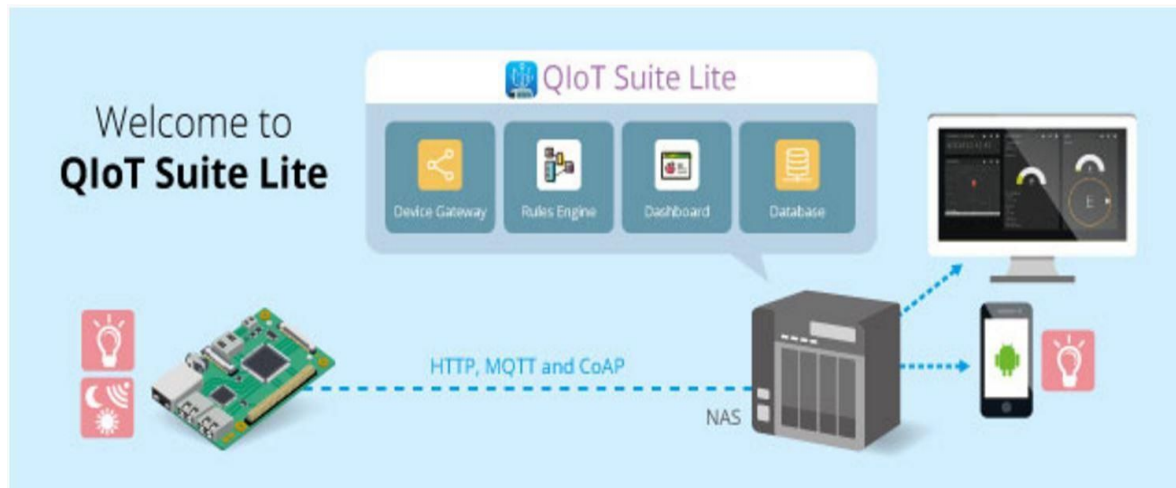


Get started with Arduino Yun (Python)

In this tutorial, you begin by learning the basics of working with Arduino Yun that's running OpenWrt which is a free operating system based on Linux for the Arduino Yun. You then learn how to seamlessly connect your devices to QNAP NAS by using QIoT Suite Lite. Please ensure your Arduino Yun and NAS is under the same LAN



Lesson 1: Configure your device

In this lesson, you configure your Arduino Yun device with an operating system, set up your development environment, and deploy an application to Arduino Yun.

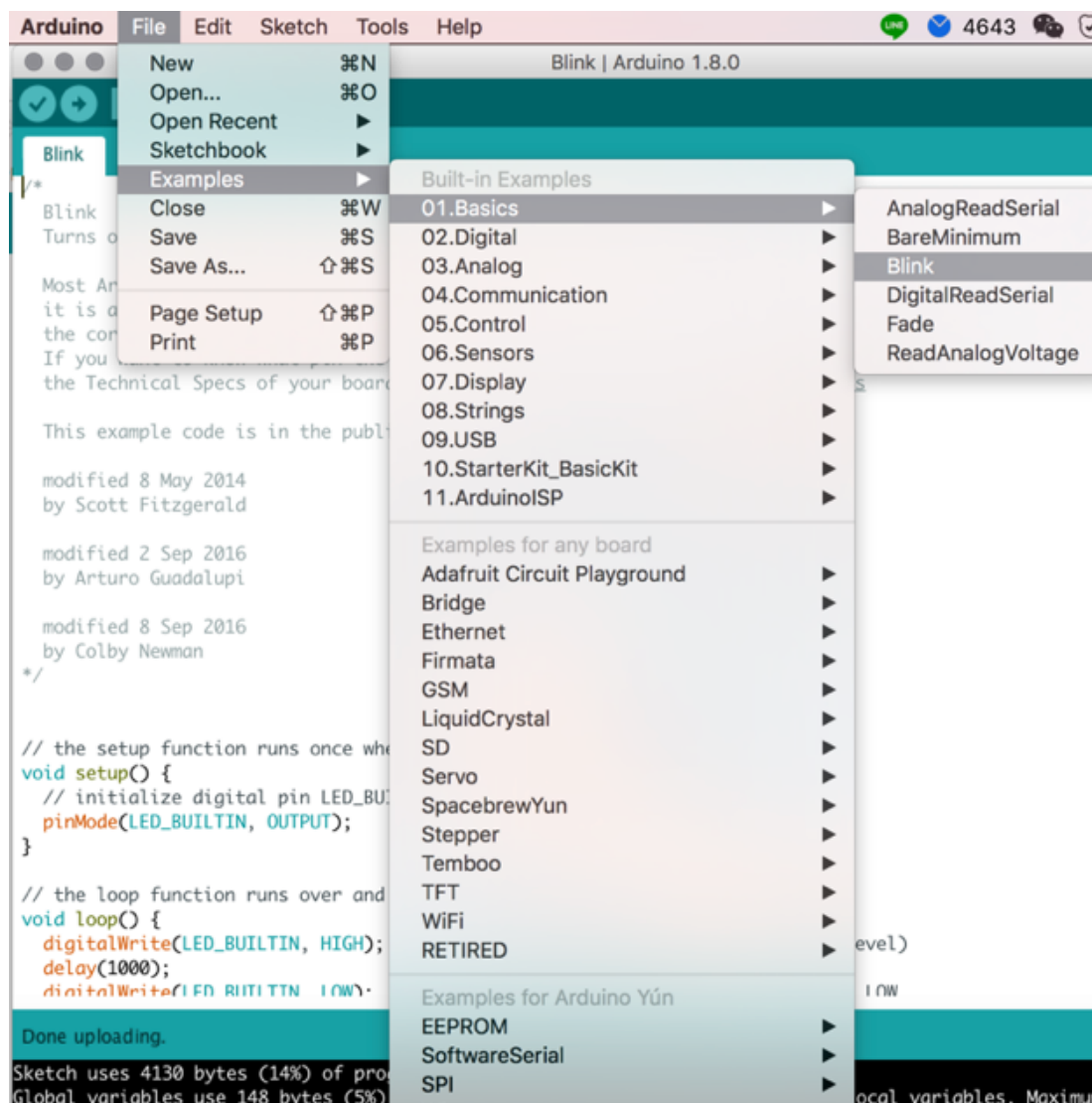
1.1 Download Arduino IDE

- Download and install Arduino IDE based on your host PC, available here: <https://www.arduino.cc/en/Main/Software>
- More Arduino tutorial: <https://www.arduino.cc/en/Tutorial/HomePage>

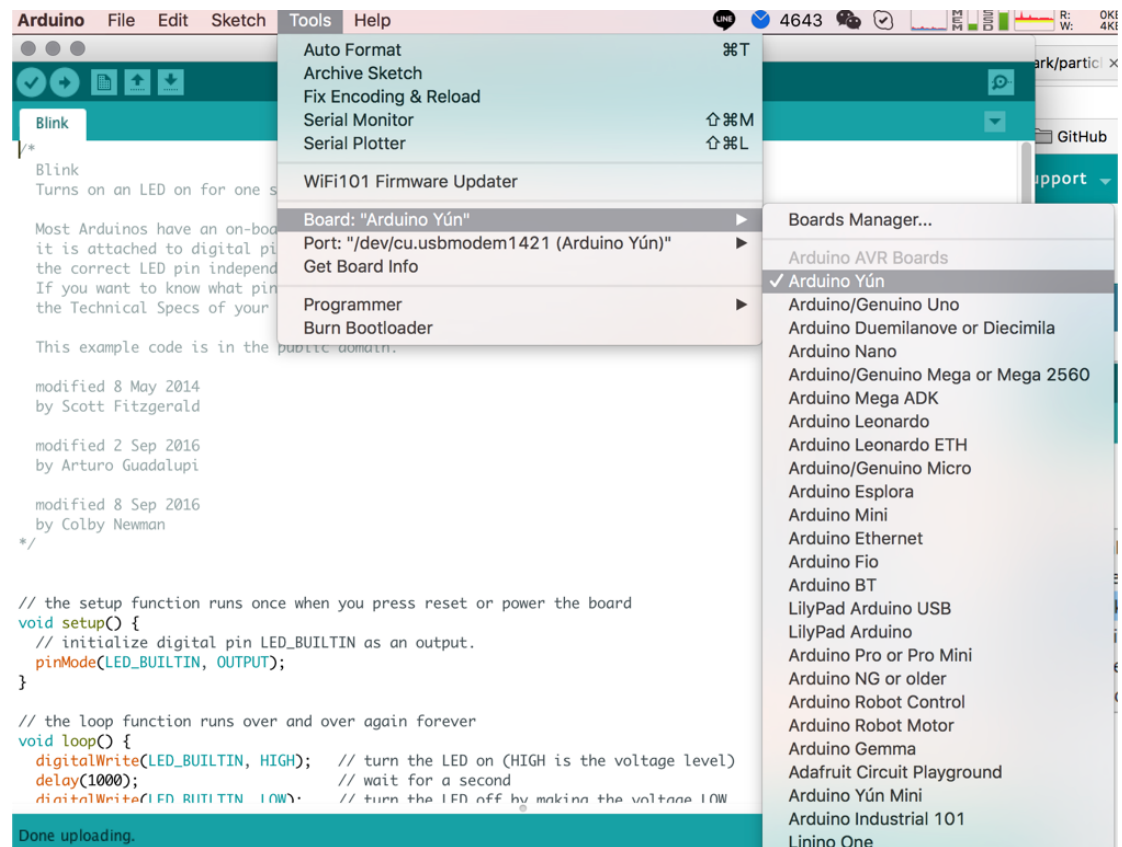
1.2 Configure and test your device

If this is the first time you use your Arduino Yun, you will have to follow some steps to assemble it.

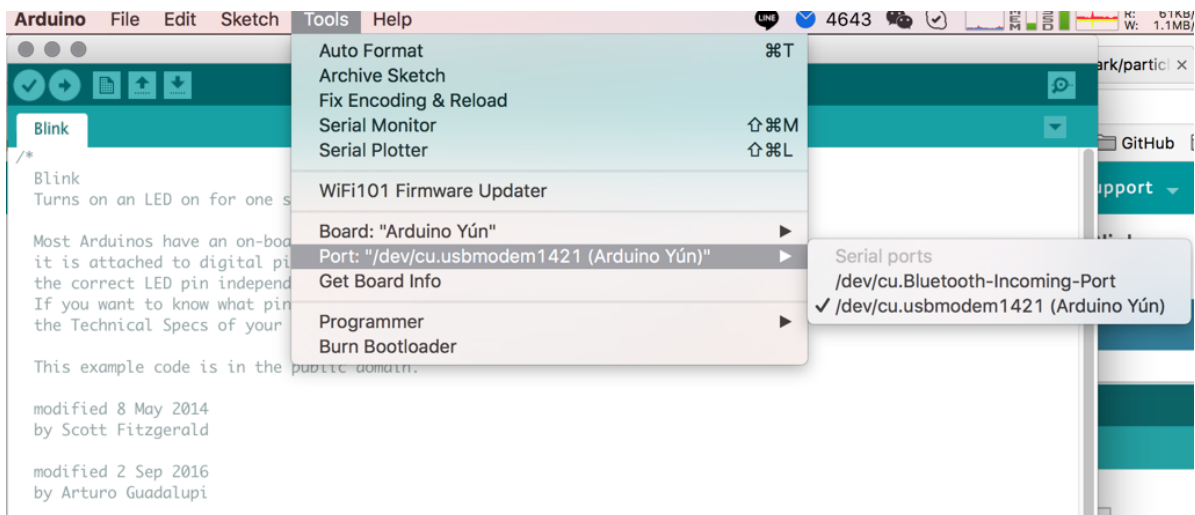
- Connect Arduino Yun to your PC with USB.
- Open your Arduino IDE and find Blink example.



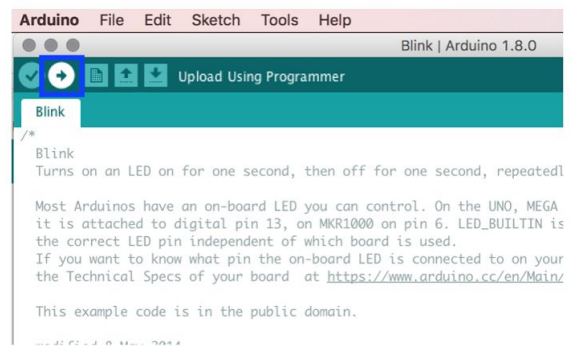
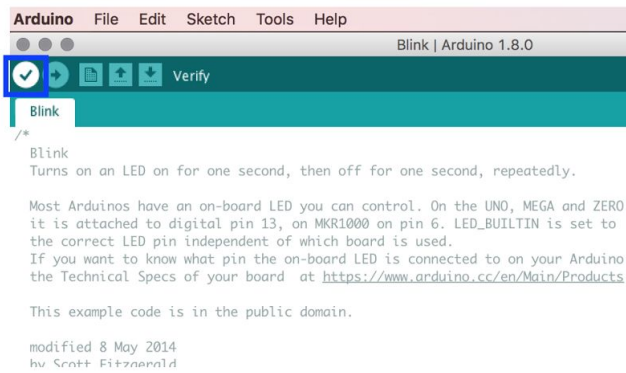
- Configure your board:



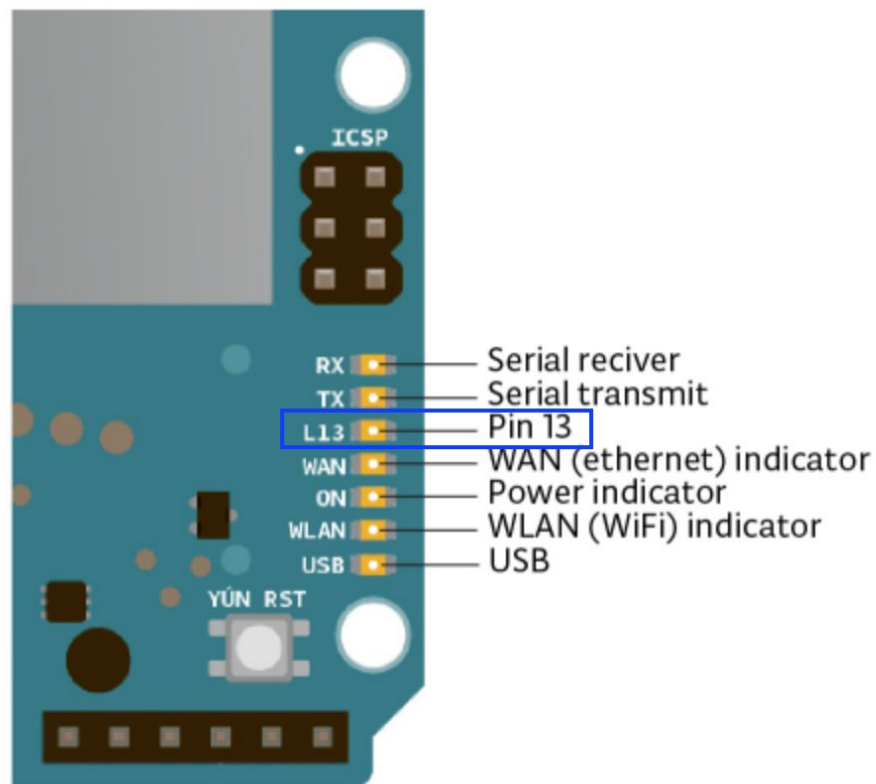
- Configure your port:



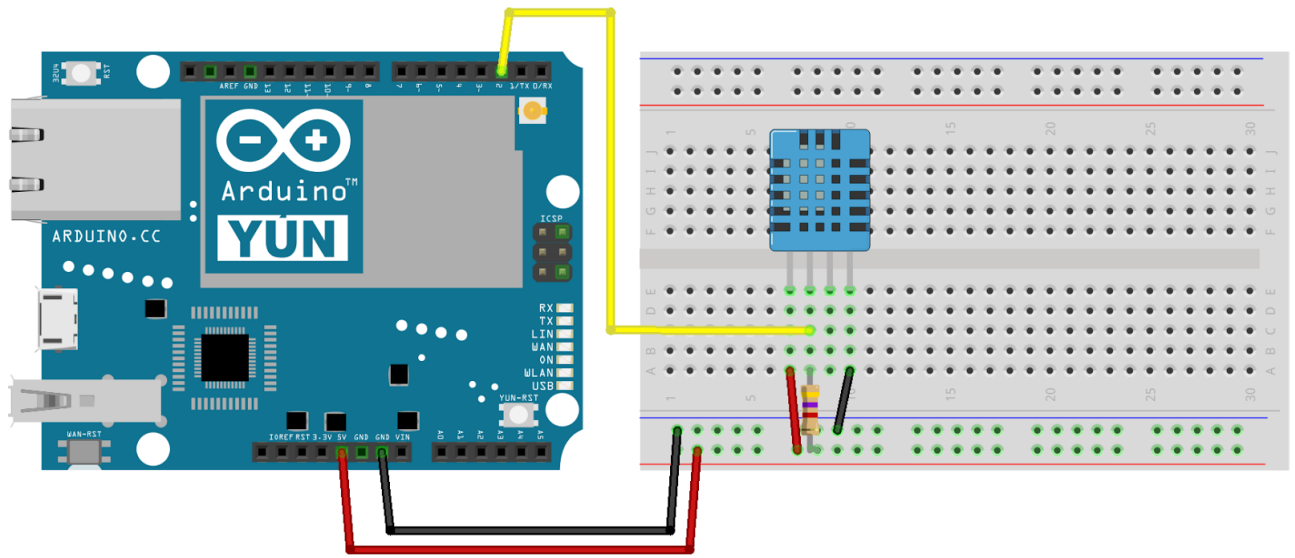
- Verify and upload your Blink example to Arduino Yun



- If you success upload your code to Arduino Yun, please check your Arduino Yun and you could see L13 is blink now:



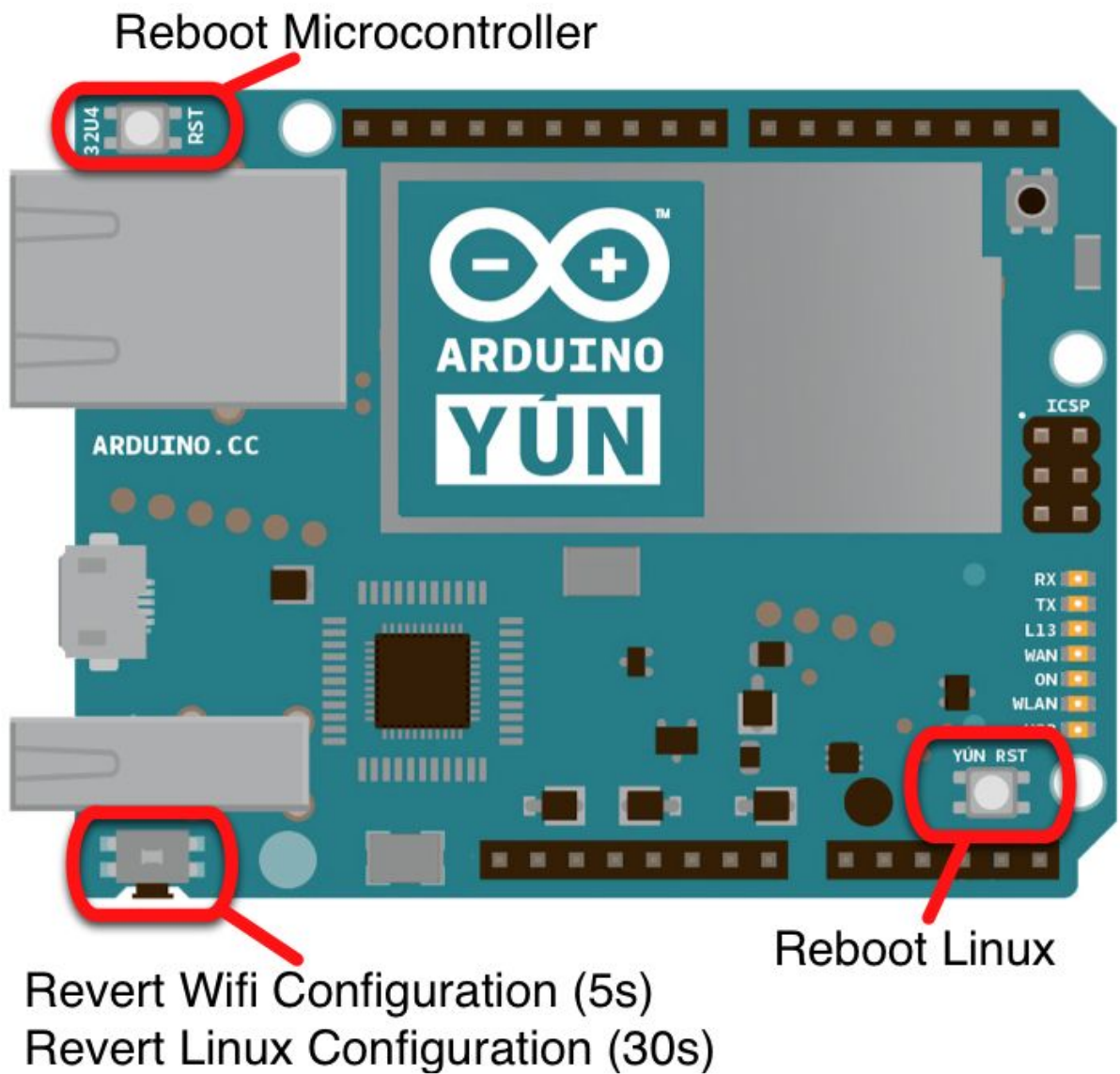
- Connect your DHT11 sensor to **“Ground”**, **“5V”**, **“Pin 2(Digital)”** on Arduino Yun respectively.



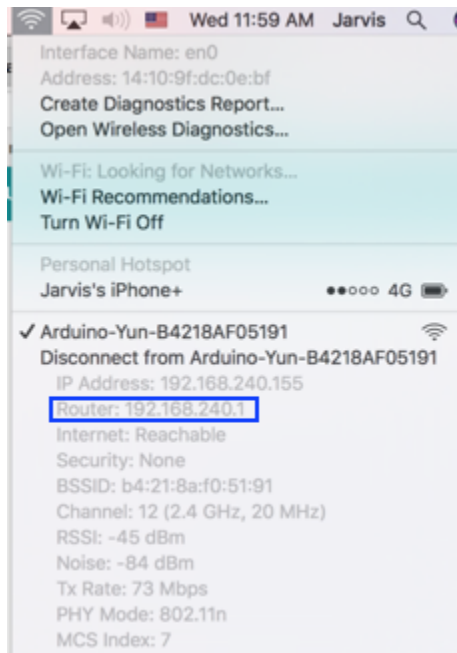
fritzing

1.3 Get IP address of your Arduino Yun

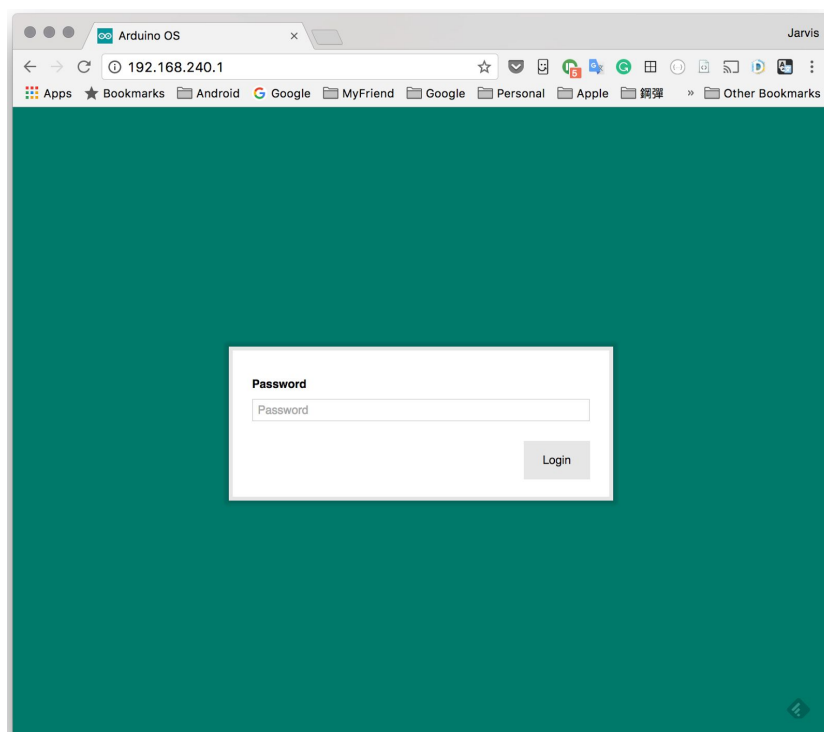
- Press Wifi reset button (need to press 5 seconds)



- Go to your Wifi settings and choose “Arduino-Yun-XXXXXXX”. The “XXXXXXX” is MAC address of your Arduino Yun.



- Open your browser and type in “<http://192.168.240.1>” or “<http://arduino.local>”. The default password is “arduino”.



- Complete your Board setting and link your Arduino Yun to the router you are using. Skip “API setting” and click on save to complete.



The screenshot shows the 'Board Settings' screen of the Arduino Configuration Wizard. The title bar reads 'Arduino Configuration Wizard'. On the left is a decorative sidebar with colorful wavy lines. The main content area has the following fields:

- Board name:** A text input field containing 'arduino'.
- Timezone:** A dropdown menu showing 'Asia/Thailand'.
- Password:** Two stacked text input fields. The top one is labeled 'System Password' and the bottom one is labeled 'System password (Repeat)'.

At the bottom right, there are 'Back' and 'Next' buttons.



The screenshot shows the 'Wireless Settings' screen of the Arduino Configuration Wizard. The title bar reads 'Arduino Configuration Wizard'. On the left is a decorative sidebar with colorful wavy lines. The main content area has the following fields:

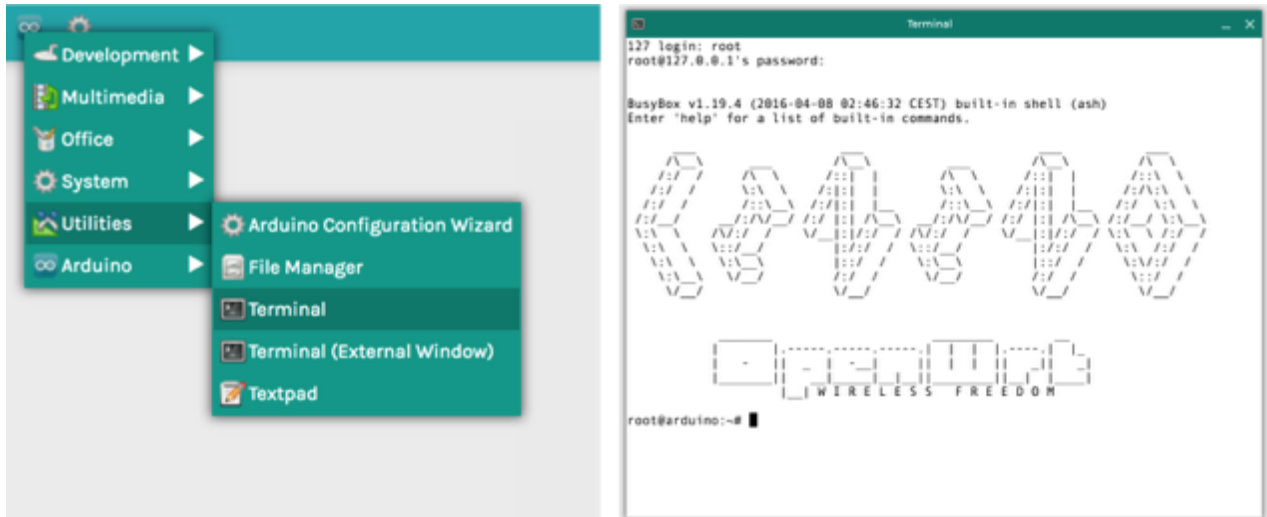
- Wireless Network Name (SSID):** A dropdown menu showing 'Jennis Chung (WiFi, 100% signal)' with a 'Scan' button to its right.
- Wireless Network Name:** A text input field containing 'Jennis Chung'.
- Security:** A dropdown menu showing 'WPA2'.
- Password:** A text input field with a masked password '*****'.

At the bottom right, there are 'Back' and 'Next' buttons.

- On your PC, change your Wifi network to your router and now your NAS and Arduino is under the same network.

1.4 Install development tools for your Arduino Yun

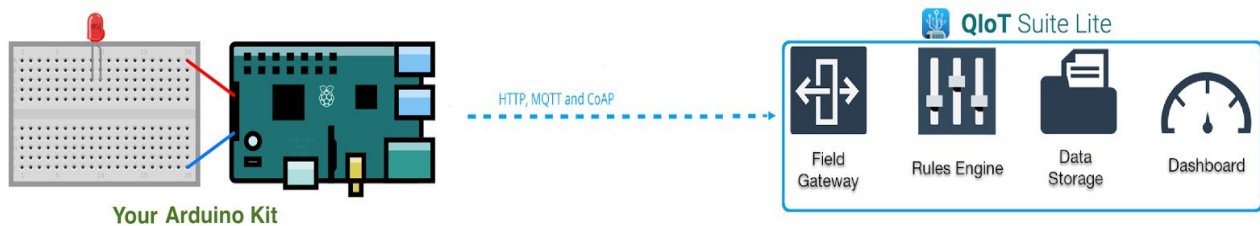
- Open Terminal from <http://arduino.local>
- Type your login account “root” and your password. The default password is “arduino”.



- Install PIP and relatives library by tying in the following commands.
root@arduino:~# opkg update
root@arduino:~# opkg install distribute
root@arduino:~# opkg install python-openssl
root@arduino:~# easy_install pip

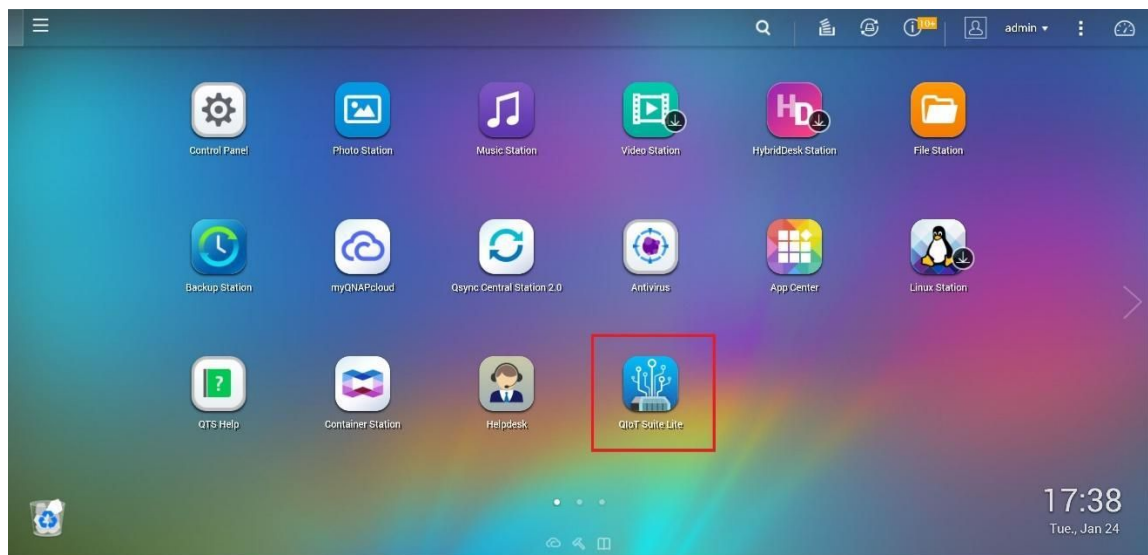
Lesson 2: Create your device in QIoT Suite Lite

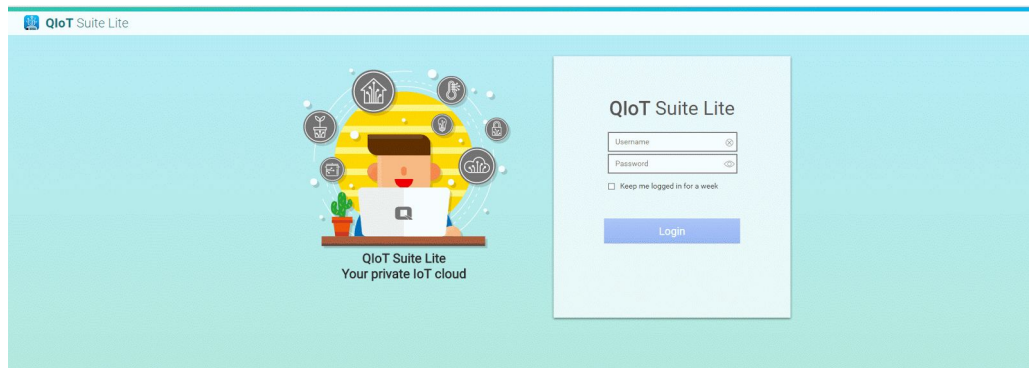
In this lesson, you provision your QNAP QIoT Suite Lite software, and create your first device in QIoT Suite Lite.



2.1 Install QIoT Suite Lite

- Go to QNAP App Center and download QIoT Suite Lite application.
- Launch and log in QIoT Suite Lite software. Use Nas admin and password to login.





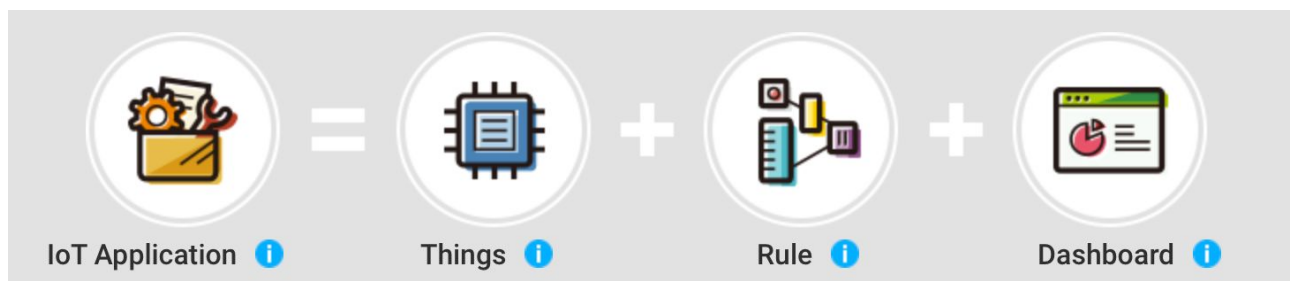
2.2 Create a new IoT application

IoT Application is a combination of multiple Things, Rule, and Dashboard. We recommend that you first create a “Things” in IoT Application. This IoT Application allows you to keep a record of all of the devices that are connected to your NAS.

The Rule makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

With Dashboard, you can turn your data processing efforts into analytics and reports that provide real-time insights into your business.

All these elements provide user a complete IoT Application environment.



- If this is your first time to use QIoT Suite Lite, QIoT Suite Lite provides a wizard to help you quick setup a IoT application.
- Click “Quick Setup Wizard” to start quick setup wizard . After you read the QIoT Suite Lite introduction in pop window, click Next button.

Quick Setup Wizard

admin

Home

IoT Applications

Things

Thing Types

Users

Settings

Home

IoT Applications

4

Things

9

Thing Types

3

NAS(Server) Performance Monitor

Firmware: 4.3.3.0154 Build 20170413

Model: TS-251A

Processor: Intel(R) Celeron(R) CPU N3060 @ 1.60GHz

Memory: 3854.7 MB

0% CPU

0% Memory

0 KB/s

0 KB/s

QIoT Suite Lite Setup Wizard

Introduction

IoT Application

Thing

Resource

API Keys

Introduction

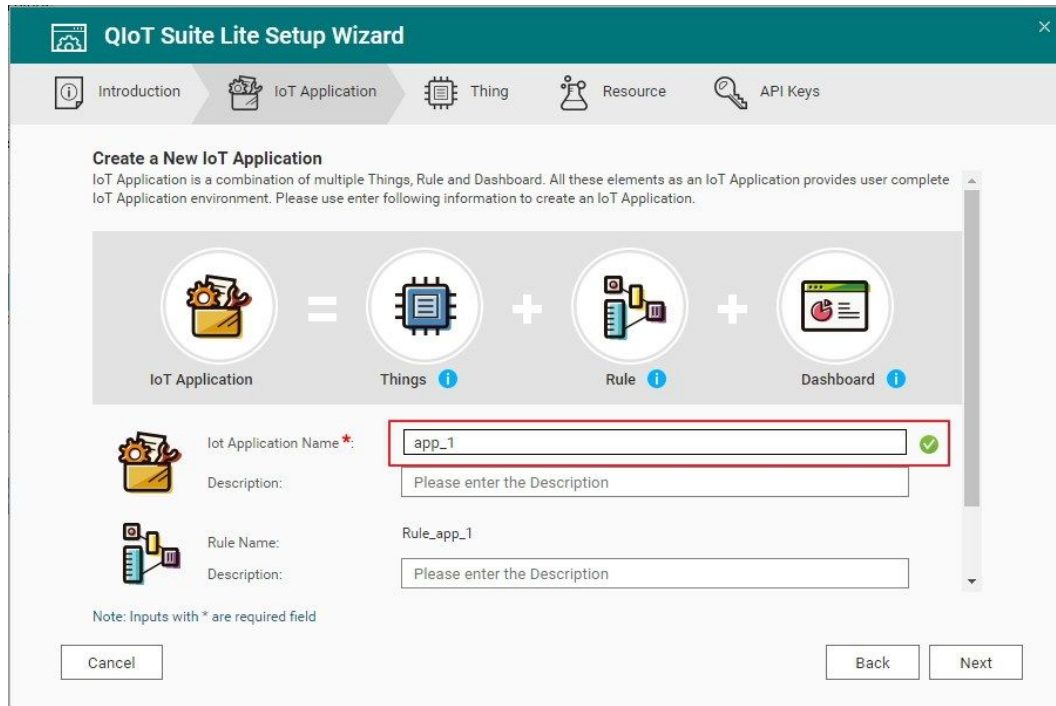
Welcome to QIoT Suite Lite. QIoT Suite Lite is QNAP's Private IoT Cloud Platform, which provides a complete solution to create your IoT Applications on QNAP NAS.

QIoT Suite consists of major components such as Device Gateway, Rule Engine, Dashboard etc. You may use various starter kits like Arduino, Raspberry Pi, Intel Edison or others along with multiple sensors and start pushing telemetry data on QIoT Suite through Device Gateway. Device Gateway supports multiple protocols such as MQTT, HTTP or COAP. This data could be processed and using our robust Rule Engine and appropriate actions could be taken. Dashboards help you to monitor and control your IoT sus-system from single state of the art interface.

Cancel

Next

- Create a New IoT application
 - Enter IoT Application name, e.g., app_1.
 - Rule name and Dashboard name will be generated automatically based on the name of IoT Application you fill in.
 - Click “Next” to complete create a new IoT application



QIoT Suite Lite Setup Wizard

Introduction IoT Application Thing Resource API Keys

Create a New IoT Application
IoT Application is a combination of multiple Things, Rule and Dashboard. All these elements as an IoT Application provides user complete IoT Application environment. Please use enter following information to create an IoT Application.

IoT Application = Things + Rule + Dashboard

IoT Application Name *: ✓

Description:

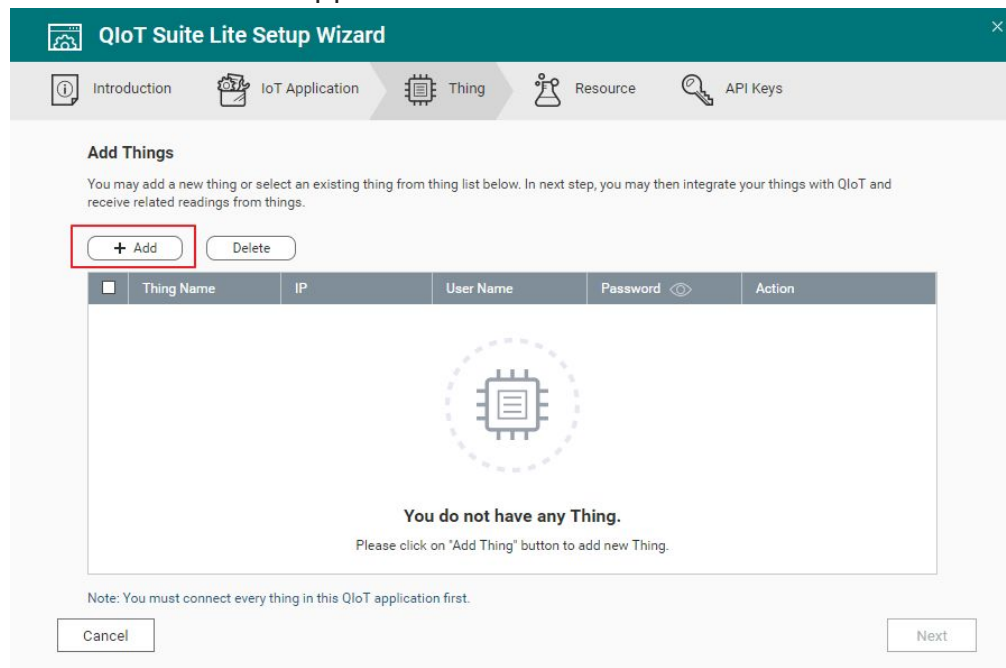
Rule Name: Rule_app_1

Description:

Note: Inputs with * are required field

Cancel Back Next

- Click “+Add” to add this application’s device.




QIoT Suite Lite Setup Wizard

Introduction IoT Application Thing Resource API Keys

Add Things
You may add a new thing or select an existing thing from thing list below. In next step, you may then integrate your things with QIoT and receive related readings from things.

+ Add Delete

<input type="checkbox"/>	Thing Name	IP	User Name	Password	Action
 <p>You do not have any Thing. Please click on "Add Thing" button to add new Thing.</p>					

Note: You must connect every thing in this QIoT application first.

Cancel Next

- After click “+Add ”,In “Add Thing” pop window:
 - Set device’s name (e.g., arduino).
 - Qlot Suite Lite now support Arduino Yun,Raspberry Pi,and Edison ,so you can select “QIoT Supported” thing category and select “Thing Type” is “Raspberry Pi”
 - (optional) add attribute to device’s detail information (e.g., its serial number, manufacturer, and more).
 Click “Add” to add the device to complete create a device.

Add Thing ⓘ

☒ **Create a New Thing**
 Create a thing to represent your device in the QIoT Suite. This step creates an entry in the thing registry or device registry.

☐ **Select from existing Things**
 Select a thing from existing things and associate it with this IoT Application.

Thing Information

Thing Name*: ✓

Description:

Thing Category: ⓘ

☒ QIoT Supported
☐ Custom

Thing Type ⓘ

Thing Type:

Attributes for arduino (Optional): ⓘ

|

Note: Inputs with * are required field

- Please provide IP address, user name, and password of your device, then click “Connect” .After waiting test connection between your device and Qlot Suite Lite success, you could click “Next” to next step.

QIoT Suite Lite Setup Wizard
×

Introduction
IoT Application
Thing
Resource
API Keys

Add Things

You may add a new thing or select an existing thing from thing list below. In next step, you may then integrate your things with QIoT and receive related readings from things.

+ Add
Delete

<input type="checkbox"/>	Thing Name	IP	User Name	Password	Action
<input type="checkbox"/>	arduino	172.17.30.115	root	*****	Connect ✓ ✕

Note: You must connect every thing in this QIoT application first.

Cancel
Next

- In “Resources” step,Click “Add Resource”.

QIoT Suite Lite Setup Wizard
×

Introduction
IoT Application
Thing
Resource
API Keys

Add Resource

Thing Resources represent Sensors, Thing Peripherals, Switches or any dynamic properties of the Thing like CPU usage etc. Please add these resources to the device with desired configurations.

+ Add Resource
Delete

Thing Name:

arduino(0) ✕

<input type="checkbox"/>	Resource Name	Resource ID	Resource Type	Data Type	Unit	Description	Actions
<p>You do not have any Thing Resource. Please click on 'Add Resource' button to add new resource.</p>							

⏪
⏩
Page
1 / 1
⏪
⏩

0 - 0 of 0

Cancel

Back
Next

- After click “Add Resource”,In “Add Thing Resource” pop window:
 - Set resource’s name (e.g., temp).
 - Set resource’s id.Resource id will be used to create a topic in the QIoT broker.This ID should be unique for the device and no duplicates should be allowed for the same device
 - And set another optional attribute.

Add Thing Resource

Resource Name*: temp ✓

Resource Description: Please enter the description ✓

Resource ID*: temp ✓

Resource Type: Temperature ▼

Data Type: Float ▼

Unit: °C ▼

☐ Set Range

Minimum Value: Please enter minimum value

Maximum Value: Please enter maximum value

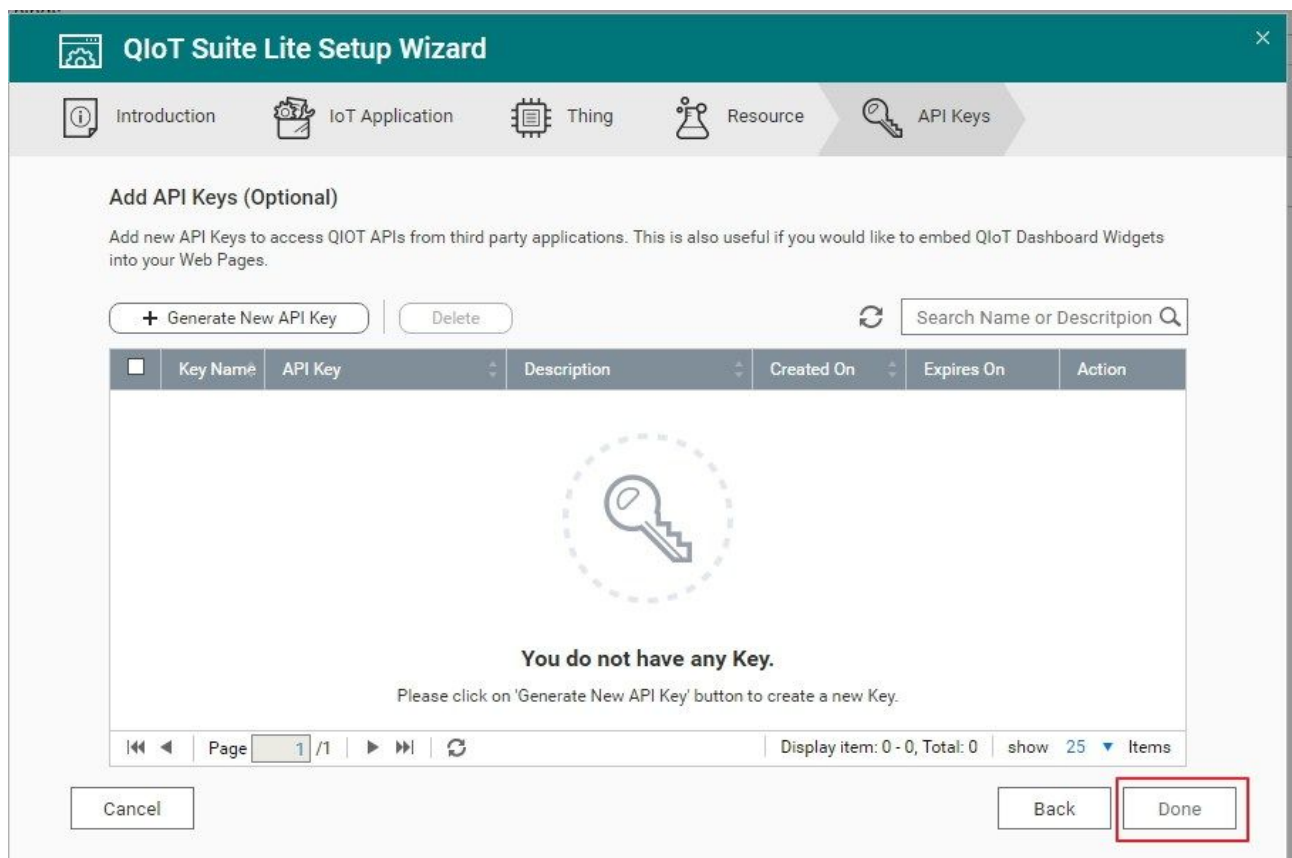
☐ Set Default Value

Default Value: Please enter default value

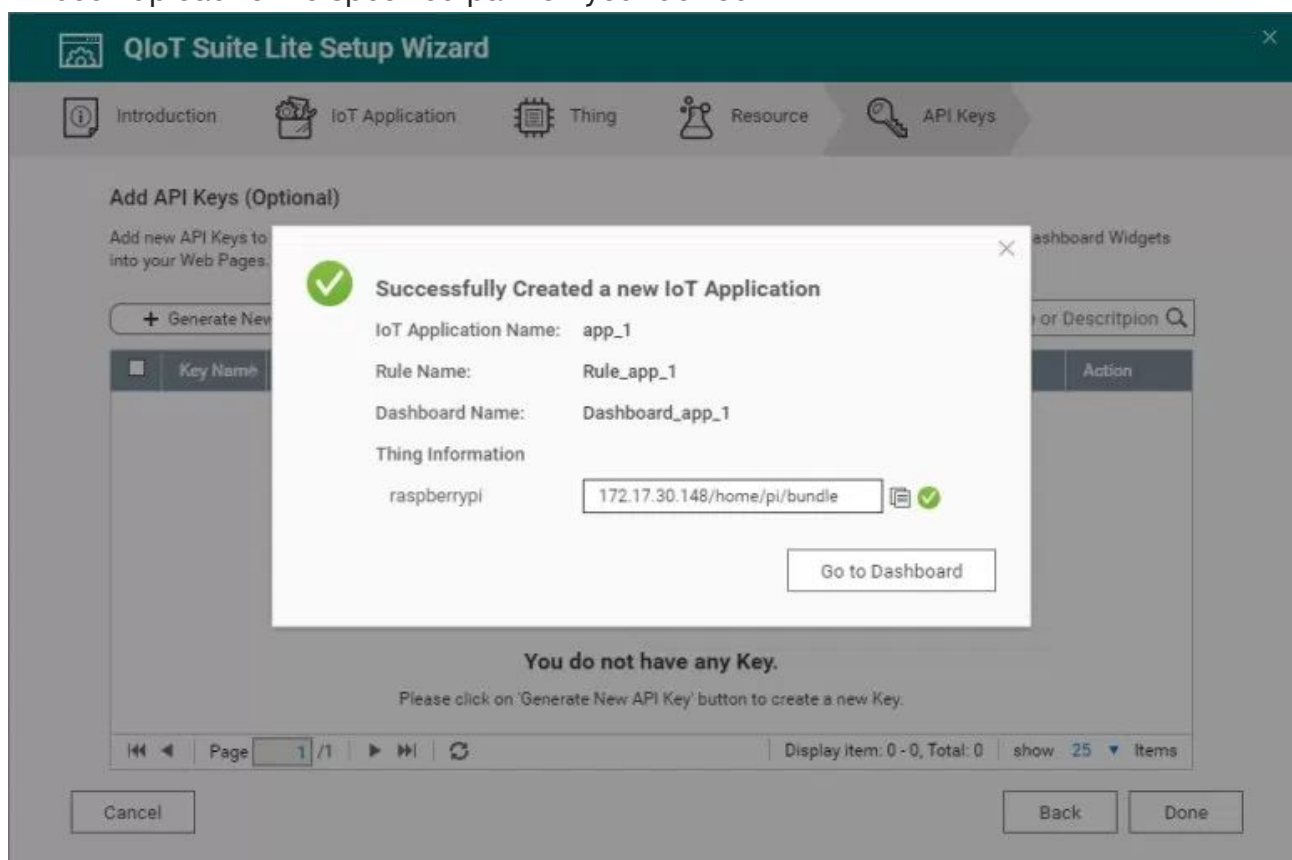
Note: Inputs with * are required field

Add Cancel

- Click Next after you add all the resources (sensors) on your device.
- In “API Keys” step :
 - If you would like to embed QIoT dashboard widgets into your web pages or access QIoT APIs from third party applications.You could click “generate new API Key” to create API key or click “done” start deploy sample code to your device.



- For a while, sample codes and related files (certificate, resource information) have been upload to the specified path on your device.



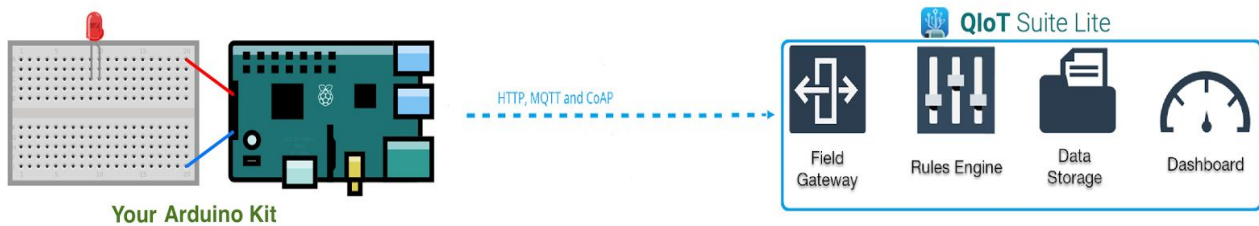
- The screenshot displays the QIoT Suite Lite web interface. On the left is a teal sidebar with navigation links: Home, IoT Applications, Things, Thing Types, Users, and Settings. The main content area has a top bar with a 'Quick Setup Wizard' button, a user profile icon labeled 'admin', and a menu icon. Below this is a breadcrumb trail: 'app_1' with sub-links for 'Dashboard', 'Rule', and 'Things'. The 'Dashboard' tab is active, showing a card titled 'Dashboard_app_1' with a subtitle 'Created on 2017/5/19 05:15:43'. The card contains a 'Temperature' gauge. The gauge is a semi-circle with a yellow-to-white gradient, showing a value of 16. The scale ranges from -40 to 125, with 'centigrade' written below the scale.

Temperature (centigrade)
16

-
- The screenshot displays the QIoT Suite Lite interface. On the left is a teal sidebar with navigation icons and labels: Home, IoT Applications, Things, Thing Types, Users, and Settings. The top header bar shows the application name 'QIoT Suite Lite' and a 'Quick Setup Wizard' button. Below the header, the main workspace is titled 'app_1' and contains tabs for 'Dashboard', 'Rule', and 'Things'. The 'Rule' tab is active, showing a configuration for 'Rule_app_1' created on 2017/5/19 05:15:43. The rule editor displays a flowchart for 'Flow 1' with the following steps:
- Configure QBroker node and connect to the dashboard**
 - 1. Subscribe things**: This step involves an 'MQTT Message In' node connected to a 'Dashboard' node. A green status bar below the MQTT node indicates it is 'connected'.
 - 2. Select resource type**: This step involves a 'Click debug tab to watch data stream' button.
 - 3. Publish things**: This step involves a 'Virtual Event' node connected to a 'dummy' data node, which is then connected to an 'MQTT Message Out' node. A green status bar below the MQTT node indicates it is 'connected'.
- The interface also includes a 'filter nodes' search bar and a list of available nodes on the left, categorized under 'QNAP' (QDashboard, QBroker, QDatabase, QHistoricData) and 'input' (inject, catch, status, link, mqtt, http, websocket, tcp).

Lesson 3: Connect your device to QIoT Suite Lite using MQTTS

In this lesson, you generate certificate from QIoT Suite Lite, download SDK, and connect Arduino Yun to QIoT Suite Lite.



3.1 Run Sample Code

- Open Terminal application (e.g., [PuTTY](#)) on your PC. Connect to your device by SSH and enter the folder where put sample code (e.g., /home/root/bundle).

[illegible]

- Install sample code dependency,enter command as following
root@arduino:~# cd /home/{{user}}/bundle
root@arduino:~/home/root/bundle# pip install paho-mqtt

- Run sample code in device will publish message to topic “temp” by MQTTS as following picture. Topic is define from resource id that you setted.

```
while 1:
    """
    about ./res/resourceinfo.json
    {
        ~
        "resources": [
            {
                ...
                "resourceid": "temp",
                "topic": "qiot/things/admin/abccccc/temp",
                ...
            }
        ]
    }
    TODO: you could replace "temp" by any resource id set form QIoT Suite Lite
    """
    connection.publish_by_id("temp", str(random.randint(0, 41)))
    """
    or publish by resource topic
    TODO: you could replace "qiot/things/admin/edison/temp" by any Topic form QIoT
```

- Run the sample application.

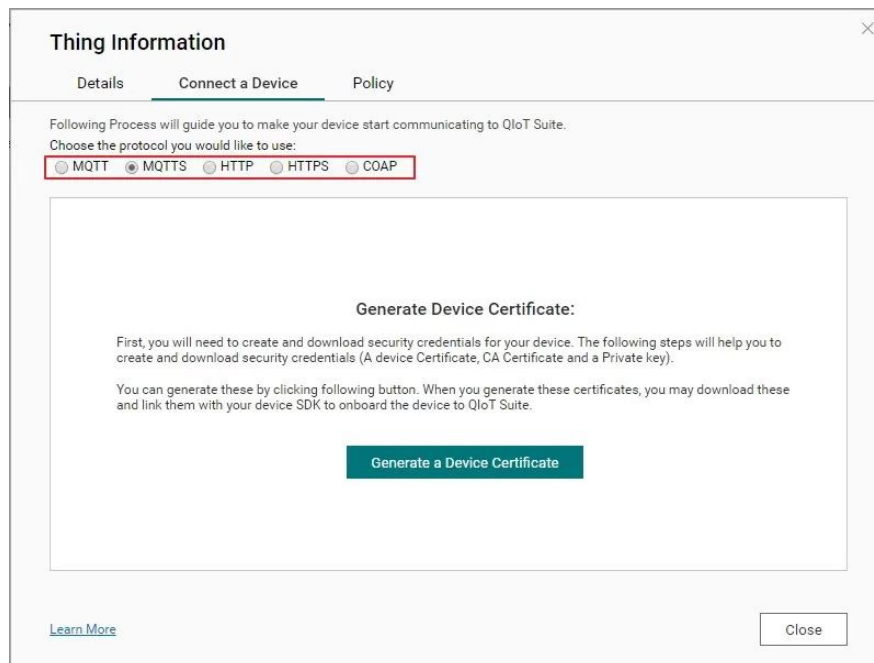
```
root@arduino:~/home/root/bundle# python mqtt.py
```

- device will send message to topic “temp” or that you defined ,as below image.

```
root@arduino:/home/root/bundle# python mqtt.py
new mqtt protocol
CLIENT_CERT path :/home/root/bundle/ssl/3048642017-05-16_10-31-51_certificate.pem
PRIVATE_CERT exists or not :True
Use MQTTS
USER_NAME : 0cf76869-43d5-49ec-95a6-197e74798ed8 USER_PASS : r:6723da30e376545f341ceb10a0d13b05
finish setup
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":39}
connect ready
connection ready
client ready
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":31}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":40}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":39}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":9}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":3}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":31}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":35}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":34}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":8}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":22}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":20}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":5}
```

3.2 Another protocol

- Click “Connection a device” button
- You can choose another protocol you would like to use



- SSH to your device , and input command as following.

```
root@arduino:~/home/root/bundle# cd /home/{{user}}/bundle
```

```
// mqtt(dont' need certificate,just put JSON file to "res" folder):
```

```
root@arduino:~/home/root/bundle# pip install paho-mqtt
```

```
root@arduino:~/home/root/bundle# python mqtt.py
```

```
// http
```

```
root@arduino:~/home/root/bundle# pip install requests
```

```
root@arduino:~/home/root/bundle# python http.py
```

```
// https
```

```
root@arduino:~/home/root/bundle# pip install requests
```

```
root@arduino:~/home/root/bundle# python https.py
```

```
// coap
```

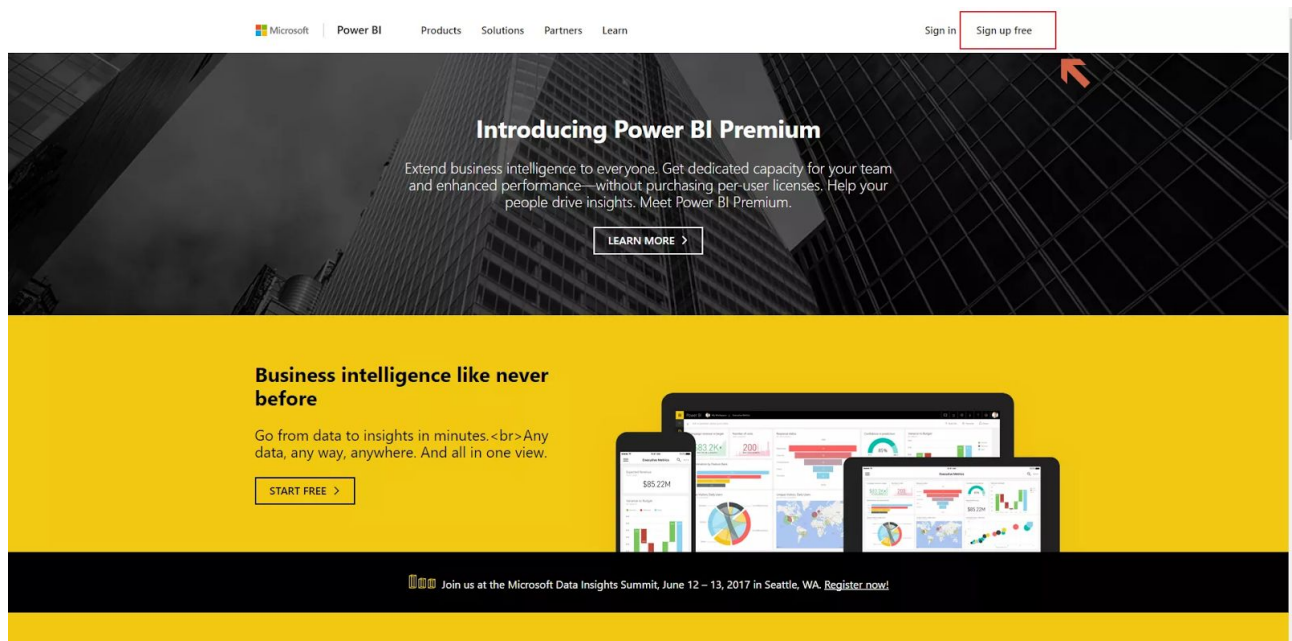
```
root@arduino:~/home/root/bundle# pip install coapthon
```

```
root@arduino:~/home/root/bundle# python coap.py
```

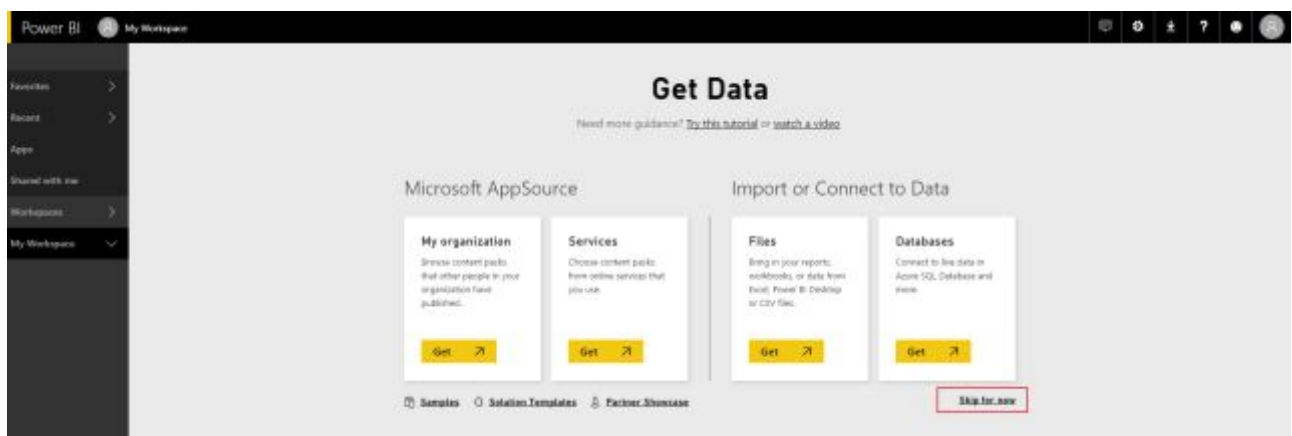

Lesson 4: Integrate Power BI

4.1 Get your first Power BI account

- Go to the official website “<https://powerbi.microsoft.com/en-us/>” to sign up your free account.

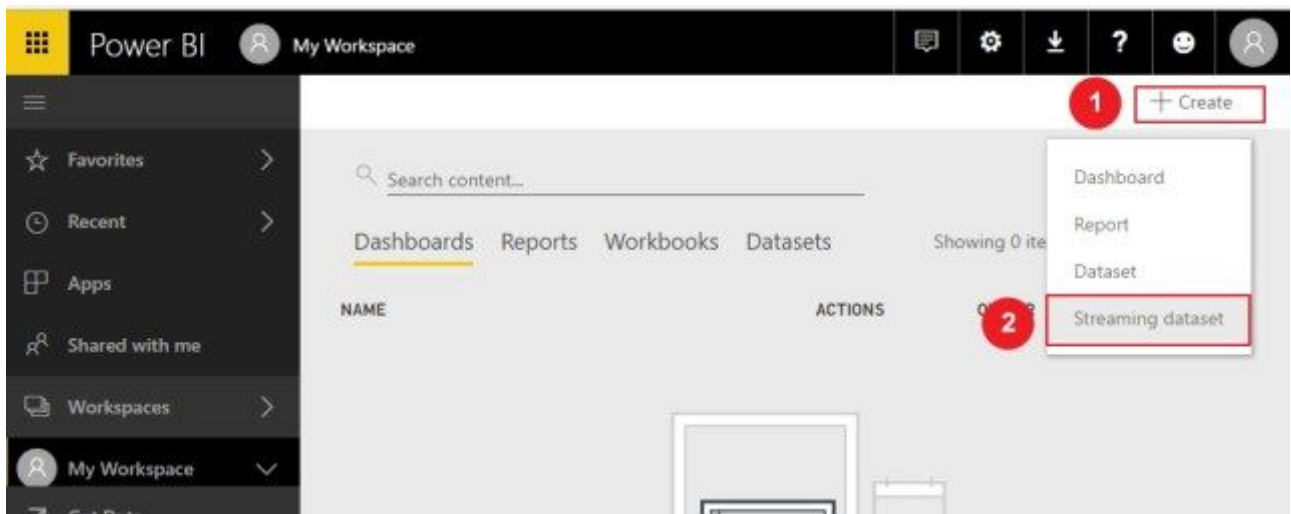


- After a sequence of registration, the page will lead you to below page, you can press “My workspace”, and “skip for now” button appear. You could click “skip for now” to start create dataset.

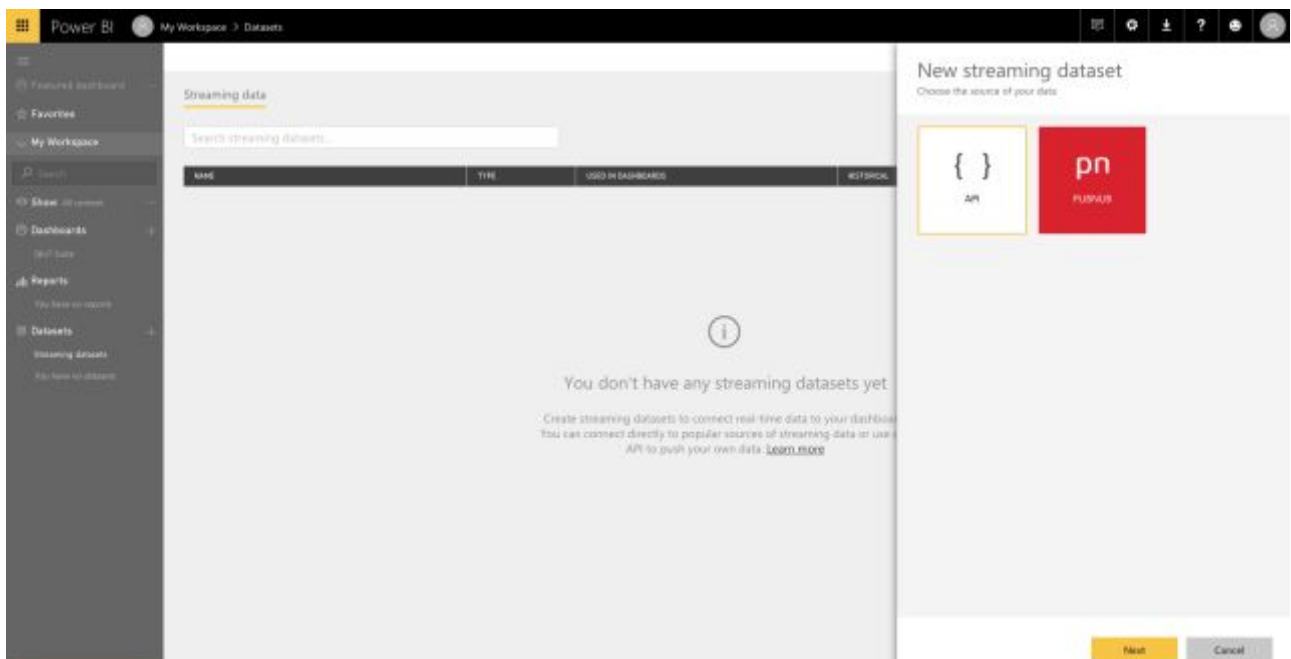


4.2 Setup your streaming dataset API

- Create “Datasets”
 - Click “Create” in screen upper right corner
 - And then click “Streaming dataset”



- Select “API”, and click “Next”.



- Define your values from stream(e.g., temp,max,min), and you will get a result of JSON in textbox. Qlot Suite’s application will post this data format to Power BI. Click the “Create” button to finish create streaming dataset.

Edit streaming dataset

Create a streaming dataset and integrate our API into your device or application to send data. [Learn more about the API.](#)

Dataset name *

Values from stream *

Number ▼



Number ▼



Number ▼



Text ▼

```
[
  {
    "temp" : 98.6,
    "max" : 98.6,
    "mix" : 98.6
  }
]
```

Historic data analysis



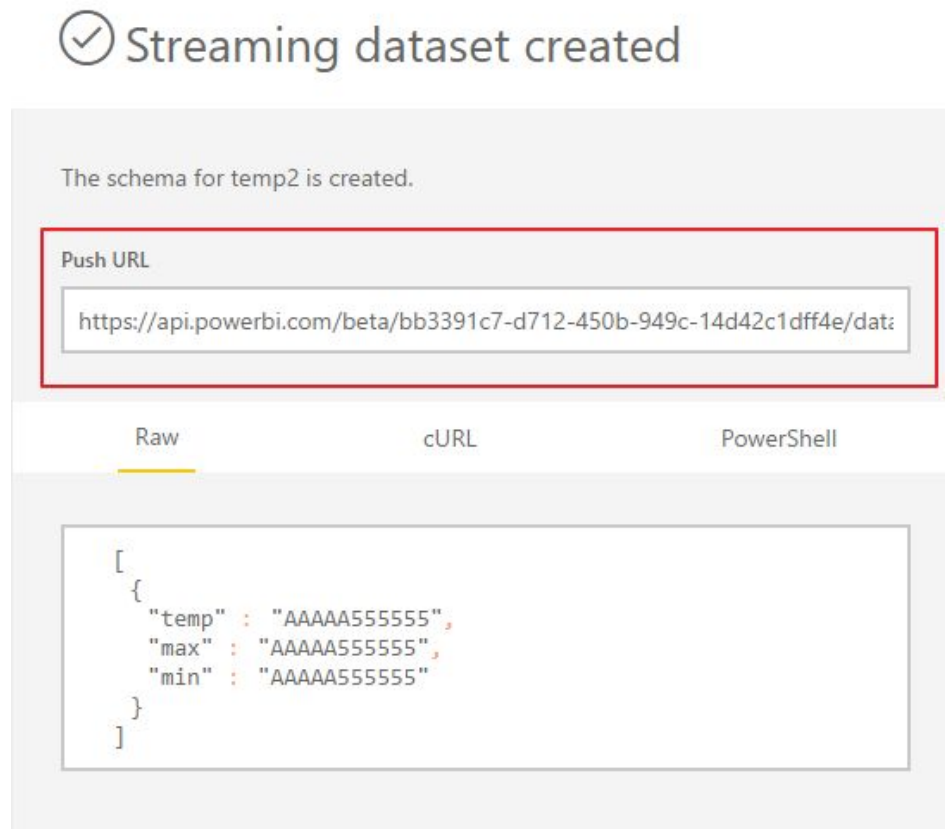
On

Back

Done

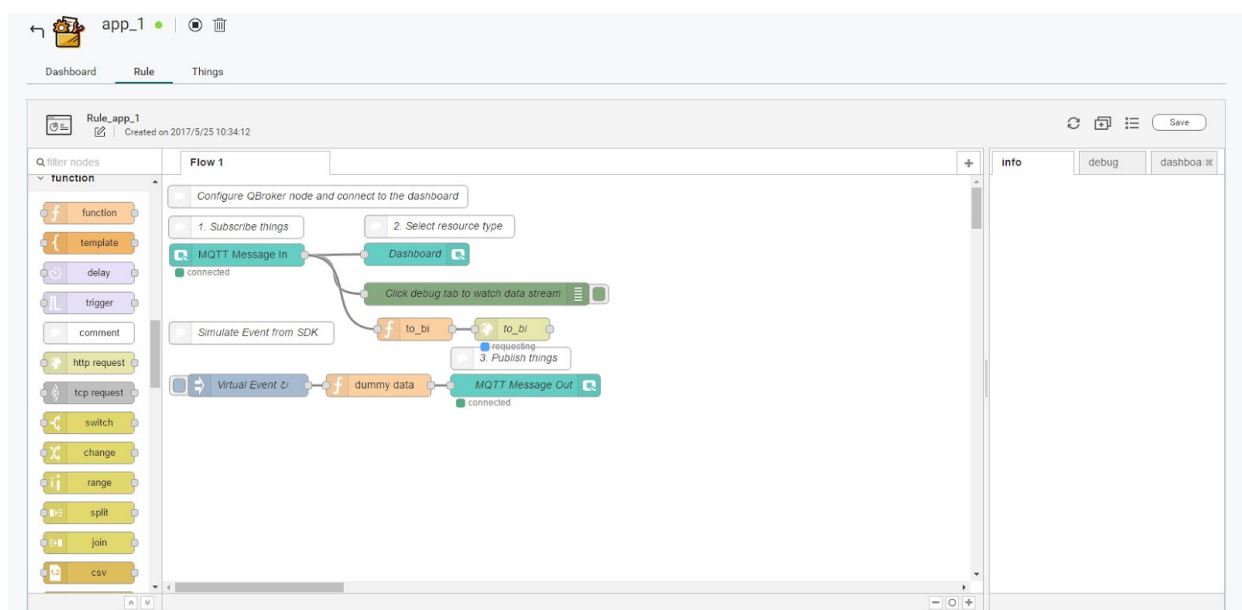
Cancel

- Once you successfully create your data stream, you get REST API URL which QIoT suite application can call using POST request to push your live data to streaming data dataset you created.



4.3 Configure Node-RED's nodes in IoT application

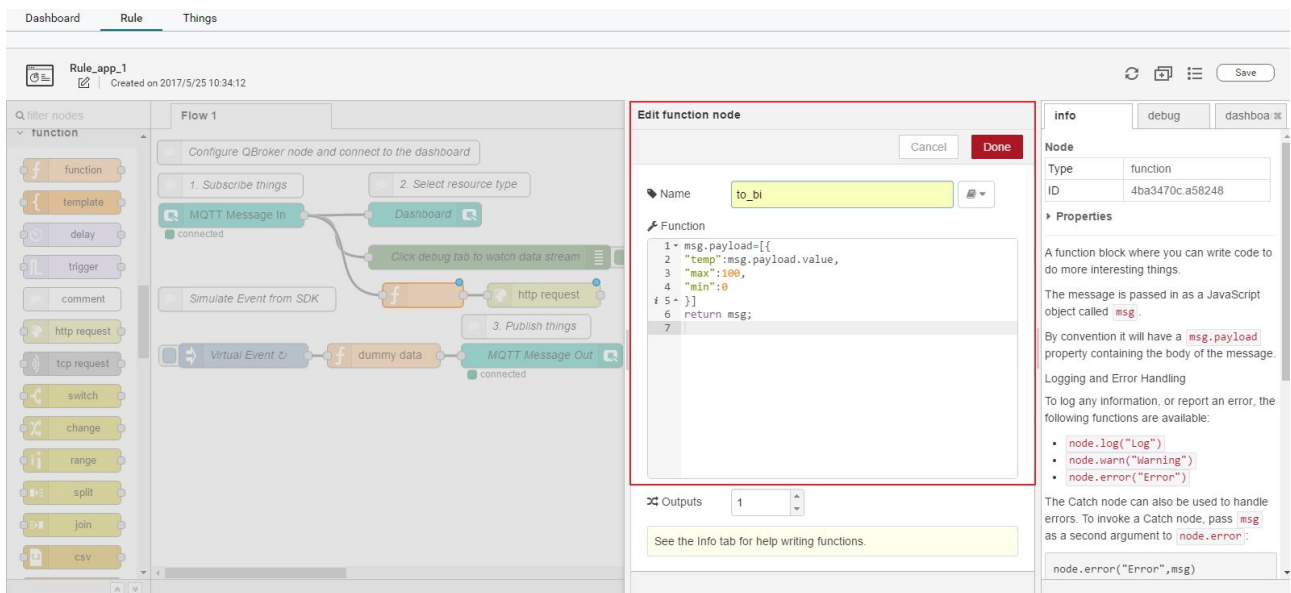
- Create IoT application in QIoT suite.
- The following one is your first node-red flow, and then you can start to create your own IoT flow. more node-red information can be found in “[Node-Red](#)”.



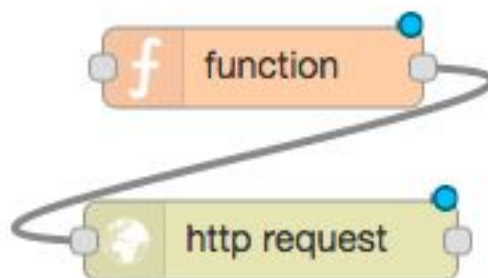
- Before you start to push live data to Power BI.
- We need a “function” node to convert IoT data to streaming data dataset. Here you can replace `msg.payload` to your JSON dataset.

- Double click function node, and type Function code as following:

```
msg.payload=[{
  "temp":msg.payload.value,
  "max":100,
  "min":0
}]
return msg;
```



- We need a “http request” node to help us to push live data to Power BI. Just drop and drag “http request” node and connect to tail of “function” node.



- Double click http request node, copy and paste REST API URL that you got from Power BI console, and set http method to POST.

app_1 | Dashboard Rule Things

Rule_app_1 Created on 2017/5/25 10:34:12

Flow 1

Configure QBroker node and connect to the dashboard

1. Subscribe things 2. Select resource type

MQTT Message In connected

Click debug tab to watch data stream

Simulate Event from SDK

to_bi http request

3. Publish things

Virtual Event dummy data MQTT Message Out connected

Method POST

URL https://api.powerbi.com/beta/bb3391c7-d712-45f1-9b32-3d9b7f9d9d9d

Enable secure (SSL/TLS) connection

Use basic authentication

Return a UTF-8 string

Name to_bi

Info debug dashboard

Node

Type http request

ID 6ab1371d.a49a58

Properties

Provides a node for making http requests.

The URL and HTTP method can be configured in the node, if they are left blank they should be set in an incoming message on `msg.url` and `msg.method`:

- `url`, if set, is used as the url of the request. Must start with http: or https:
- `method`, if set, is used as the HTTP method of the request. Must be one of GET, PUT, POST, PATCH or DELETE (default: GET)
- `headers`, if set, should be an object containing field/value pairs to be added as request headers
- `payload` is sent as the body of the request

When configured within the node, the URL property can contain `mustache-style` tags. These allow the url to be constructed using

- Finally, don't forget to press "Save" button to save changes.

app_1 | Dashboard Rule Things

Rule_app_1 Created on 2017/5/25 10:34:12

Flow 1

Configure QBroker node and connect to the dashboard

1. Subscribe things 2. Select resource type

MQTT Message In connected

Click debug tab to watch data stream

Simulate Event from SDK

to_bi http request

3. Publish things

Virtual Event dummy data MQTT Message Out connected

Successfully deployed

Save

Info debug dashboard

Node

Name to_bi

Type http request

ID 6ab1371d.a49a58

Properties

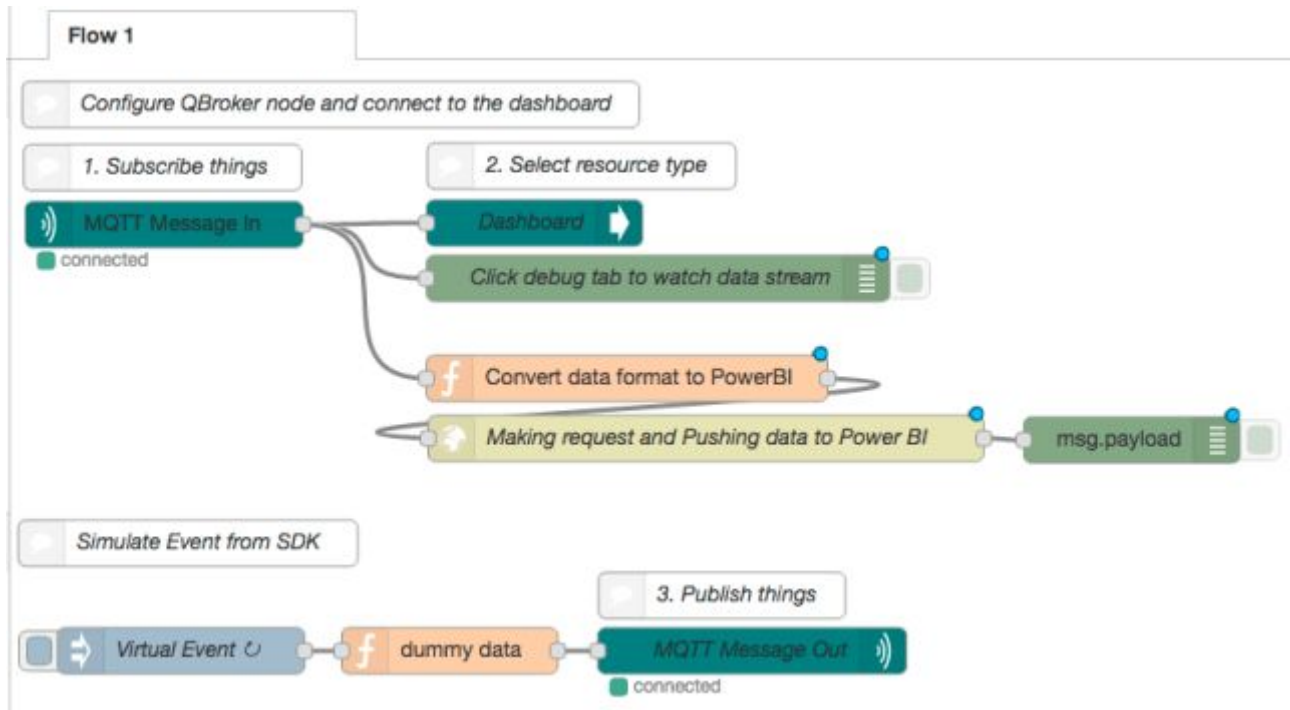
Provides a node for making http requests.

The URL and HTTP method can be configured in the node, if they are left blank they should be set in an incoming message on `msg.url` and `msg.method`:

- `url`, if set, is used as the url of the request. Must start with http: or https:
- `method`, if set, is used as the HTTP method of the request. Must be one of GET, PUT, POST, PATCH or DELETE (default: GET)
- `headers`, if set, should be an object containing field/value pairs to be added as request headers
- `payload` is sent as the body of the request

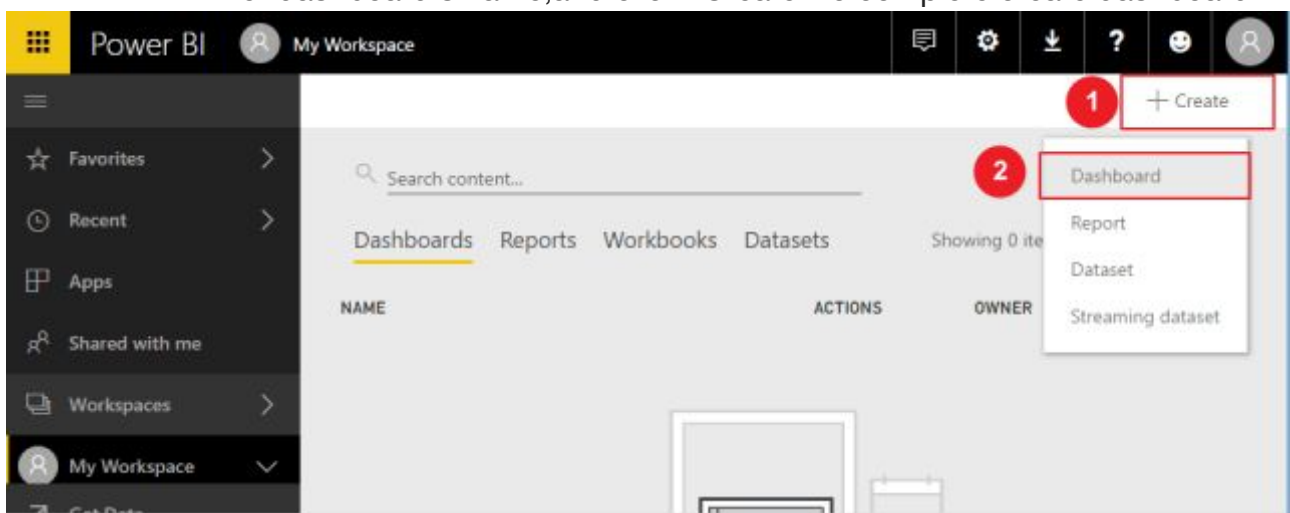
When configured within the node, the URL property can contain `mustache-style` tags. These allow the url to be constructed using

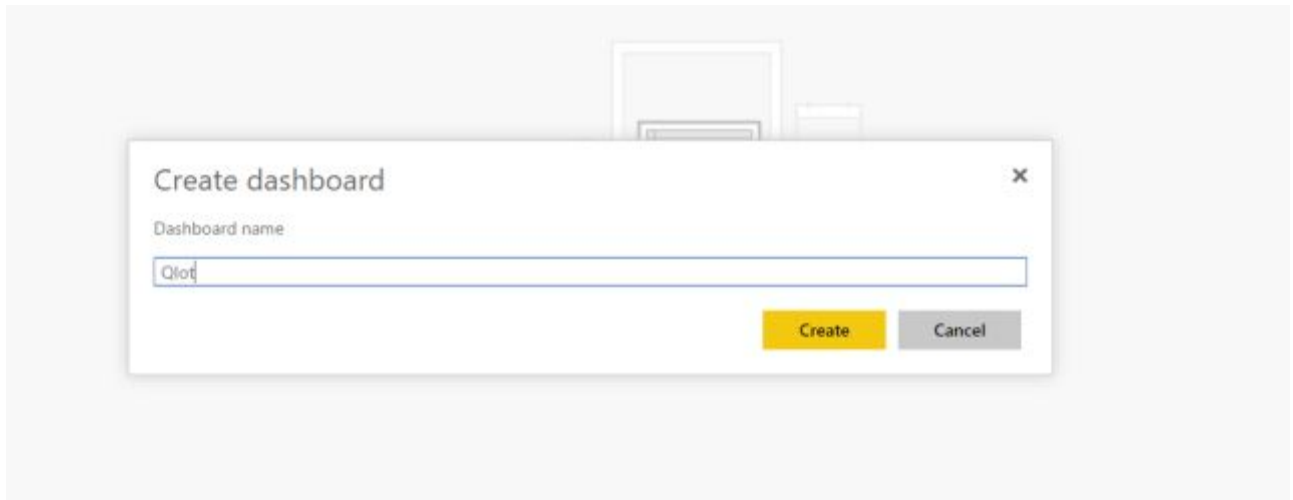
- Finally, your node-red flow will look like below one.



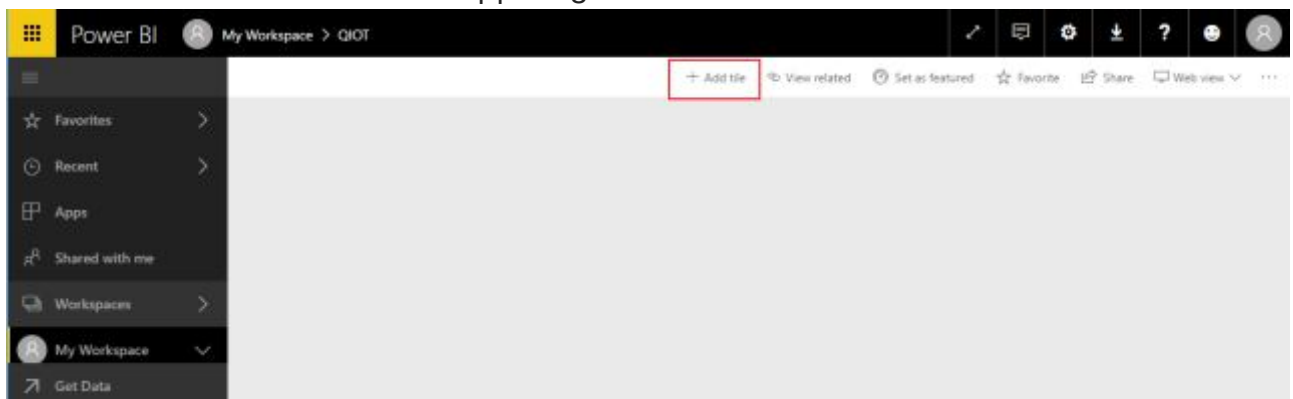
4.4 Add tile to display real-time data

- Create "Dashboard"
 - Click "Create" in screen upper right corner
 - And then click "Dashboard"
 - Enter dashboard's name, and click "Create" to complete create dashboard.

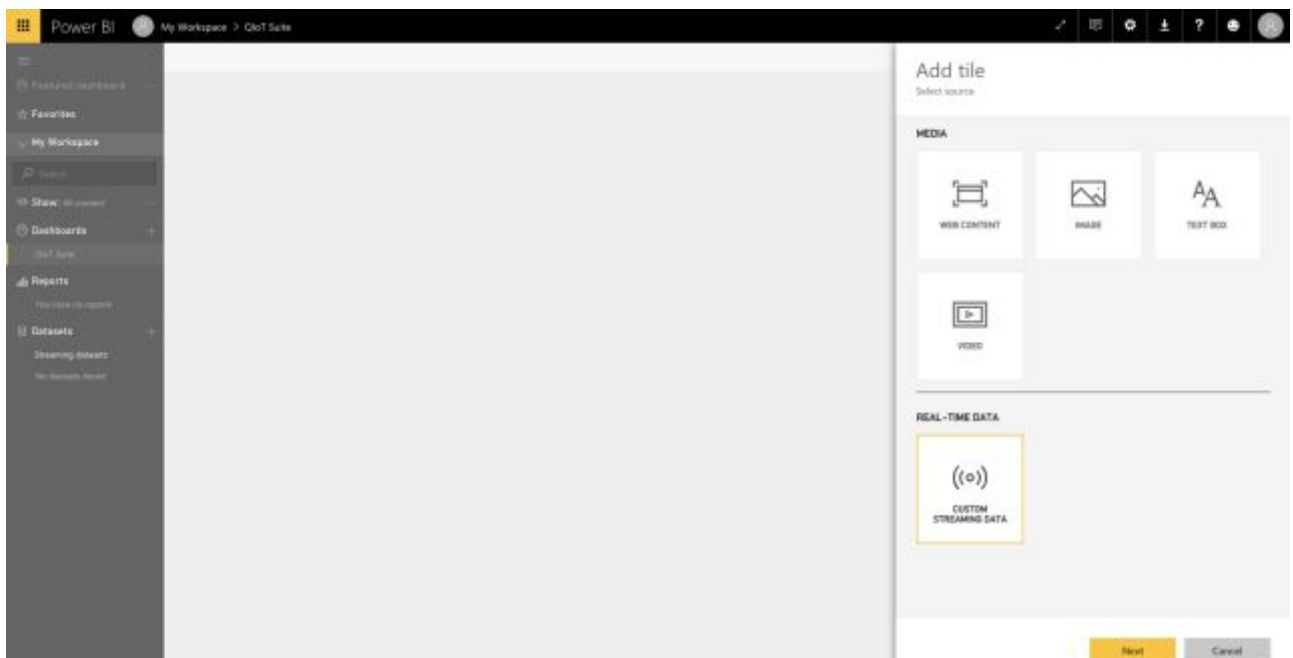




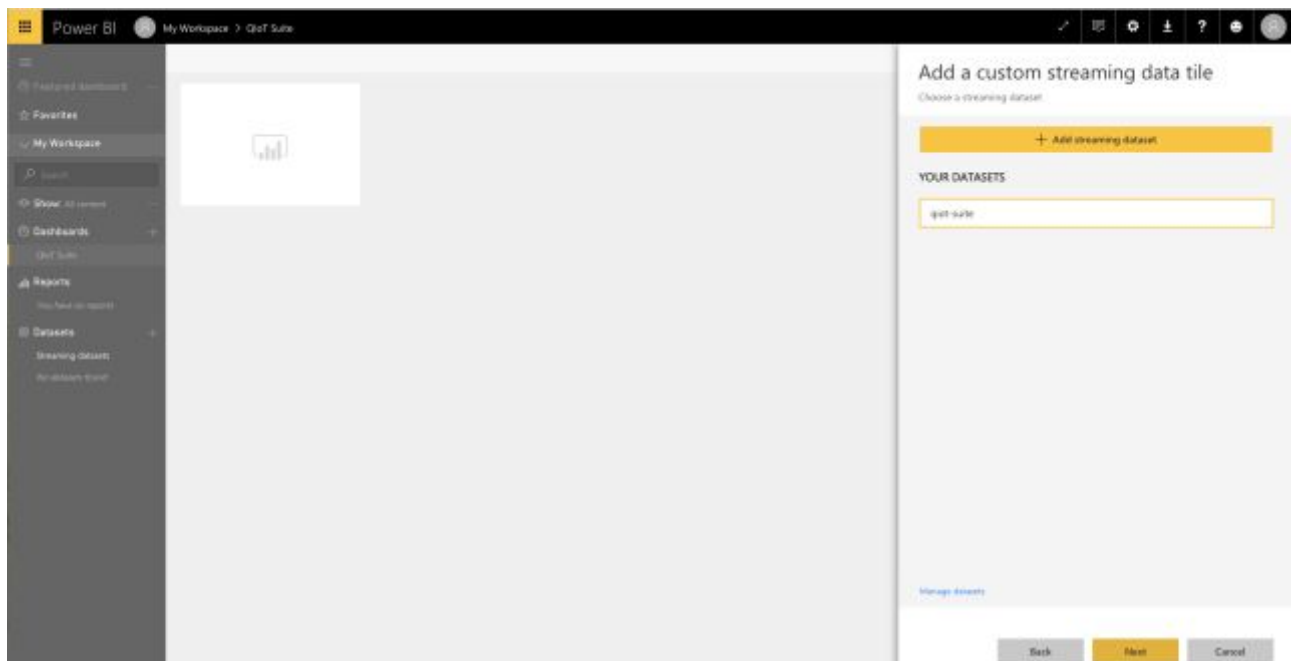
- Click “Add tile” in screen upper right corner



- Select “CUSTOM STREAMING DATA” and then select the “Next” button.



- Select **datasets** and then click the “Next” button.



- Select visualization type (e.g., gauge), and set value, min, and max value.

Add a custom streaming data tile

Choose a streaming dataset > Visualization design

Visualization Type

Gauge

Value

temp

+ Add value

Minimum value

min

+ Add value

Maximum value

max

+ Add value

Target value

+ Add value


[Manage datasets](#)

Back Next Cancel

- You have a streaming dataset to work with, you can get a real time gauge that looks like as following.

Power BI | My Workspace > QoT Suite

Temperature



36.00

0 100

Tile details

Visualization design | Tile details

Visualization Type: Gauge

Value: temperature

Minimum value: min

Maximum value: max

Target value: Add value

Apply Cancel

Appendix

QNAP QIoT Startkit Sample Code Introduction

- GitHub: [Sample Code](#)
- Sample Code Structure

```

qnap-qiotsdks/
  python/                                # python program language
  device/                                # arduino-yun...
    arduino-yun/
      examples/
        lib/                             # QIoT command Lib
        res/                             # QIoT resourceinfo.json folder
        ssl/                             # QIoT certificate files folder.
        mqtt.py                          # sample code - mqtt/mqtts publish
        http.py                           # sample code - http post
        https.py                          # sample code - https post
        coap.py                           # sample code - coap postt
        mqtt_subscribe.py                 # sample code - mqtt/mqtts subscribe
        http_get.py                       # sample code - http get
        https_get.py                      # sample code - https get
        coap_observe.py                   # sample code - coap get
  nodejs/                                # node.js program language
    device/
      intel-edison/
        examples/
          lib/
          res/
          ssl/
          mqtt.js
          http.js
          ...

```

- content of resourceinfo.json

protocol	resourceinfo.json content
mqtts	<pre> { "host": ["172.17.28.28" # nas ip], "myqnapcloudHost": "Not Available", # myqnapcloudHost "port": 8883, # mqtts port "clientId": "adfa_1491561635", # thing Id "username": "00477f86-425b-49de-8590-xx", # username "password": "r:2825dedfb012969e1dfb6adb8", # password }</pre>

	<pre> "resources": [# resource des { "resourcename": "adf", # resource name "resourceid": "dfadf", # resource id "resourcetyponame": "Temperature", # resource type "datatype": "Float", # data type "unit": "°C", # data unit "description": "adfa", # resource des "topic": "qiot/things/admin/adfa/dfadf" # topic name }], "caCert": "/v1/media/ca-crt.pem", # certificate file "clientCert": "/v1/media/xx-04-07_10-40-35/xx_certificate.pem", "privateCert": "/v1/media/xx-04-07_10-40-35/xx_privatekey.pem" } </pre>
https	<pre> { "accesstoken": "r:2825dedfb012969e1dfb6adb8", # password "myqnapcloudHost": "Not Available", "clientId": "adfa_1491562164", "host": ["172.17.28.28"], "requesterid": "00477f86-425b-49de-8590-xx", # username "port": 3443, # https port ... } </pre>
CoAP	<pre> { "myqnapcloudHost": "Not Available", "clientId": "adfa_1491562176", "host": ["172.17.28.28"], "r": "00477f86-425b-49de-8590-1282c65b4348", # username "t": "r:2825dedfb012969e1dfb6adb80a419df", # password "port": 5683, # coap port ... } </pre>