AtlasScientific™
Environmental Robotics

Datasheet 2.1

Atlas·iot™

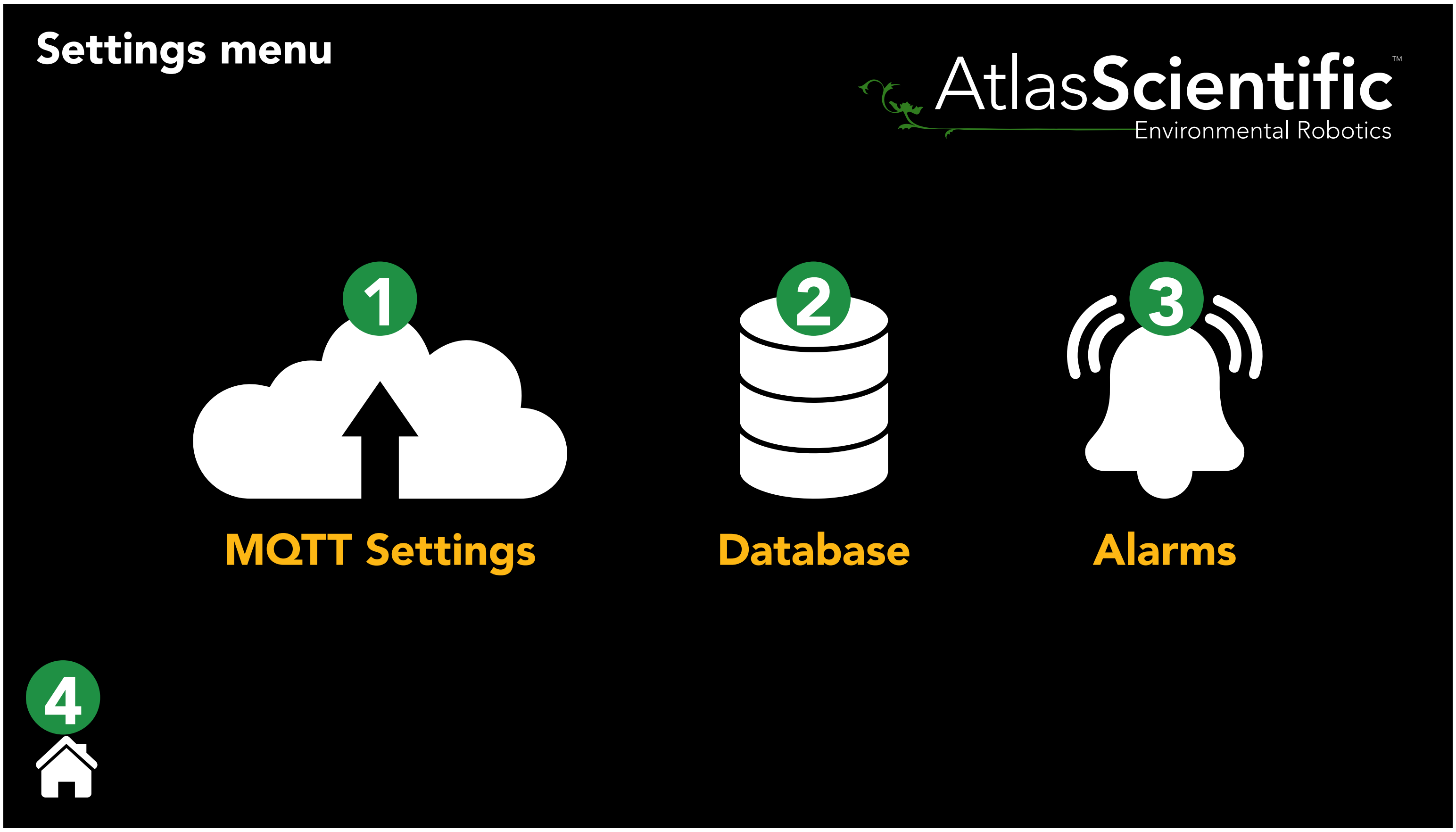
Settings guide

Table of contents

Settings Menu

You can access the settings menu for the The Atlas iot™ software, by pressing the "⚙" icon, located at the top left corner of the main screen.

Inside the Settings Menu, you'll see 4 options.



Settings menu

1. MQTT Settings

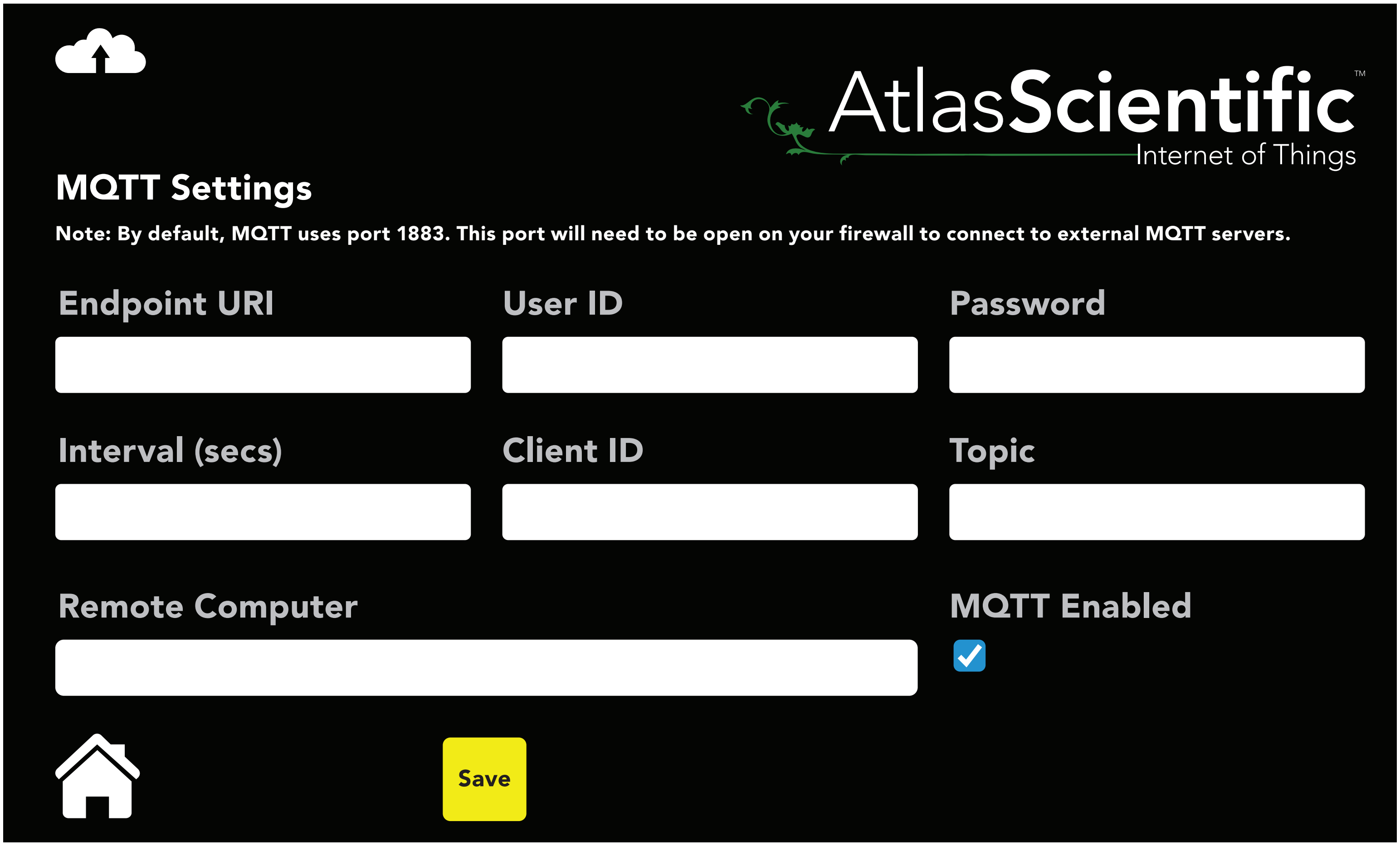
2. Database

3. Alarms

4. Home Button

Currently, the Alarms section is in BETA and the Home Button is self explanatory ; So we'll go over both the MQTT Settings and the Database Settings.

MQTT Settings



MQTT Settings

Note: By default, MQTT uses port 1883. This port will need to be open on your firewall to connect to external MQTT servers.

MQTT Enabled ☑

Endpoint URI - The web address of your MQTT broker.

Example io.adafruit.com

User ID - If your broker requires a login, this is where your user ID goes.

Example myUserId

Password - If your broker requires a login, this is where your password goes.

Example mypassword

Interval (secs) - How often the IoT should send to data to the broker.

Example 60

ClientID - A random ClientID. *(no spaces)*

Example My_Device

Topic - The MQTT topic the data will be sent out under.

See http://www.steves-internet-guide.com/understanding-mqtt-topics/

Note
We append on some subtopics /SensorType/Add_(SensorAddress)/SensorName

Example /RTD/add_102/MyName

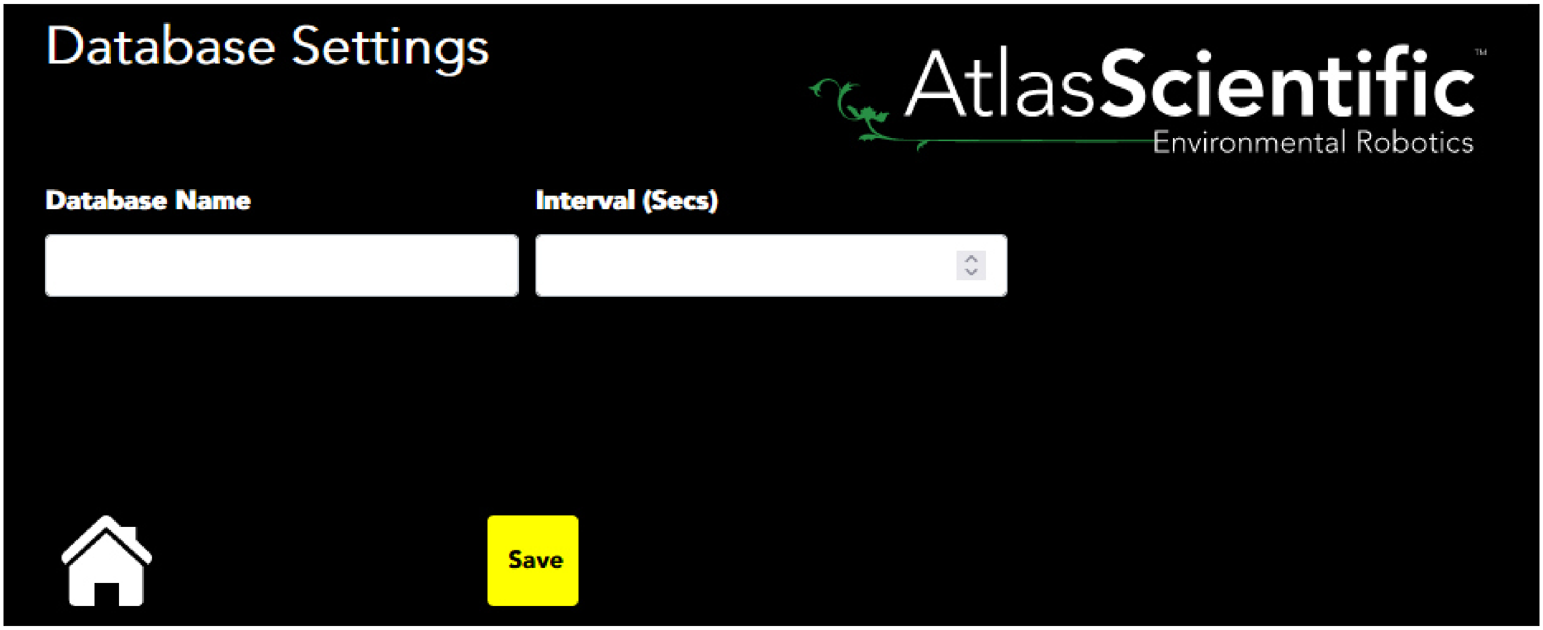
If there is no name set, we leave off that subtopic.

Remote Computer - Remotely connect to another Atlas IoT and display its readings.

Example remotepi.local

Note
After entering the IP address or computer name of the remote computer; the Atlas IoT must be restarted, inorder for the connection to be established.

Database Settings



Database Name - This field shows the filename and location of the database.

./IoTData.db - This is the default filename and location of the database.

Interval (secs) - How often the database is saved.

Example 60

In the database is a table named SensorData, which have the following fields:

id INTEGER PRIMARY KEY	=	**contains the ID number of the record**
RecordDateUTC text	=	**contains time and date of readings: yyyy-mm-dd-00:00:00**
sensorType text	=	**contains the type of sensor(s) you are using**
moduleName text	=	**contains the name you have given to the sensor(s)**
lastReading text	=	**contains the last reading from the sensor(s)**
sensorData text	=	**contains All JSON information**

MQTT with Mosquitto

Note: By default, MQTT uses port 1883. This port will need to be open on your firewall to connect to external MQTT servers.

Mosquitto is an open source MQTT broker that works quite well and is easy to troubleshoot. It can be found at https://mosquitto.org and is well supported.

If you assume your mosquitto broker is on a computer called **MyComputer.**
Your settings for the Atlas IoT would be

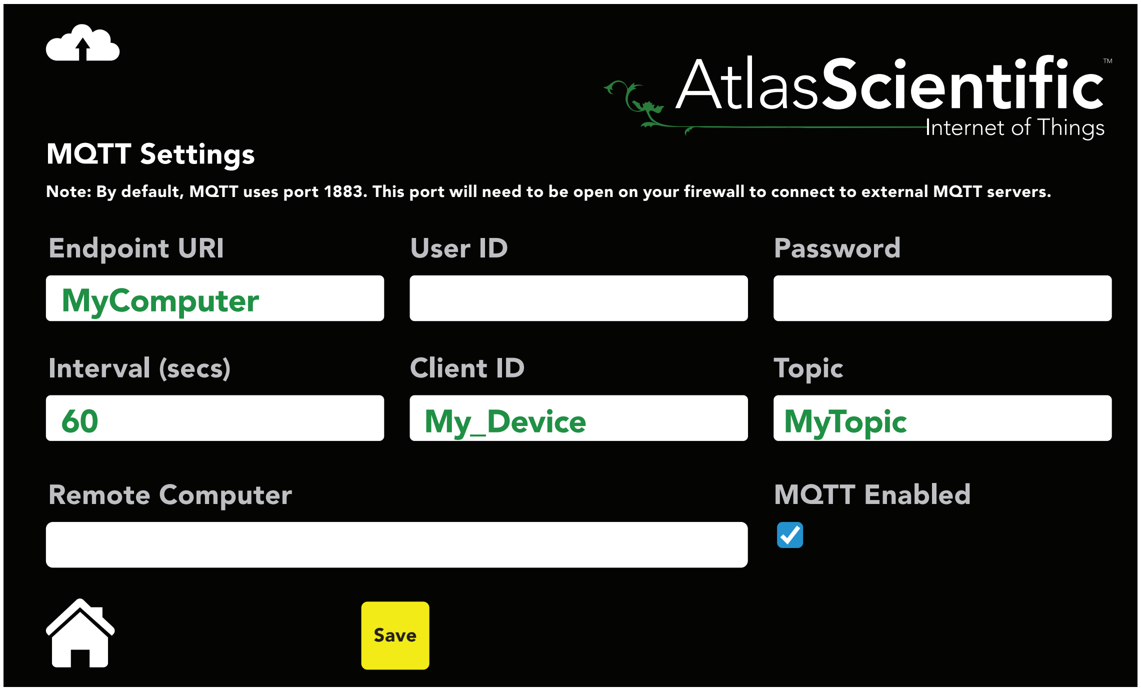
Endpoint URI	MyComputer
User ID	leave blank, if you did not setup a user ID
Password	leave blank, if you did not setup a password
Interval (Secs)	60 is a good place to start
Client ID	My_Device *(No spaces)*
Topic	MyTopic *(I usually do NOT put the starting "/")*

We will append a series of subtopic onto your topic

Example
If you are running an EZO™ RTD Temperature Circuit on the default address, and have named it, mosquitto will receive the topic **MyTopic/RTD/add_102/SensorName**

If the circuit is unnamed (default) it will be **MyTopic/RTD/add_102**

If you have multiple sensors, there will be multiple topics, all underneath **MyTopic**.
You should see them in your mosquitto_sub session



MQTT Settings

Note: By default, MQTT uses port 1883. This port will need to be open on your firewall to connect to external MQTT servers.

Endpoint URI	User ID	Password
MyComputer		
Interval (secs)	Client ID	Topic
60	My_Device	MyTopic
Remote Computer		MQTT Enabled
		☑

MQTT with Mosquitto

Troubleshooting

If we assume that you have installed mosquitto on a computer called **MyComputer**
(see their website https://mosquitto.org for ports to open, etc),

You can monitor all traffic to that broker with a program call **mosquitto_sub**
(comes in the install package) The Manual can be found by clicking HERE

Assuming you have not setup mosquitto for login, the command would be
mosquitto_sub -v -h MyComputer -t '#'

That command subscribes to ALL topics (**'#'**) in a verbose manner.

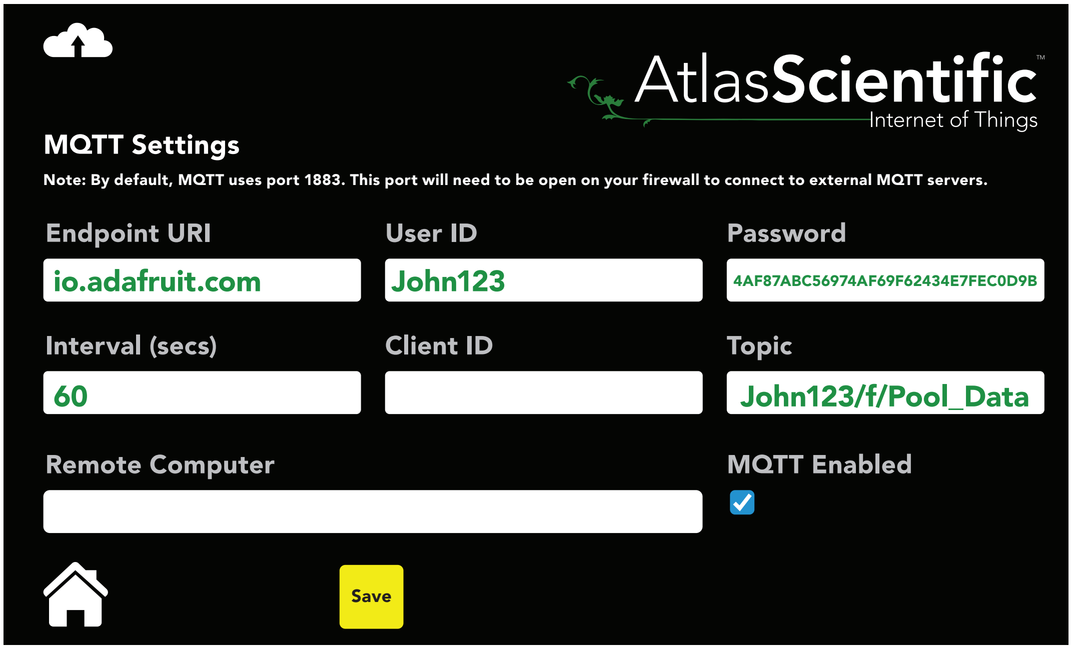
MQTT with io.adafruit.com

Note: By default, MQTT uses port 1883. This port will need to be open on your firewall to connect to external MQTT servers.

io.adafruit.com is a MQTT broker that has some limitations, but is usually quite reliable. That said, it has some differences from stock mosquitto.

Your settings for the Atlas IoT would be

Endpoint URI	io.adafruit.com
User ID	your loginID. ***Example*** John123
Password	This is the Adafruit IO key, a 32digit ID that you generate and download from the adafruit site. ***Example*** 4AF87ABC56974AF69F62434E7FEC0D9B
Interval (Secs)	60 is a good place to start
Client ID	*(Not used for adafruit, leave blank)*
Topic	*This is the name of your data set. (choose a name,* ***Example*** *Pool_Data*
	Format *UserID/f/MyTopicNoSpaces*
	Example *John123/f/Pool_Data*

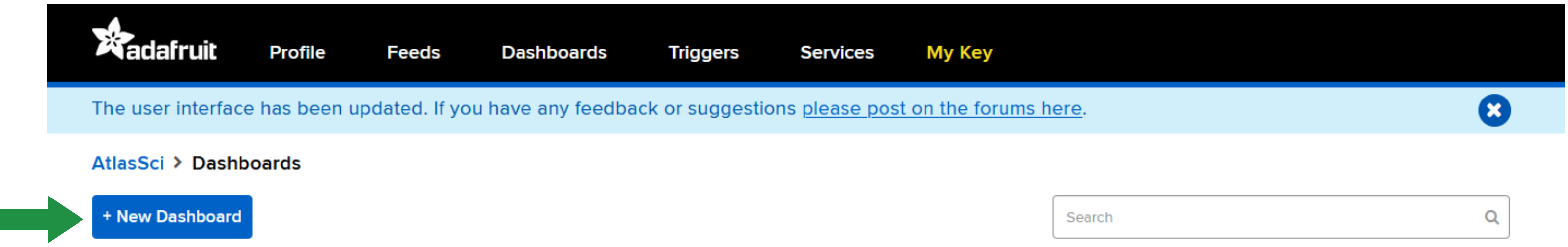


MQTT Settings

Note: By default, MQTT uses port 1883. This port will need to be open on your firewall to connect to external MQTT servers.

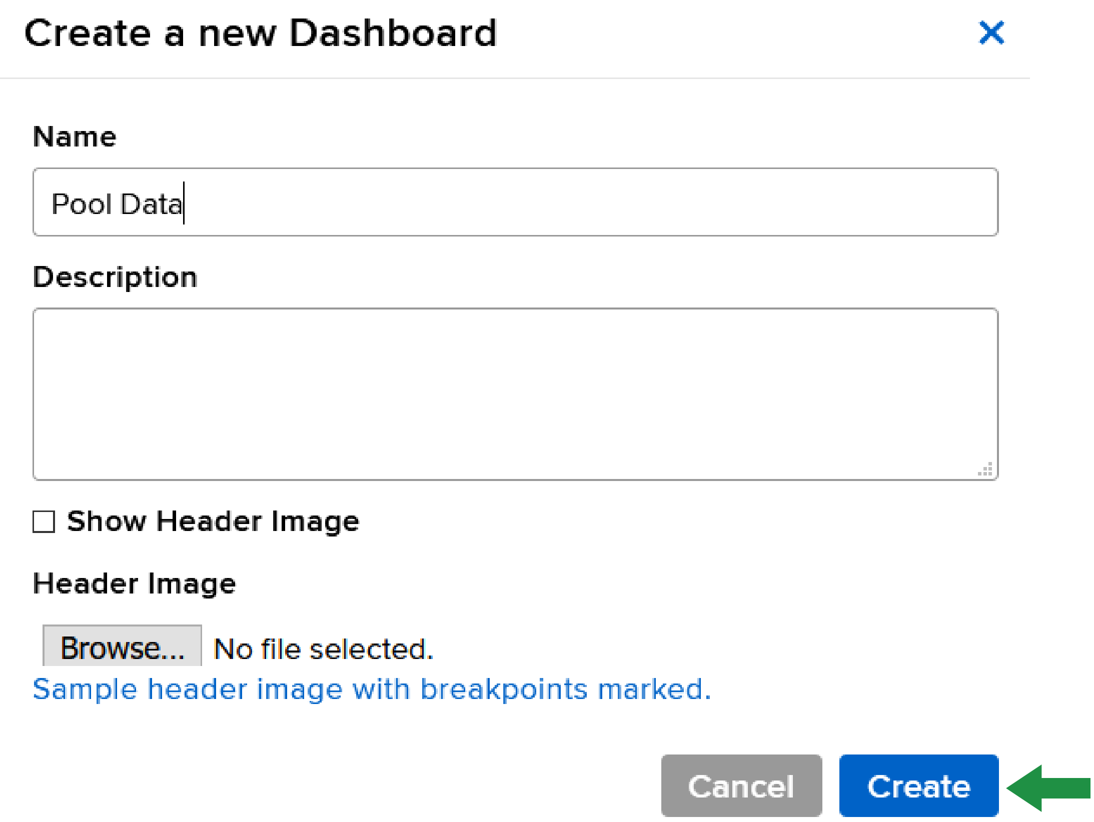
Endpoint URI	User ID	Password
io.adafruit.com	John123	4AF87ABC56974AF69F62434E7FEC0D9B
Interval (secs)	Client ID	Topic
60		John123/f/Pool_Data
Remote Computer		MQTT Enabled
		☑

Once you hit save, go to io.adafruit.com and click on dashboard, + New Dashboard.



The user interface has been updated. If you have any feedback or suggestions please post on the forums here.

Name your dashboard and hit create.



Create a new Dashboard

Name

Pool Data

Description

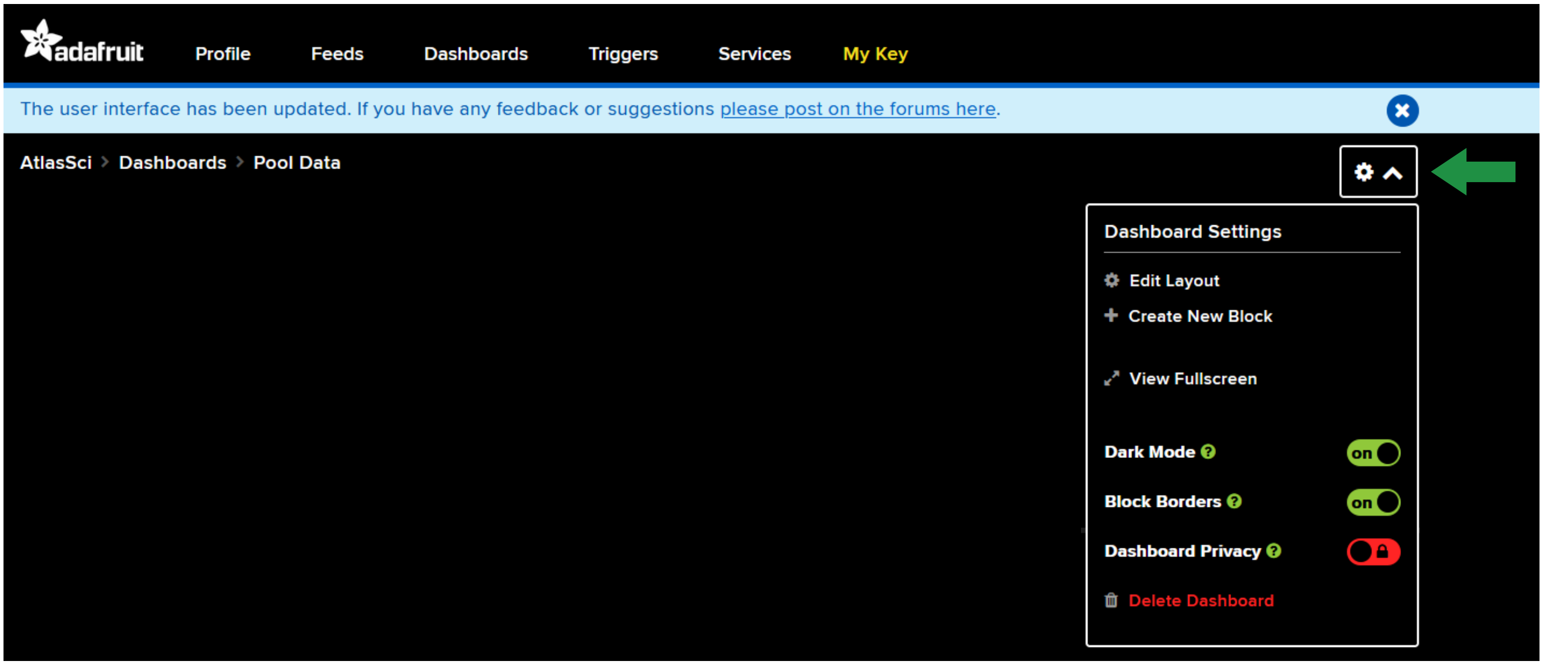
□ Show Header Image

Header Image

No file selected.

Sample header image with breakpoints marked.

Once complete, click on your new dashboard.



The user interface has been updated. If you have any feedback or suggestions please post on the forums here.

AtlasSci › Dashboards › Pool Data

Dashboard Settings

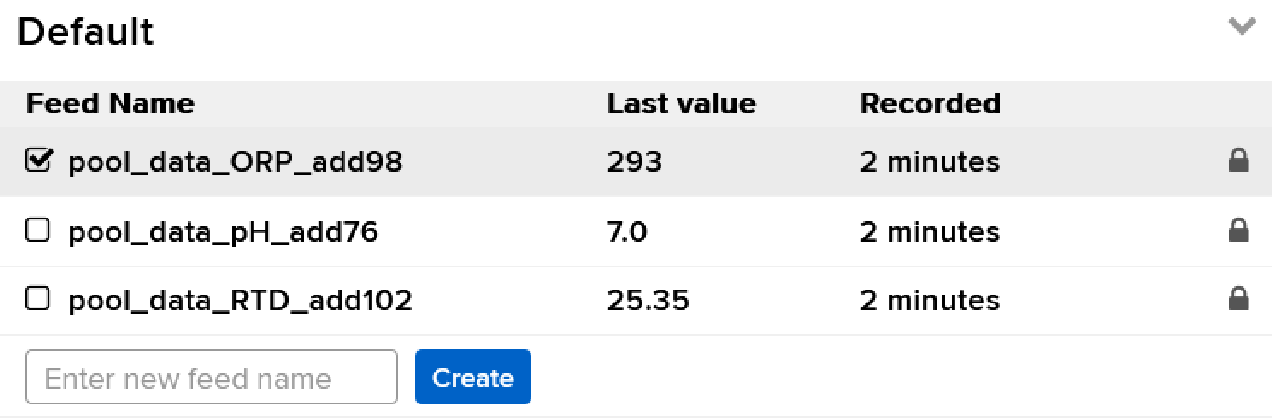
Dark Mode: on

Block Borders: on

Dashboard Privacy: 🔒

Click the gear icon located on the top right; Then click on ***+ Create New Block.***

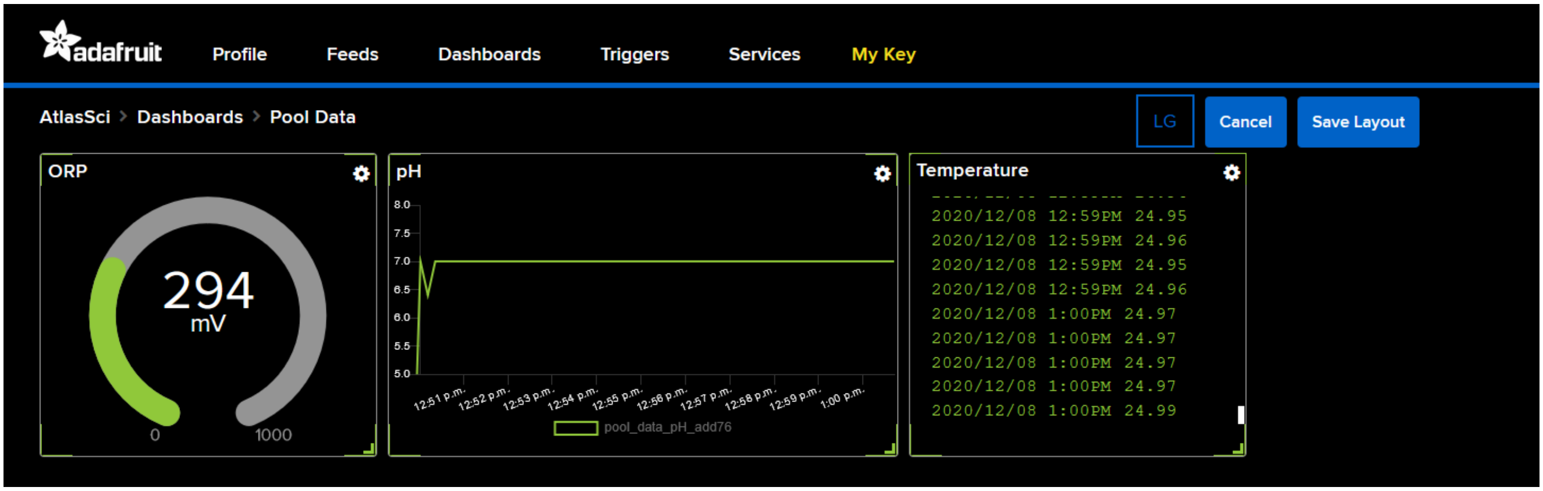
Choose the style you would like to see your data in, and apply the style to your sensor feed.



Default

Feed Name	Last value	Recorded	
☑ pool_data_ORP_add98	293	2 minutes	🔒
□ pool_data_pH_add76	7.0	2 minutes	🔒
□ pool_data_RTD_add102	25.35	2 minutes	🔒

Once complete your dashboard could look like this:



```
2020/12/08 12:59PM 24.95
2020/12/08 12:59PM 24.96
2020/12/08 12:59PM 24.95
2020/12/08 12:59PM 24.96
2020/12/08 1:00PM 24.97
2020/12/08 1:00PM 24.97
2020/12/08 1:00PM 24.97
2020/12/08 1:00PM 24.97
2020/12/08 1:00PM 24.99
```

API Settings

We've added web API support to the Atlas iot™ software.

Simply go into your web browser and enter in:

[Your raspberry pi address]/api/values/

Displayed in your browser will be a JSON containing all of the connected EZO™ circuits and senors, along with their readings, device name, and device addresses.

Example

```
[{"created_at":"2021-07-23T17:08:39.4038617Z","ModuleType":"CO2","value":"1005","Address":77,"
Name":"test)co2"},{"Temperature":24.7,"Humidity":35.25,"THI":69.79,"HeatIndex":75.46,"created_at":"2
021-07-23T17:08:38.8037542Z","ModuleType":"HUM","value":"35.2","Address":111,"Name":""}]
```

From the example above, you can see the readings from two EZO™ sensors:
EZO-CO2™ and **EZO-HUM™**

```
[{"created_at":"2021-07-23T17:08:39.4038617Z","ModuleType":"CO2","value":"1005","Address":77,"
Name":"test)co2"},{"Temperature":24.7,"Humidity":35.25,"THI":69.79,"HeatIndex":75.46,"created_at":"2
021-07-23T17:08:38.8037542Z","ModuleType":"HUM","value":"35.2","Address":111,"Name":""}]
```

You can also display the readings of a specific EZO™ circuit or sensor, by entering their I2C or USB address after the API command in your web browser.

[Your raspberry pi address]/api/values/77

Example

```
[{"created_at":"2021-07-23T17:08:39.4038617Z","ModuleType":"CO2","value":"1005","Address":77,"
Name":"test)co2"}]
```

You will only see the readings from the **EZO-CO2™** as this sensors I2C address is 77.

TLS / SSL Connection

There is new section we've added into the appsettings.JSON file.

```
"MQTT": {
   "CAFile": "",
   "ClientFile": "",
   "SSLProtocol": "None",
   "UseSecure": false,
   "Port": 1883
 }
```

To edit this new section, open the "appsettings.JSON" file via text editor.
If this section is removed entirely, the app will continue to functions as it currently does.

CAFile can be set to the path of your certificate authority file.

ClientFile is set to the path of the Client Key file.

SSLProtocol can be any of the following:

None *(default if an error is made)*
sslv3
tlsv1_0
tlsv1_1
tlsv1_2

UseSecure is set to your UserID and Password.

Port is the TCP/IP port to use.

By default, non TLS/SSL connections use **1883**, secure uses **8883**.