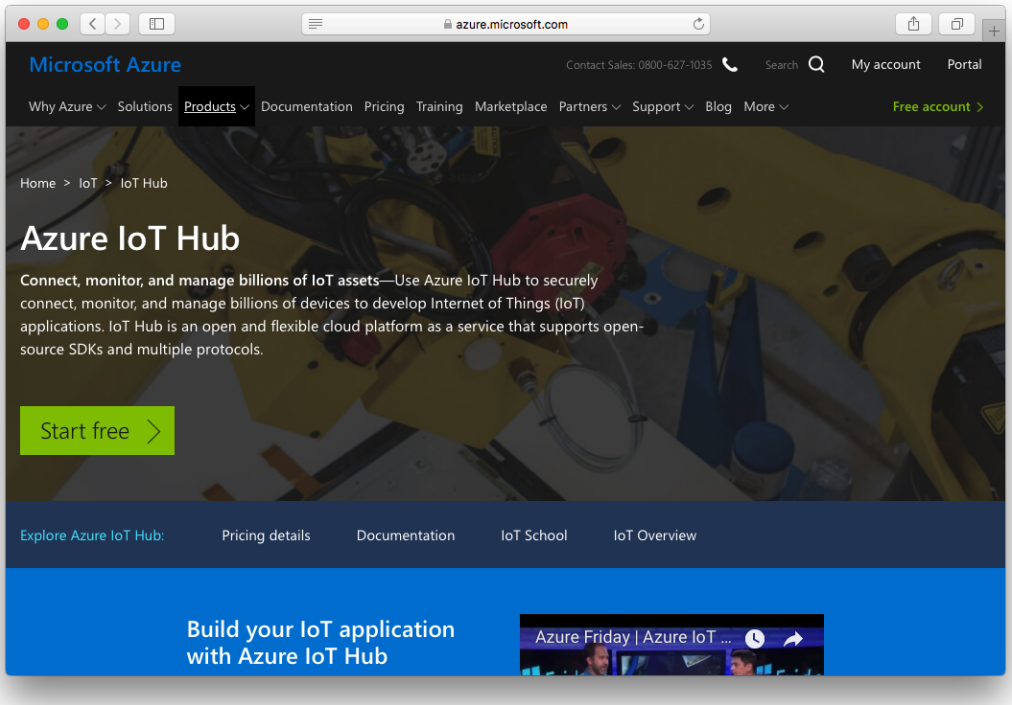


User Guide

Connecting your EpiSensor Gateway to Microsoft Azure IoT Hub

Applies to: NGR-30-3, NGR-30-5, ASAVIE-SKV1

EPI-103-01

Table of Contents

Introduction

EpiSensor's Internet of Things platform is easy to deploy, configure and scale and includes a range of sensor products that can monitor a variety of environmental and energy usage parameters in commercial and industrial environments.

This user guide contains technical information on how to connect your EpiSensor Gateway to Azure IoT Hub, the managed IoT Cloud platform from Microsoft. Azure IoT Hub can be used to securely manage the flow of data between your EpiSensor Gateway, and other Azure services for data storage, analysis and visualisation.

This guide requires a minimum of version V04.00.01.00.00 of EpiSensor Gateway software.

Related Documents

Related installation and configuration documents are listed in the following table:

Document	Reference No.
EpiSensor NGR-30-3 Datasheet	EPI-102-00
EpiSensor NGR-30-5 Datasheet	EPI-077-00
Gateway API User Guide	EPI-009-08
User Guide for NGR	EPI-075-00

Getting Started

There are currently two versions of the EpiSensor Gateway available (NGR-30-3 and NGR-30-5) based on different hardware platforms. The items included with each are as follows:

NGR-30-3		
	Qty	**Item**
	1	NGR-30-3 Gateway
	1	Mains Power Supply
	1	Ethernet Cable
	1	2.4GHz Antenna for ZigBee

NGR-30-5		
	**Qty**	**Item**
	1	NGR-30-5 Gateway
	1	2.4GHz Antenna for ZigBee
	1	Cellular Antenna
	1	WLAN Antenna
	1	Ethernet Cable

The NGR-30-5 Gateway (based on the Dell Edge Gateway 3002) requires an external 12/24V power supply that is not included as standard, unless you're using one of our starter/accelerator kits. For more information on Dell Edge Gateway 3002 hardware, click here to access the user manual and here for a spec sheet.

You will need an EpiSensor Gateway with an Internet connection (Cellular, Ethernet or Wi-Fi) with TCP port 8883 open, and at least one EpiSensor wireless sensor to get data flowing to Azure IoT Hub.

Configure Azure IoT Hub

Azure IoT Hub is a managed cloud platform that makes it easy to connect, monitor, authenticate, and automate IoT devices. Azure IoT Hub can support billions of connected devices and can process and route those messages to other Microsoft Azure services for storage, analysis and visualisation.

Data can be transferred to Azure IoT Hub in a number of ways, but in this guide we'll focus on sending data in JSON format via MQTT (Message Queuing Telemetry Transport), which is an efficient publish-subscribe-based messaging protocol optonised for high-latency, low-bandwidth networks connections.

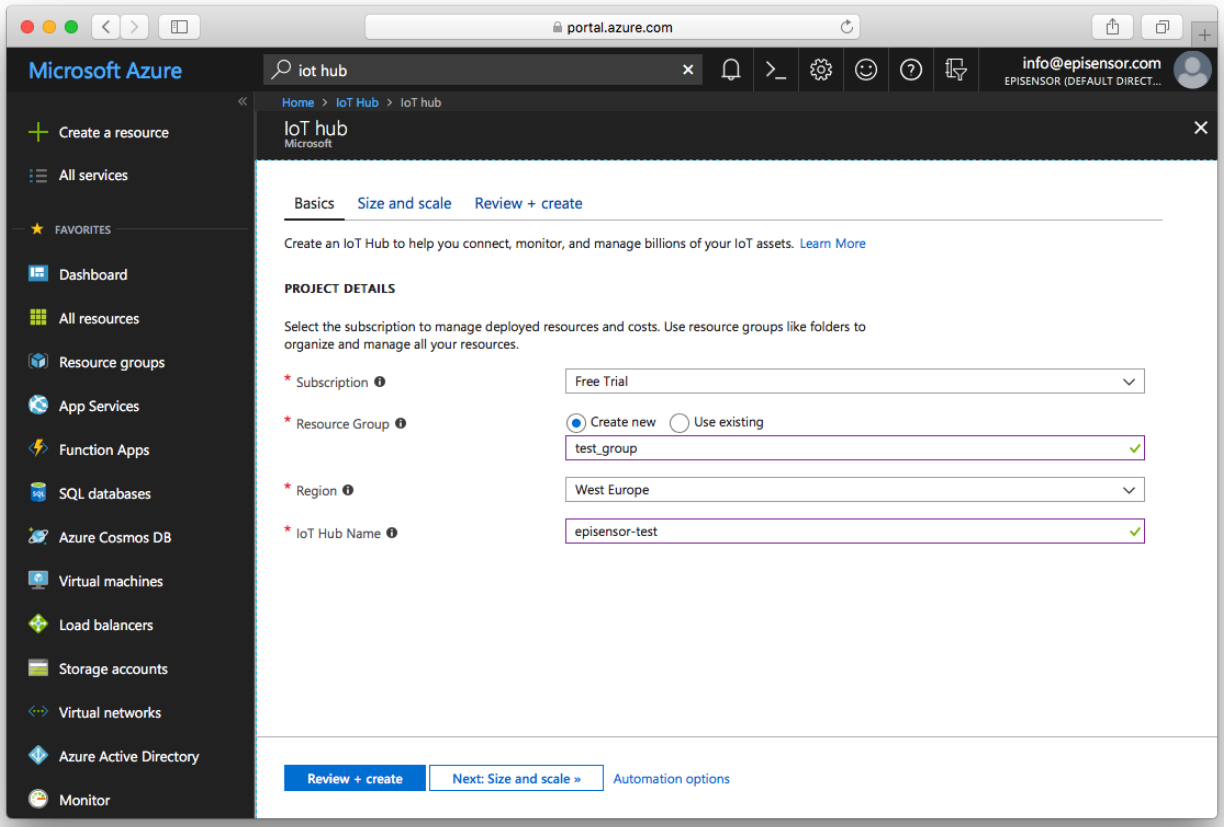
Log in to Microsoft Azure

To get started, sign up for an Azure account or log in to your existing account at the following link: https://portal.azure.com/

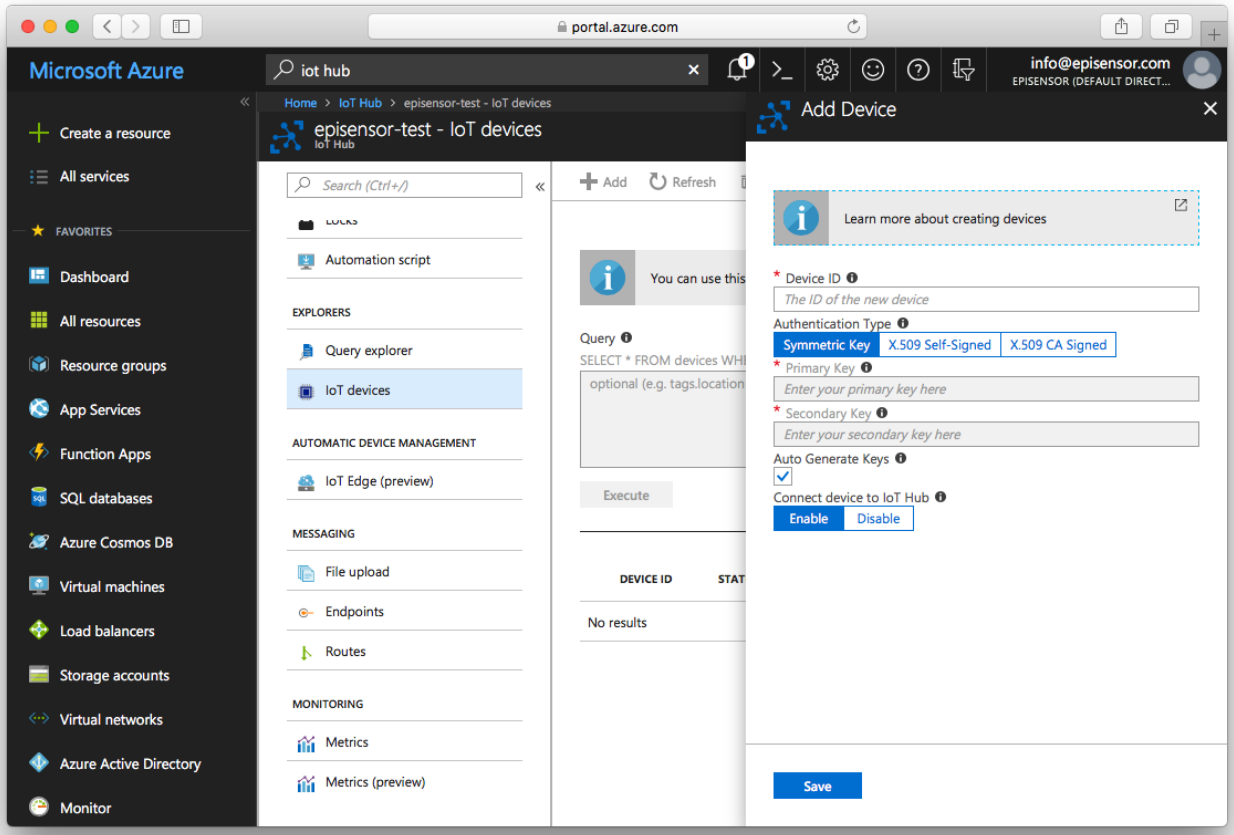
We'll first need to configure create a new IoT Hub instance that can accept connections from our EpiSensor Gateway.

Create a new IoT Hub

From your Microsoft Azure account dashboard, create a new Resource of type 'IoT Hub'. Using the Free tier is fine for this demonstration, and it can be upgraded to a higher capacity instance in the future. Take note of the hub name you choose.



The using the IoT Devices Explorers tab, add a new device to the IoT hub, and again take note of the device name you choose. Choose the Symmetric Key Authentication type as shown in the screenshot below.



Generating the SAS Token

Generate a SAS token with the following procedure:

https://docs.microsoft.com/en-us/rest/api/eventhub/generate-sas-token

Configure your EpiSensor Gateway

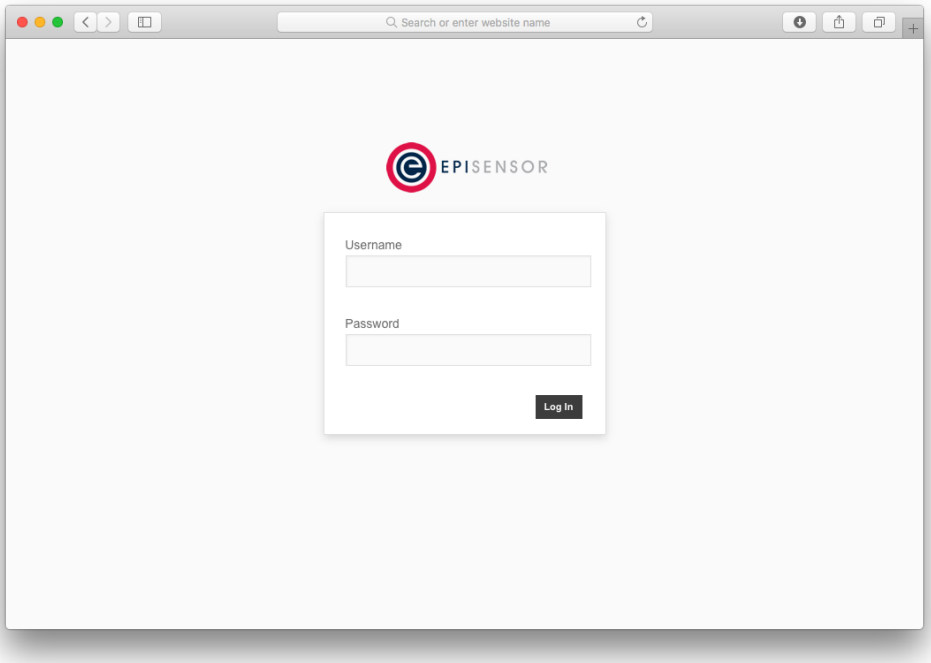
In this section, we'll join a node to the EpiSensor Gateway, configure it to 'export' data, and upload the Azure IoT Hub security certificates so a secure MQTT connection can be established.

Logging in

To log in to your EpiSensor Gateway, first check if the SKU of your hardware is NGR-30-3 or NGR-30-5, as the procedure is slightly different depending on the model.

Important Note
The factory default IP address of the NGR-30-3 Gateway is 172.31.255.1

It can take up to 5 minutes for the Gateway initialise. After this time, go to http://172.31.255.1:8081/ in a modern browser and you should see the Gateway login interface below.



The Gateway supports all recent versions of Internet Explorer (IE 9 +), Chrome, Firefox and Safari web browsers. On older browsers, some features may not display correctly.

Important Note
The default Gateway user account is Administrator; the default login details for this account are as follows: **Username:** Administrator **Password:** A1

If an incorrect password is entered more than four time, users will be locked out for five minutes. After five minutes has elapsed you can try again, up to another five attempts are allowed, and so on. In the event of the password being irretrievable, please contact support@episensor.com

Joining a Node

This section of the guide covers the four steps that are required to connect a node to the Gateway and get data flowing. This assumes that you are using a factory default EpiSensor Gateway, not a kit or a system that has already been in use.

Forming a Network

The ZigBee network on the Gateway should be 'Formed' once. This will force it to scan for the best wireless channel to operate on, and configure unique encryption keys so the sensors can communicate securely with the Gateway.

To Form a new network, go to Settings → System → Form New Network and click "Submit".

Important Note	
	"Form New Network" should only be done once for every Gateway. Sending this command on a Gateway that already has nodes joined will erase the security keys, and all of the nodes on that Gateway will need to be factory reset and rejoined.

After about 30 seconds, the scan will have completed, and the Status field on the Home page should show "OK".

Enable 'Allow Join' Mode

Next, we'll temporarily enable "Allow Join" mode which tells the Gateway to permit new nodes to connect to the Gateway. To do this, go to Settings → Add Nodes, and enable this mode for 15 minutes.

Waking up your Nodes

If you are joining a battery powered node to the Gateway (assuming it's in factory default condition) you'll need to wake it up from deep sleep mode by pressing and holding the Mode button until the status LED flickers, and then release.

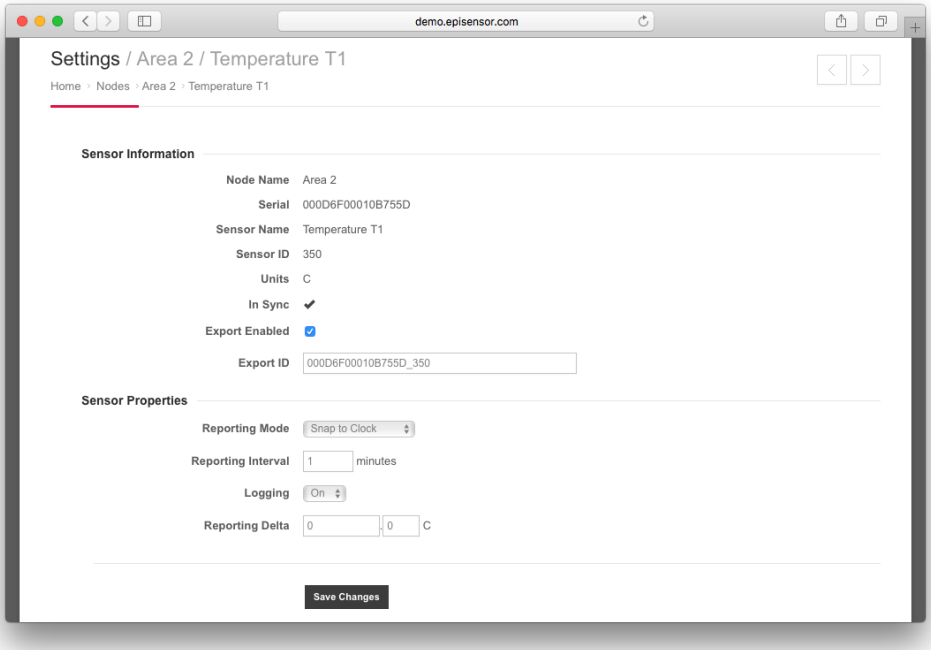
If you're joining a mains-powered node, it should automatically join if it's in factory default condition. To factory reset a node, refer to the node's user manual.

Enable data export

When you have joined a node to your Gateway, it will need a couple of minutes to synchronise its settings. On the 'Nodes' page, click on Action → Settings. Towards the bottom of this page, you should see a list of the 'sensors' available on the node.

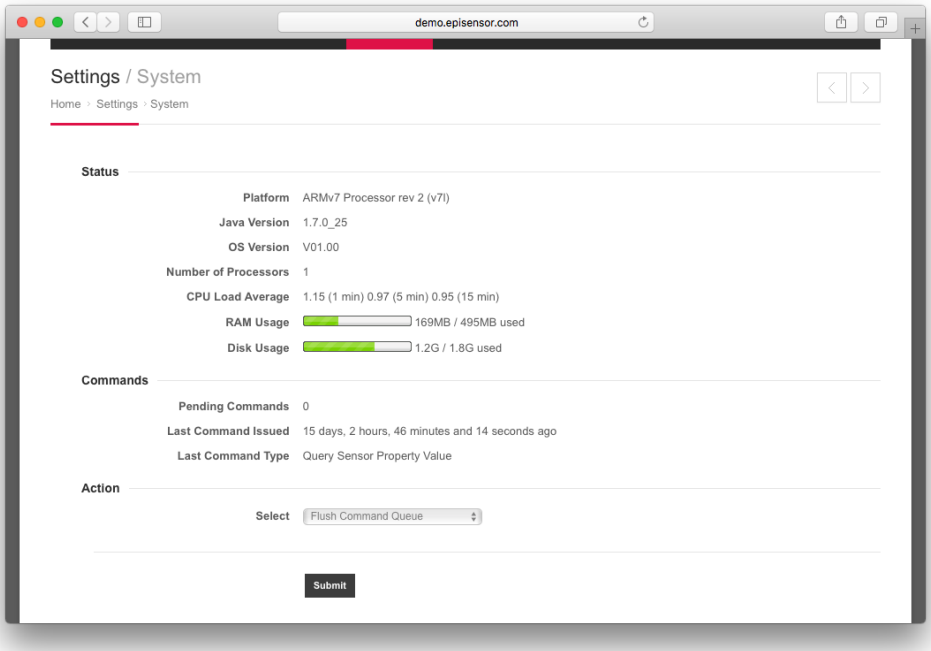
Select one that you are interested in, and click Action → Settings to go a level deeper into the settings of that particular sensor.

Make sure “Export Enabled” is checked, set the ‘Reporting Mode’ to “Snap to Clock”, then click ‘Save Changes’. This will tell the Gateway that data from this sensor should be sent to Azure IoT Hub, as opposed to just being shown on the Gateway’s web interface.

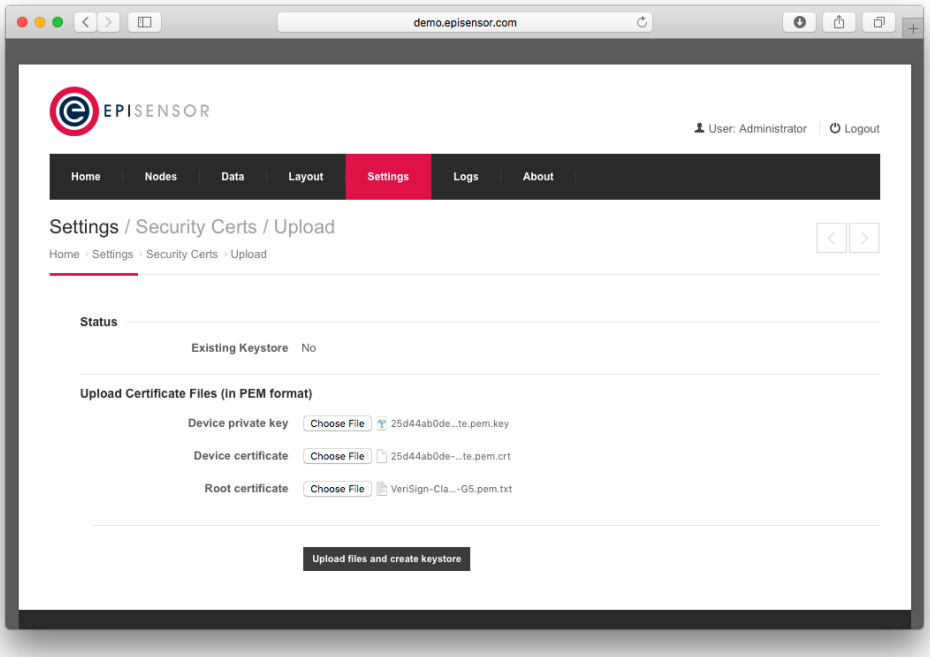


Deleting the existing security keystore

For connection to the Azure IoT Hub, a Shared Access Signature (SAS) token is used instead of a certificate. If there is already a keystore on the Gateway, it will need to be deleted from the EpiSensor Gateway before it can connect to Azure IoT Hub. This can be done from the Settings → System page as shown in the following screenshot.



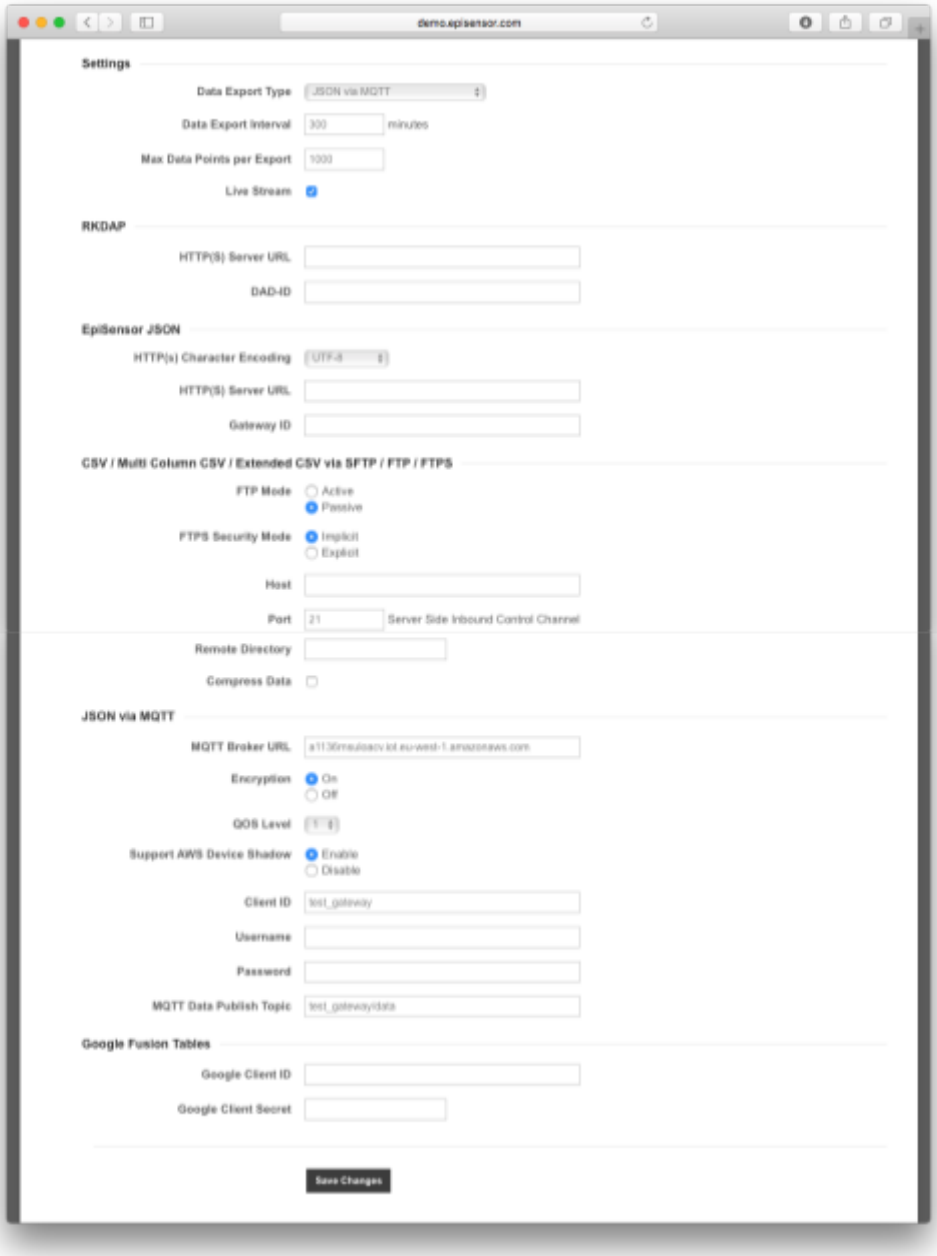
Once the keystore has been deleted, the Status on the Settings → Security Certs → Upload page should reflect that as shown in the following screenshot.



Enable MQTT Data Export

This section describes the settings required on the Settings → Data Export page of the EpiSensor Gateway to connect to Azure IoT Hub.

There are a number of different data export mechanisms available on the EpiSensor Gateway and each has different configuration options as shown on the Data Export page.



These are described in the following table:

Data Export General Settings	Description
Data Export Type	This drop down list shows all the supported Data Export Formats and Transport Mechanisms. For exporting to Azure IoT Hub, select "JSON via MQTT" on the data export page as shown in the following screenshot.
Data Export Interval	The 'Data Export Interval' field specifies the maximum time interval (in minutes) between attempted exports. The EpiSensor Gateway will export when either the 'Max Data Points per Export' has been reached or the 'Data Export Interval' has elapsed.
Max Data Points per Export	The 'Max Data Points per Export' field specifies the maximum number of discrete data points that will be accumulated on the Gateway before exporting. The EpiSensor Gateway will export when either the 'Max Data Points per Export' has been reached or the 'Data Export Interval' has elapsed.
Live Stream	Live Stream export means data points will be exported as soon as possible. If 'Live Stream' export is selected, both the 'Data Export Interval' and 'Max Data points per Export' settings are overridden. In other words the EpiSensor Gateway will export data as soon as it has arrived and been processed by the EpiSensor Gateway. This can be useful when data is required to be as real-time as is possible. However, it can also result in inefficient use of the data transport mechanism because the Gateway will be exporting smaller amounts of data more frequently.

In addition, there are configuration options specifically required for MQTT via JSON export type as described in the following table:

MQTT via JSON Settings	Description
MQTT Broker URL	This should be configured with custom IoT Hub host name which will be of the format {iothubname}.azure-devices.net where {iothubname} is the name of your IoT hub.
Encryption	Encryption must be enabled (On) for connection to Azure IoT Hub. Azure IoT Hub does not support unencrypted connections. For this configuration, port 8883 should be open in any firewall between the EpiSensor Gateway and the Internet.
QOS Level	Quality of Service Level for published data. Level 0, 1 or 2 may be selected. Azure IoT Hub supports QoS values of 0 or 1.
Support AWS Device Shadow	Not applicable for Azure IoT Hub, should be disabled.

Client ID	The name of your IoT Hub device should be entered in this field.
Username	The username when client authentication is to be used. This field is required for connection to Azure IoT Hub. For the username field use {iothubhostname}/{deviceId}/api-version=2016-11-14. The iothushostname is the custom IoT Hub host name (same as used in the MQTT Broker URL field above). The deviceId is the name of your IoT Hub device (same as used in the Client ID field above).
Password	This field is required for connection to Azure IoT Hub. It should be set to the Shared Access Signature part of the SAS token generated. For example if the SAS token is: HostName={iothubhostname};DeviceId={deviceId};SharedAccessSignature=SharedAccessSignature sr=%2Fdevices%2F{deviceId}%2Fapi-version%3D2016-11-14&sig=vSgHBMUG.....Ntg%3d&se=1456481802 Then the password field should be set to: SharedAccessSignature sr=%2Fdevices%2F{deviceId}%2Fapi-version%3D2016-11-14 &sig=vSgHBMUG.....Ntg%3d&se=1456481802
MQTT Data Publish Topic	The data public topic should be set to devices/{deviceID}/messages/events/ where deviceId is the name of your IoT Hub device (same as used in the Client ID field above)

If for any reason the export of data fails (for example lost internet connection or expired security certificates), data which failed to export will be saved on the Gateway. The export of this data will be retried based on the number of minutes defined in the 'Data Export Interval' field.

Files queued for export can be deleted from the system using the drop down list on the Settings → System page.

Testing and Next Steps

The "Last Data Export" field on the Settings → Data Export page of the EpiSensor Gateway will show how long ago data was last sent to the server.

If you refresh the page and this field shows a recent data export (i.e. less than the data export interval you defined) then the connection to Azure IoT Hub has been established!

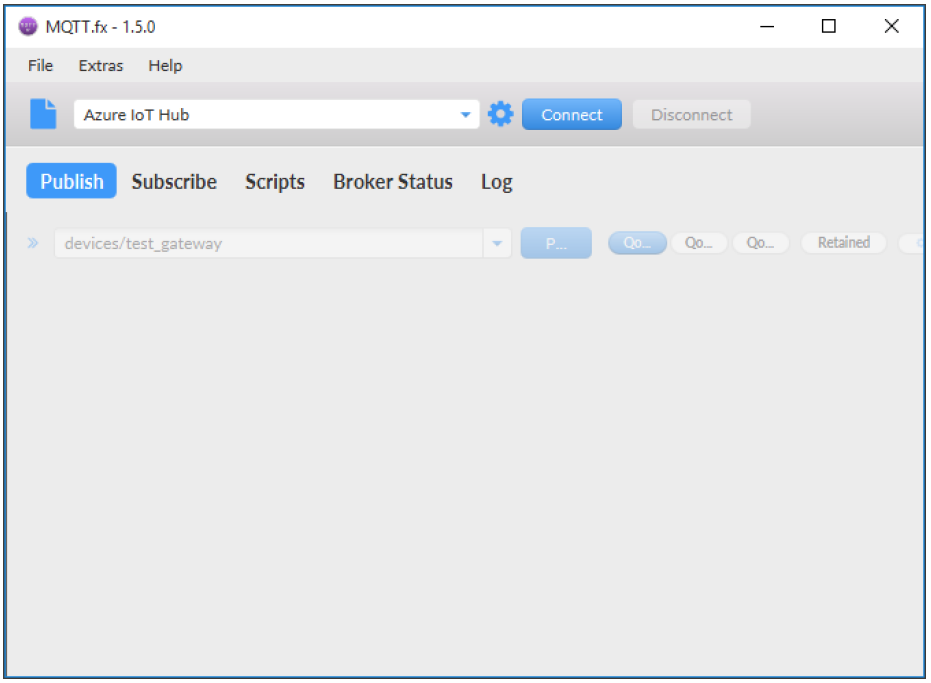
Depending on how the reporting intervals of the sensors are configured and the data export interval, there may be a delay in data being sent to Azure IoT Hub.

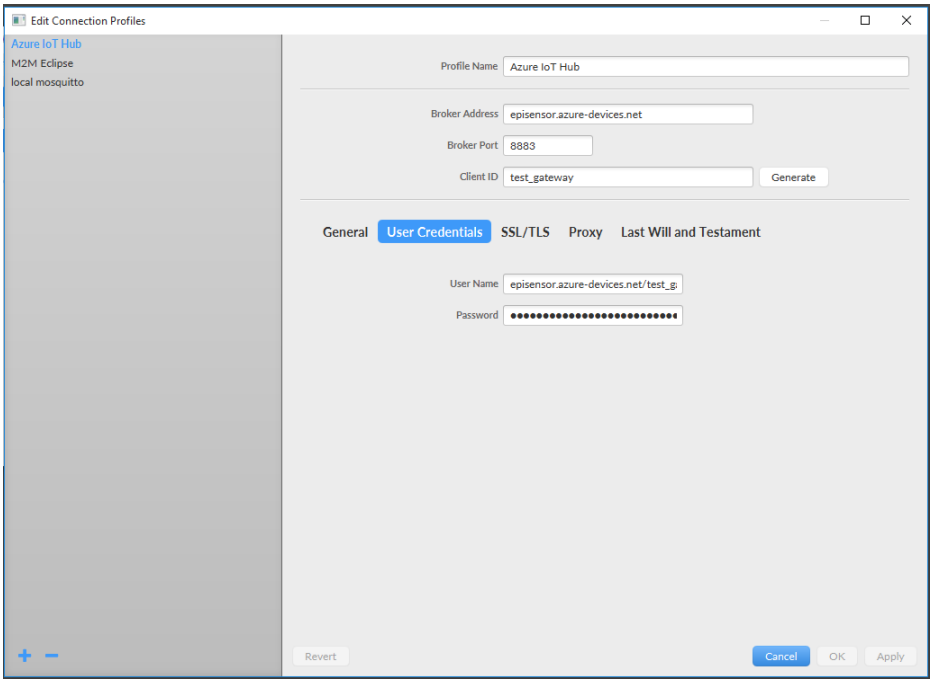
Testing the Connection from Azure IoT Hub

To test the success of the connection to Azure Hub IoT, a third-party MQTT client like MQTT.fx can be used to connect to the Azure IoT hub and subscribed to the topics that the EpiSensor Gateway publishes to. MQTT.fx can be downloaded from the following link:

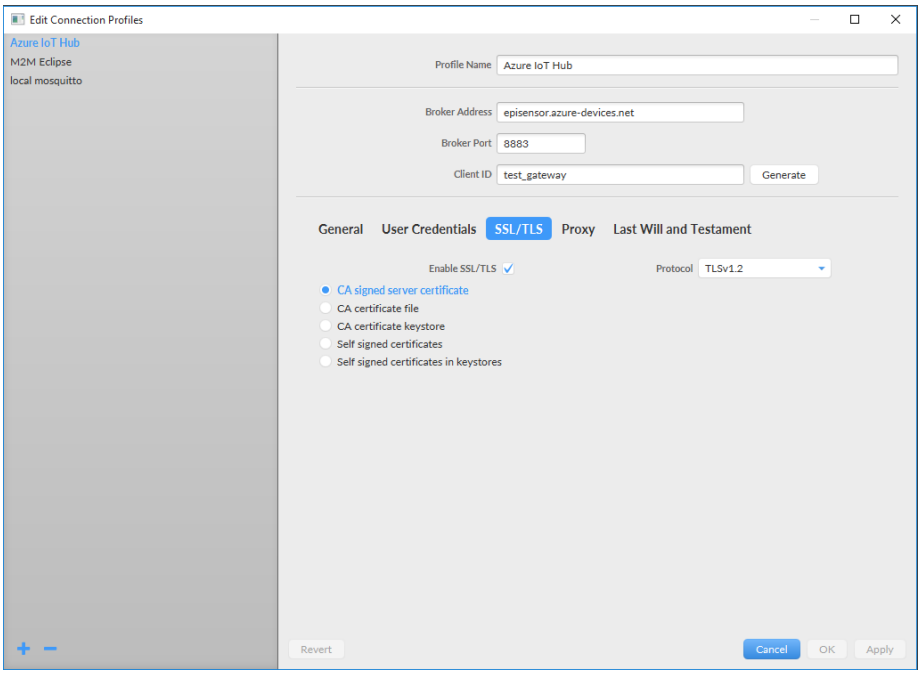
http://mqttfx.jensd.de/index.php/download

The connection profile for the connection to Azure Hub IoT should use the same settings for Client ID, Username and Password as specified on the Settings → Data Export page on the EpiSensor Gateway. In addition, the Broker Port should be set to 8883 and the Broker Address should be the same as the Broker URL set on that page.





The CA signed server certificate option should be ticked on the SSL/TLS settings list.



After the connection has been successfully established, the MQTT client should subscribe to the **devices/{deviceID}/messages/events/** topic to see incoming data from the EpiSensor Gateway.

Working with the sensor data

Data from sensors of the nodes joined to the EpiSensor Gateway is published to Azure IoT Hub in the following format:


```
{
  "gateway": "000D6F00027FE565",
  "timestamp": 1525770720,
  "values": {
    "000D6F00010B755D": {
      "350": "20.1"
    },
    "000D6F0001A310AE": {
      "305": "0.2",
      "307": "237.9",
      "308": "237.2",
      "309": "177.86",
      "324": "145.0",
      "328": "150.0",
      "335": "0.266"
    },
    "000D6F0001A31E69": {
      "505": "230.0"
    }
  }
}
```


In the example above, the Serial Number of the EpiSensor Gateway publishing the data is 000D6F00027FE565.

The timestamp of the received data is 1525770720 (seconds since the Epoch). The data is then grouped per node connected to the EpiSensor Gateway. In this example, we have one data point from a node with serial number “000D6F00010B755D” at that timestamp from a sensor with ID = 350, and the value of the data point is 20.1.

Ordering Information

EpiSensor products are available to order directly or via EpiSensor's distribution partners. The following table lists the available Gateways and Starter Kits that are compatible with Azure IoT Hub.

SKU	Description
NGR-30-3	Ethernet communications, incl. 1yr EpiSensor Gateway software license, up to 50 nodes/200 sensors
NGR-30-5	Dell Edge Gateway 3002, incl. 1yr EpiSensor Gateway software license, up to 100 nodes/1000 sensors
ASAVIE-SKV1	Industrial IoT Accelerator Kit (in partnership with Asavie and Dell), incl. Dell Edge Gateway 3002 with EpiSensor Gateway software, 2x wireless temperature sensors

Troubleshooting & Support

If you are experiencing problems with your NGR Gateway or any other part of your EpiSensor system, or you notice something unusual - please contact EpiSensor support at the following email address, phone number or via live chat on our website.

- Email: support@episensor.com
- Tel: +353 61 512 500
- Website: http://episensor.com

For customers and partners who are deploying systems in business-critical environments, there are a number of support packages available that offer a higher level of service and response time. For more information on EpiSensor Premium Support, visit: http://episensor.com/premium-support/

Warranty

All EpiSensor products and provided with a 365 day limited warranty effective from the shipping/invoice date of an order. During the warranty period, under the conditions of normal use, EpiSensor will repair or replace any product that has a manufacturing defect.

Warranty can be extended by up to 4 years within 30 days of a purchase. For more information on warranty, visit: http://episensor.com/warranty/

Glossary

Definitions for terms and abbreviations used in this document are listed in the following table:

Term	Description
Allow join mode	A mode that can be enabled on the Gateway that allows new wireless nodes to join
Azure	Cloud platform from Microsoft
Gateway	The central computer that managed the EpiSensor system
Interval and Delta	Reporting mode where data is produced when the reporting interval has elapsed, unless a change is detected
JKS	Java Keystore
MQTT	MQTT (Message Queuing Telemetry Transport) is an ISO standard publish-subscribe-based messaging protocol
Node	Used to describe a physical EpiSensor product
Reporting Interval	The length of time between each data point produced by a node
Reporting Mode	Defines how an EpiSensor node should report data to the Gateway
SAS	Shared Access Signature
Sensor	Describes a feed of data within the EpiSensor system
Snap to Clock	Reporting mode where data is 'snapped' to the nearest 1 minute / 5 minute / 15 minute interval etc.
WSN	Wireless Sensor Network
ZigBee	IEEE 802.15.4 Wireless communications standard that EpiSensor nodes use.