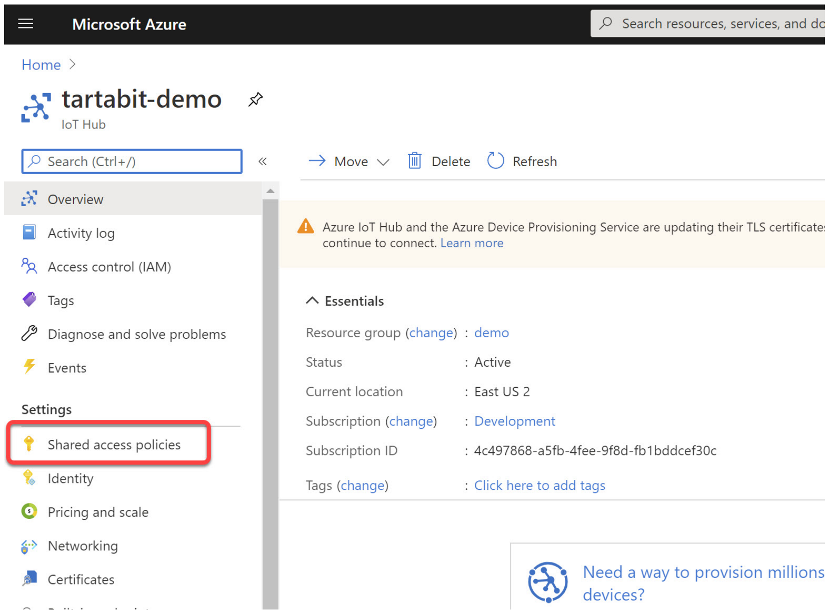
Project details
Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription *	Development
Resource group *	demo
	Create new
Region *	East US 2
IoT hub name *	tartabit-quickstart

- A: Select an existing resource group or create one.
- B: Provide a unique name for your IoT Hub.
- C: You can skip the other tabs and create a new Hub.

2. Retrieve your IoT Hub connection string

1. Navigate to your new Azure IoT Hub.
2. Click **Shared Access Policies**.



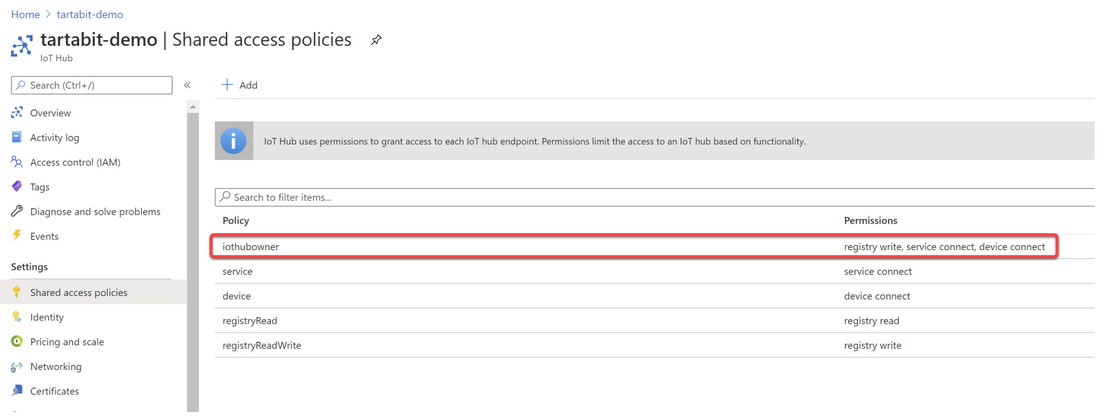
tartabit-demo
IoT Hub

Azure IoT Hub and the Azure Device Provisioning Service are updating their TLS certificate continue to connect. Learn more

Essentials

Resource group (change)	: demo
Status	: Active
Current location	: East US 2
Subscription (change)	: Development
Subscription ID	: 4c497868-a5fb-4fee-9f8d-fb1bddcef30c
Tags (change)	: Click here to add tags

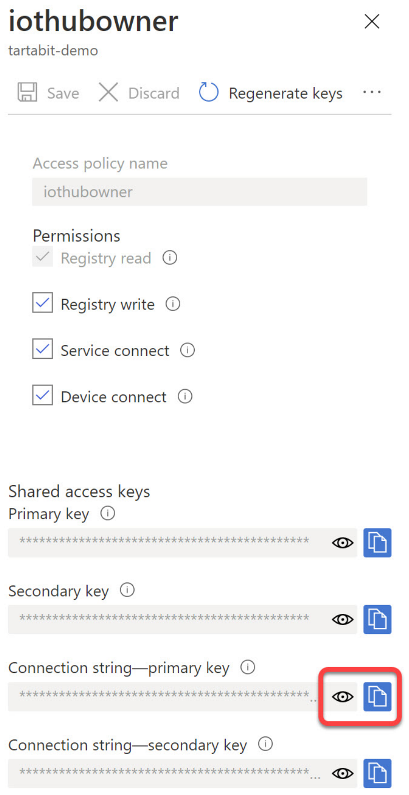
3. Select **iothubowner** (this is required because the IoT Bridge will automatically provision devices in the IoT Hub as they are needed).



IoT Hub uses permissions to grant access to each IoT hub endpoint. Permissions limit the access to an IoT hub based on functionality.

Policy	Permissions
iothubowner	registry write, service connect, device connect
service	service connect
device	device connect
registryRead	registry read
registryReadWrite	registry write

4. Copy the **Connection string - primary key** value and save it for later.



3. Prepare a Queclink device

Ensure your devices is configured according to the instructions below to ensure accurate reporting.

GV500 device

1. Configure backend server registration

- Report Mode: `UDP mode`
- Main Server IP: `queclink-us.tartabit.com` (or the correct URL for your server).
- Main Server Port: `10000`
- Buffer Mode: `0: Disable`
- Protocol Format: `HEX`

```
AT+GTSRI=gv500,4,,0,queclink-us.tartabit.com,10000,,0,,0,0,1,0,,,FFFF$
```

2. Ensure the correct minimum hex report masks are present, the `RSP Mask` and `EVT Mask` must have the following fields enabled:

- *Check* Device Type
- *Check* Length
- *Uncheck* Device Name (highly recommended to use IMEI instead of custom name)

```
AT+GTHRM=gv500,,,6F,FDF7FF,FC37FF,FD7D,EF,7D,7D,,,FFFF$
```

GL500 device

1. Configure backend server registration

- Report Mode: `UDP mode`
- Main Server IP: `queclink-us.tartabit.com` (or the correct URL for your server).
- Main Server Port: `10000`
- Buffer Mode: `0: Disable`
- Protocol Format: `HEX`

```
AT+GTSRI=gl500m,4,1,0,queclink-us.tartabit.com,10000,,0,,0,0,0,0,1,,FFFF$
```

2. Ensure the correct minimum hex report masks are present, the `RSP Mask` and `EVT Mask` must have the following fields enabled:

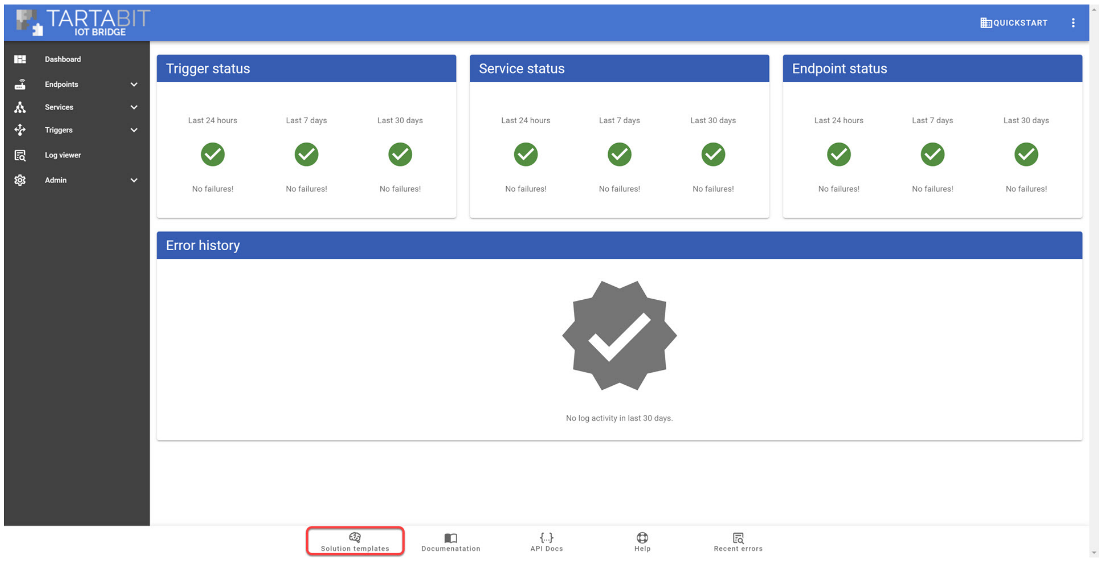
- *Check* Device Type
- *Check* Length
- *Uncheck* Device Name (highly recommended to use IMEI instead of custom name)

```
AT+GTHRM=gl500m,,,7F,FF6F,FF6F,F77F,7F,,,,,FFFF$
```

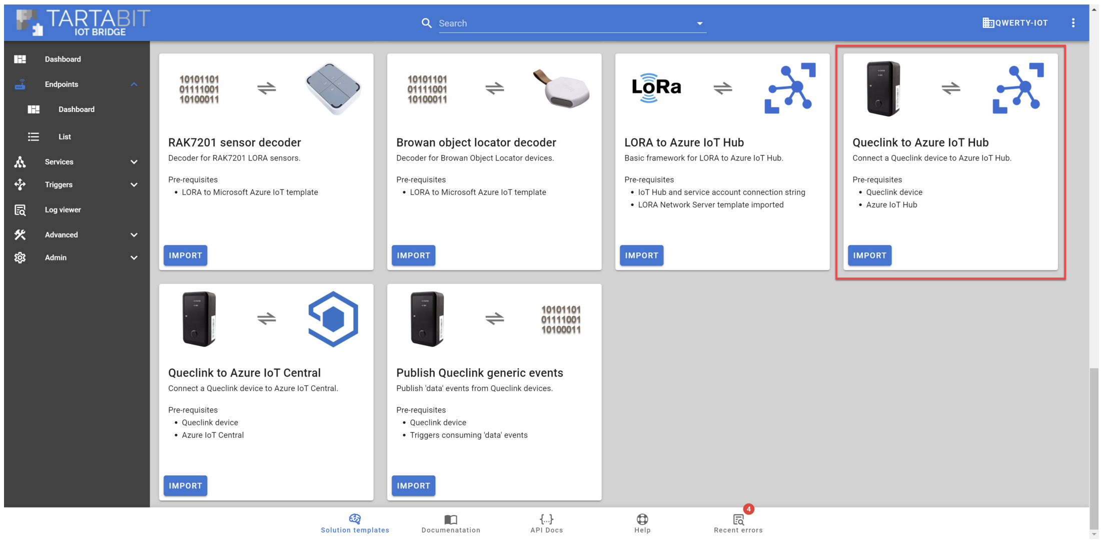
4. Import the solution template

Follow the steps below to import the solution template:

1. Click **Solution Templates** at the bottom of any page.



2. Find the **Queclink to Microsoft Azure IoT Hub** template and click **Import**.



RAK7201 sensor decoder
Decoder for RAK7201 LORA sensors.
Pre-requisites
- LORA to Microsoft Azure IoT template

IMPORT

Browan object locator decoder
Decoder for Browan Object Locator devices.
Pre-requisites
- LORA to Microsoft Azure IoT template

IMPORT

LORA to Azure IoT Hub
Basic framework for LORA to Azure IoT Hub.
Pre-requisites
- IoT Hub and service account connection string
- LORA Network Server template imported

IMPORT

Queclink to Azure IoT Hub
Connect a Queclink device to Azure IoT Hub.
Pre-requisites
- Queclink device
- Azure IoT Hub

IMPORT

Queclink to Azure IoT Central
Connect a Queclink device to Azure IoT Central.
Pre-requisites
- Queclink device
- Azure IoT Central

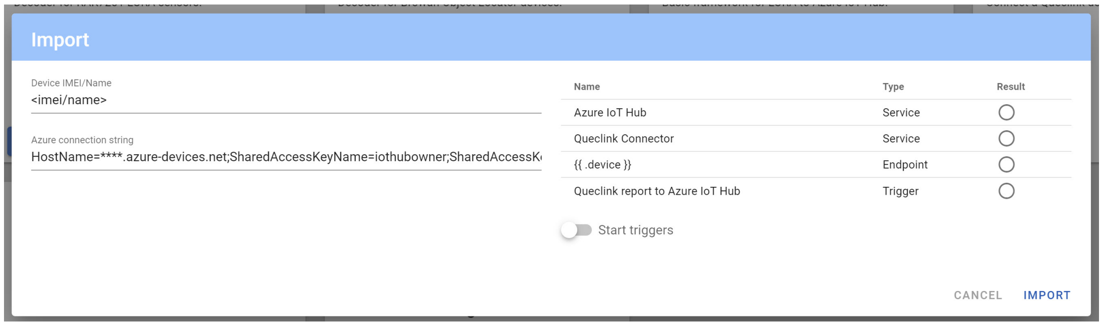
IMPORT

Publish Queclink generic events
Publish 'data' events from Queclink devices.
Pre-requisites
- Queclink device
- Triggers consuming 'data' events

IMPORT

3. Follow the instructions below to import the template:

- A: Enter the IMEI or device name of your device.
- B: Enter the Azure IoT Hub connection string that you previously saved in Step #1.
- C: Select the option to **Start Triggers** this will automatically start all of the new triggers upon import.
- D: Click import when finished.



Import

Device IMEI/Name
<imei/name>

Azure connection string
HostName=****.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessK

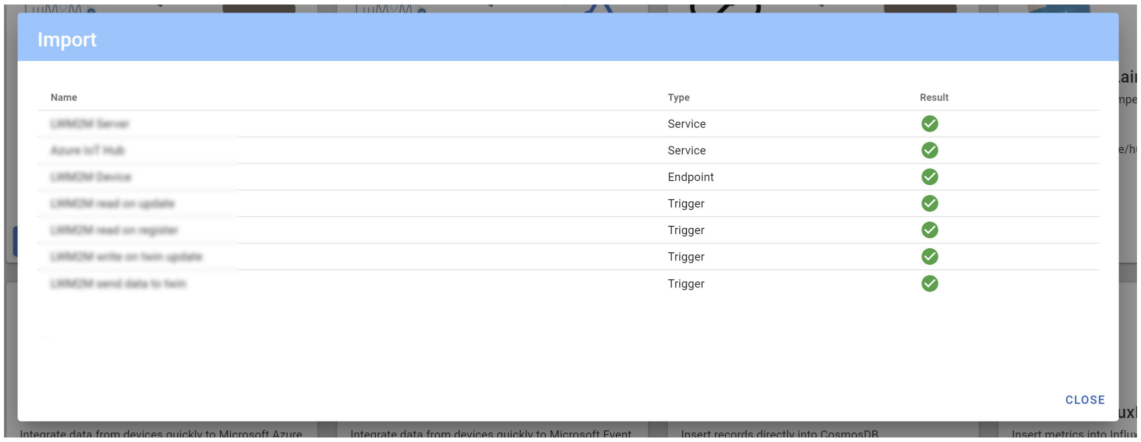
Name	Type	Result
Azure IoT Hub	Service	○
Queclink Connector	Service	○
{{ .device }}	Endpoint	○
Queclink report to Azure IoT Hub	Trigger	○

Start triggers

CANCEL IMPORT

5. Verify the import was successful

You should receive an import result like this:



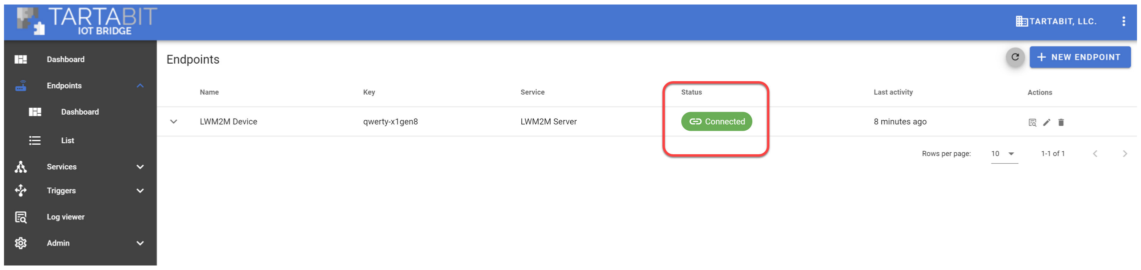
Import

Name	Type	Result
LWM2M Server	Service	✓
Azure IoT Hub	Service	✓
LWM2M Device	Endpoint	✓
LWM2M read on update	Trigger	✓
LWM2M read on register	Trigger	✓
LWM2M write on twin update	Trigger	✓
LWM2M send data to twin	Trigger	✓

6. Connect your Queclink device

Now that everything has been imported, you can connect your first Queclink device.

Verify that the client is connected by navigating to **Endpoints -> List** and checking the status.



Endpoints

Name	Key	Service	Status	Last activity	Actions
LWM2M Device	qwerty-x1gen8	LWM2M Server	Connected	8 minutes ago	

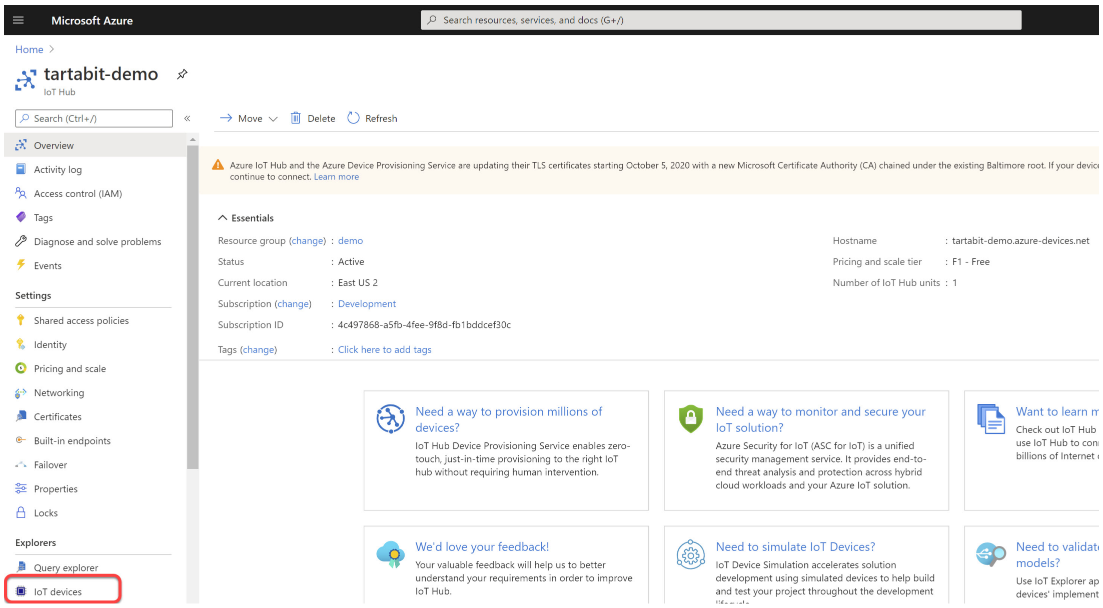
Rows per page: 10 1-1 of 1

7. Check your device in Azure IoT Hub

With the device connected, you should now see it connected in the IoT Hub, and data being updated in the device twin.

1. Navigate to your Azure IoT Hub

2. Click **IoT Devices**



Azure IoT Hub and the Azure Device Provisioning Service are updating their TLS certificates starting October 5, 2020 with a new Microsoft Certificate Authority (CA) chained under the existing Baltimore root. If your devic
continue to connect. Learn more

Essentials

Resource group (change)	: demo	Hostname	: tartabit-demo.azure-devices.net
Status	: Active	Pricing and scale tier	: F1 - Free
Current location	: East US 2	Number of IoT Hub units	: 1
Subscription (change)	: Development		
Subscription ID	: 4c497868-a5fb-4fee-9f8d-fb1bddcef30c		
Tags (change)	: Click here to add tags		

Need a way to provision millions of devices?
IoT Hub Device Provisioning Service enables zero-touch, just-in-time provisioning to the right IoT hub without requiring human intervention.

Need a way to monitor and secure your IoT solution?
Azure Security for IoT (ASC for IoT) is a unified security management service. It provides end-to-end threat analysis and protection across hybrid cloud workloads and your Azure IoT solution.

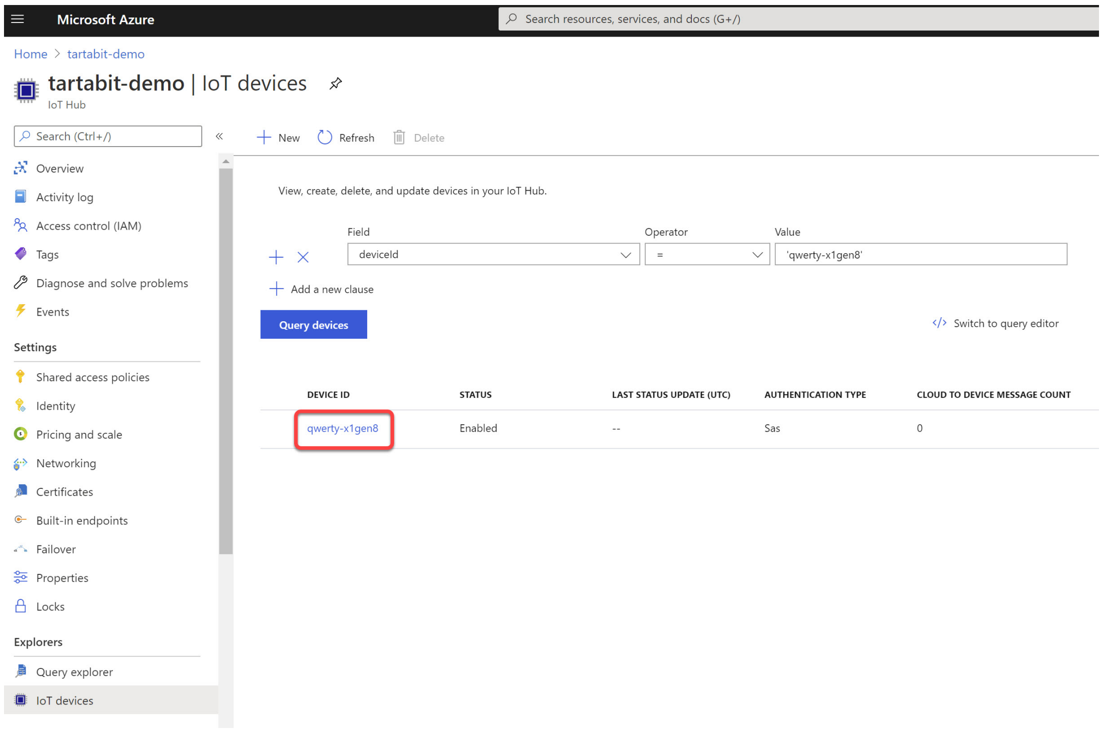
Want to learn m
Check out IoT Hub
use IoT Hub to con
billions of Internet

We'd love your feedback!
Your valuable feedback will help us to better understand your requirements in order to improve IoT Hub.

Need to simulate IoT Devices?
IoT Device Simulation accelerates solution development using simulated devices to help build and test your project throughout the development

Need to validat
models?
Use IoT Explorer ap
devices' implement

3. You should see your newly created device, click on it.

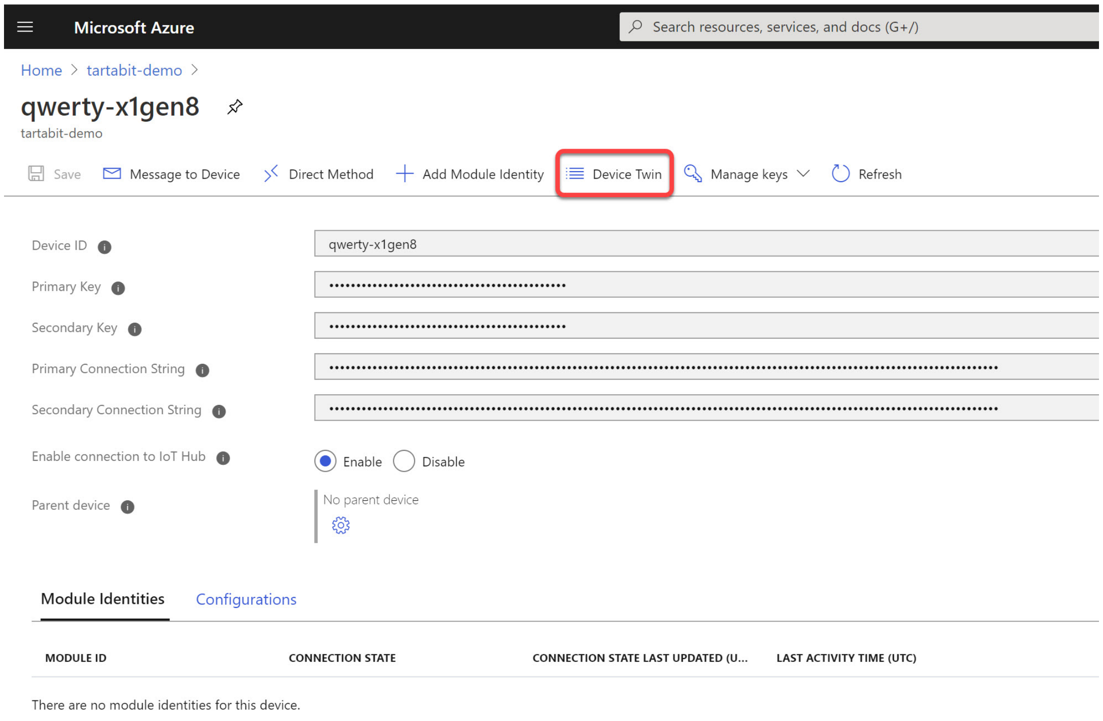


View, create, delete, and update devices in your IoT Hub.

Field	Operator	Value
deviceId	=	'qwerty-x1gen8'

DEVICE ID	STATUS	LAST STATUS UPDATE (UTC)	AUTHENTICATION TYPE	CLOUD TO DEVICE MESSAGE COUNT
qwerty-x1gen8	Enabled	--	Sas	0

4. Click on **Device Twin**



qwerty-x1gen8

tartabit-demo

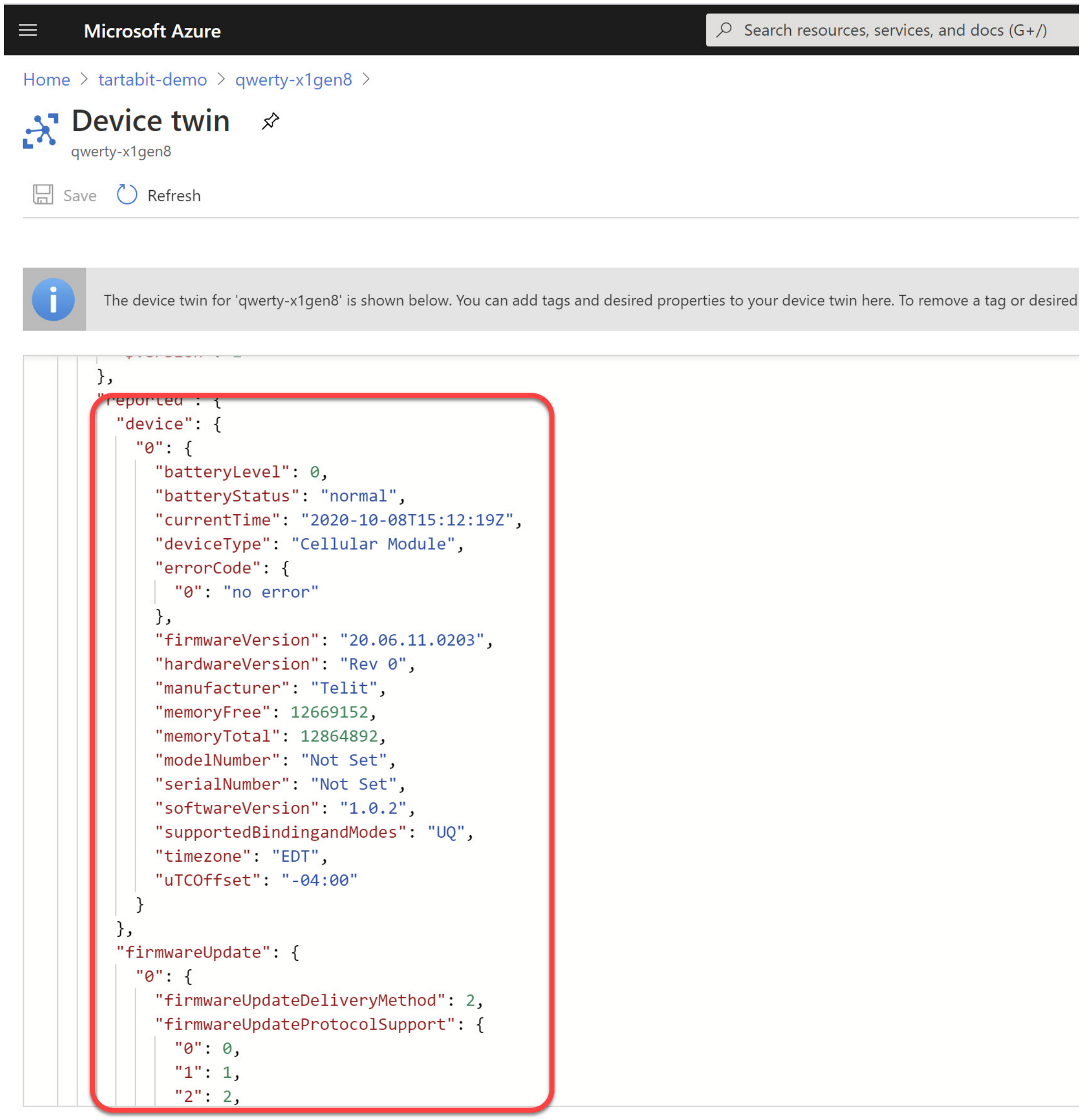
Device ID	qwerty-x1gen8
Primary Key	•••
Secondary Key	•••
Primary Connection String	••
Secondary Connection String	••
Enable connection to IoT Hub	[✓ Enable] Disable
Parent device	No parent device

Module Identities | Configurations

MODULE ID	CONNECTION STATE	CONNECTION STATE LAST UPDATED (U...	LAST ACTIVITY TIME (UTC)

There are no module identities for this device.

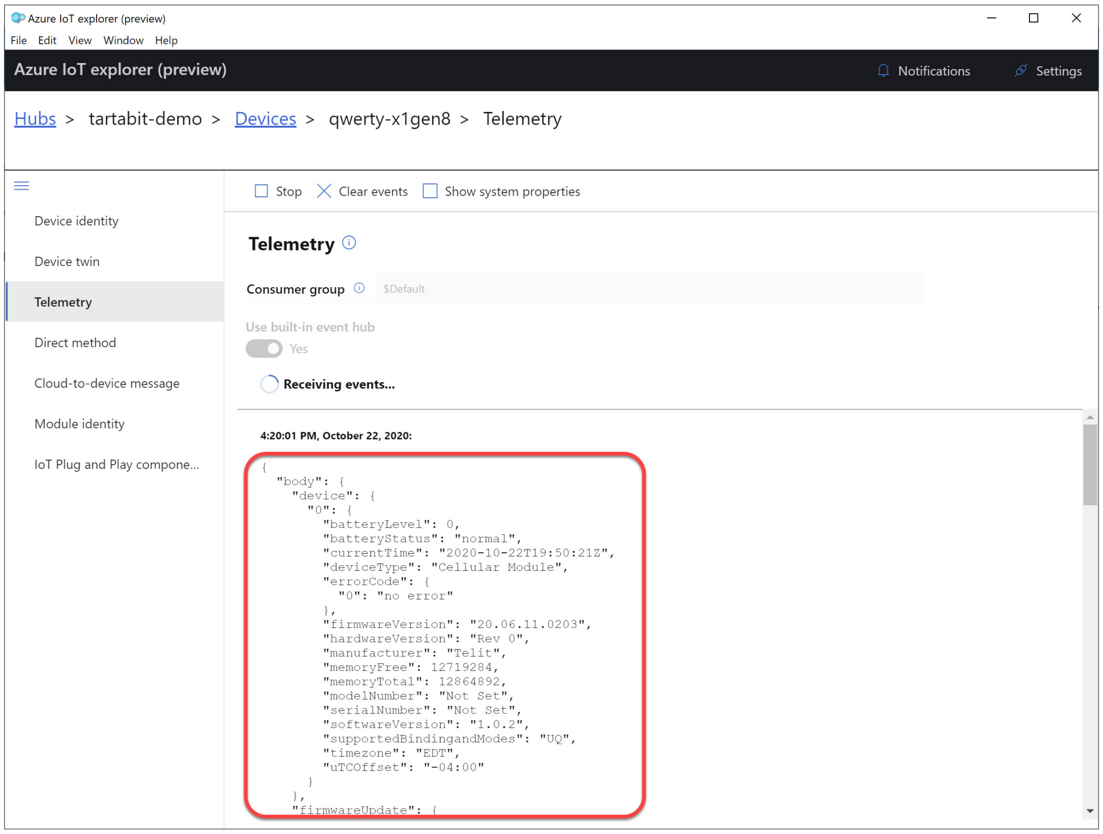
5. You will see the LWM2M object data in your twin.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > tartabit-demo > qwerty-x1gen8 >
Device twin
qwerty-x1gen8
Save
Refresh

The device twin for 'qwerty-x1gen8' is shown below. You can add tags and desired properties to your device twin here. To remove a tag or desired

```
  },
  "reported": {
    "device": {
      "0": {
        "batteryLevel": 0,
        "batteryStatus": "normal",
        "currentTime": "2020-10-08T15:12:19Z",
        "deviceType": "Cellular Module",
        "errorCode": {
          "0": "no error"
        },
        "firmwareVersion": "20.06.11.0203",
        "hardwareVersion": "Rev 0",
        "manufacturer": "Telit",
        "memoryFree": 12669152,
        "memoryTotal": 12864892,
        "modelNumber": "Not Set",
        "serialNumber": "Not Set",
        "softwareVersion": "1.0.2",
        "supportedBindingandModes": "UQ",
        "timezone": "EDT",
        "uTCOffset": "-04:00"
      }
    },
    "firmwareUpdate": {
      "0": {
        "firmwareUpdateDeliveryMethod": 2,
        "firmwareUpdateProtocolSupport": {
          "0": 0,
          "1": 1,
          "2": 2,
```

6. You can use the Azure IoT Explorer to view the telemetry as well.



4:20:01 PM, October 22, 2020:

```
{
  "body": {
    "device": {
      "0": {
        "batteryLevel": 0,
        "batteryStatus": "normal",
        "currentTime": "2020-10-22T19:50:21Z",
        "deviceType": "Cellular Module",
        "errorCode": {
          "0": "no error"
        },
        "firmwareVersion": "20.06.11.0203",
        "hardwareVersion": "Rev 0",
        "manufacturer": "Telit",
        "memoryFree": 12719284,
        "memoryTotal": 12864892,
        "modelNumber": "Not Set",
        "serialNumber": "Not Set",
        "softwareVersion": "1.0.2",
        "supportedBindingandModes": "UQ",
        "timezone": "EDT",
        "uTCOffset": "-04:00"
      }
    },
    "firmwareUpdate": {
```

You are done!

Powered by Wiki.js