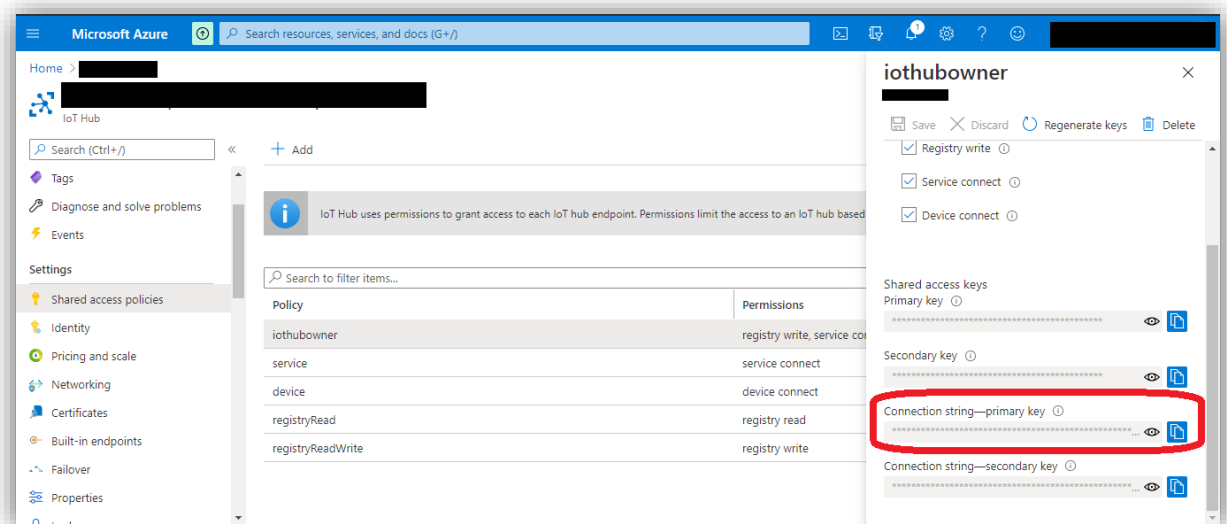


DEVICE EXPLORER

Device Explorer which useful to monitor and do some basic operations in Azure IoT hub like publish and subscribe via mqtt client.

Its releases can be found here : <https://github.com/Azure/azure-iot-sdk-csharp/releases?after=2017-8-10>

This tool runs on your machine and connects to IoT hub in Azure. So you need to get the connection details which tool would use to connect to your hub. For that we need to go back to Azure portal and copy the connection string (primary) value for iothubowner (not from device explorer) from Shared access policies in Azure IoT hub.



We need to initiate the connection to Azure IoT hub firstly so we need this connection string to insert into in the device explorer window.

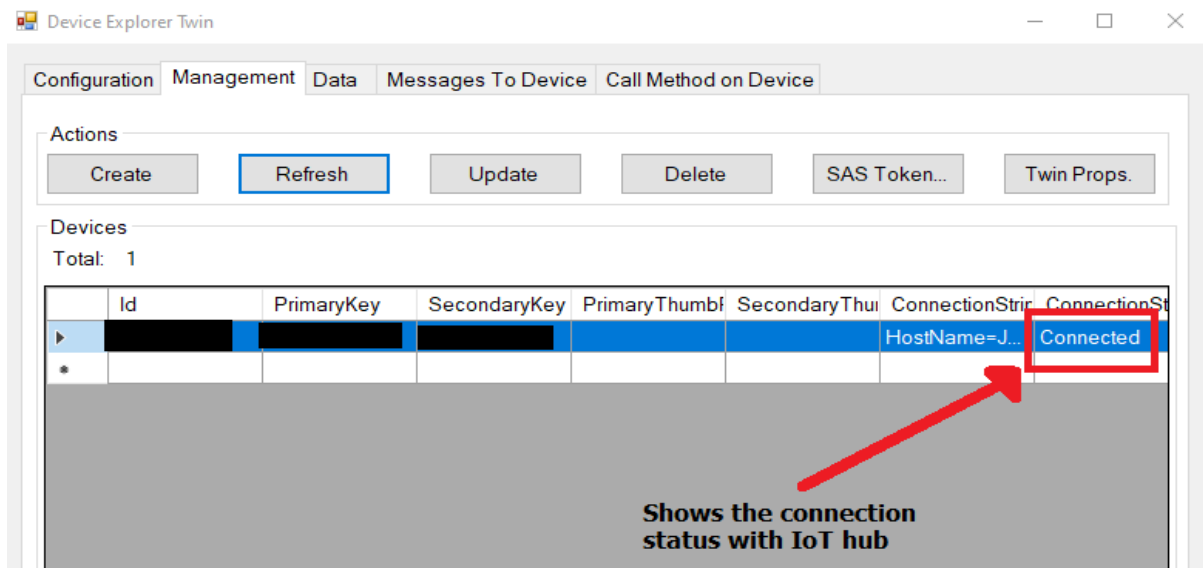
Then paste the **connection string** on the slot and press **update** button for initiating connection.

The screenshot shows the 'Configuration' tab of the 'Device Explorer Twin' application. Under the 'Connection Information' section, the 'IoT Hub Connection String' is displayed as 'HostName=[redacted].azure-devices.net,SharedAccessKeyName=iothubowner,SharedAccessKey=[redacted]'. Below this, the 'Protocol Gateway HostName' field is empty. A red rectangle highlights the 'Update' button. In the 'Shared Access Signature' section, the 'Key Name' is 'iothubowner', the 'Key Value' is a long string of characters, and the 'Target' is '[redacted].azure-devices.net'. The 'TTL (Days)' is set to 365. A 'Generate SAS' button is also visible.

Once Connection String is updated, we can see the key name, value and target will automatically updated based on your iot hub credentials. Next go to the **data** tab on top and click it and press monitor to see the **publishing** message in real time from ESP8266 on **Event Hub Data** window.

The screenshot shows the 'Data' tab of the 'Device Explorer Twin' application. The 'Monitoring' section contains fields for 'Event Hub', 'Device ID', 'Start Time' (set to 10/04/2020 14:39:18), and 'Consumer Group' (set to \$Default). A red arrow points to the 'Consumer Group' field with the text 'Click here'. Below these fields, a red rectangle highlights the 'Monitor' button. There are also 'Cancel' and 'Clear' buttons, and a checkbox for 'Show system properties'. The 'Event Hub Data' section at the bottom displays two JSON messages: one from device JJESP8266 with temperature 25 and humidity 33, and another with temperature 27 and humidity 38.

We can ensure the connection status of device by clicking in the **Management** button and if it connected its shows **connected** otherwise **disconnected**.



We can send message from device explorer as like in iot hub **message to device** method. Here also click on **message to device** tab and type a message as you like and click **send** button and listen in your ESP8266 to see it.

