Meshlium Xtreme

Technical Guide

libelium

Document version: v8.1 - December 19, 2018

INDEX

1. General and safety information

Important:

- All documents and any examples they contain are provided as-is and are subject to change without notice. Except to the extent prohibited by law, Libelium makes no express or implied representation or warranty of any kind with regard to the documents, and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose.
- The information on Libelium's websites has been included in good faith for general informational purposes only. It should not be relied upon for any specific purpose and no representation or warranty is given as to its accuracy or completeness.

Read carefully the Limited Warranty and Terms and Conditions of Use before using "Meshlium".

- Read carefully the "General Conditions of Sale and Use of Libelium". This document can be found at: https://www.libelium.com/development/meshlium/technical_service.
- As specified in the Warranty document which you can find at: https://www.libelium.com/development/meshlium/documentation, the client has 7 days from the day the order is received to detect any failure and report that to Libelium. Any other failure reported after these 7 days may not be considered under warranty.
- Do NOT open the enclosure. If you do so, you will lose the guarantee.
- Do not remove any of the components.
- Do not allow contact between metallic objects and the electronic part to avoid injury and burns.
- NEVER immerse the equipment in any liquid.
- Keep the equipment in a dry place away from any liquids that could spill.
- Check from the label that comes with the equipment the maximum permitted voltage and amperage range.
- Keep the equipment within the temperature range indicated in the specifications section.
- Do not connect or power the equipment using cables that have been damaged.
- Place the equipment in an area to which only maintenance personnel can have access (in a restricted access zone). In any case, keep children away from the machine.
- If there is a power failure, immediately disconnect from the mains electricity.
- If using the car lighter as a power source, make sure that you follow the voltage and current specifications indicated in the section "How to use Meshlium".
- If a software failure occurs, consult the section web support.
- Do not place the equipment on trees or plants as they could be damaged by its weight.
- Be particularly careful if you are connected through Ethernet or WiFi; if the settings are incorrectly altered, Meshlium could become inaccessible.

2. Important: read me before using

The following list shows **just some** of the actions that produce **the most common failures and warranty-voiding**. Complete documentation about usage can be found at:

https://www.libelium.com/development/meshlium/technical_service

Failure to comply with the recommendations of use will entail the guarantee cancellation.

- Do not interrupt the power supply before shutting down Meshlium properly through the “Shutdown” or “Restart” buttons in the Manager System.
- Do not open the Meshlium enclosure in any case. This will automatically make the warranty void.
- Do not submerge Meshlium in liquids.
- Do not place Meshlium on places or equipment where the device could be exposed to shocks and/or vibrations.
- Do not expose Meshlium to temperatures below -20°C or above 50°C.
- Meshlium’s microprocessor must not overpass 75 Celsius degrees. The user must ensure that this temperature never overpass. Especially when using WiFi Scan.
- Do not power Meshlium with other power sources than the original provided by Libelium.

For more information: https://www.libelium.com/meshlium

3. Meshlium v4.0 vs Meshlium v3.5

This evolution of Meshlium includes an important upgrade of the hardware capabilities. The most important changes are:

- Big step forward in performance, CPU performance 10 times better and RAM capacity 8 times bigger.
- Cellular connection upgraded to 4G for a very fast Internet connection and data synchronization.
- WiFi AP upgraded to WiFi b/g/n (up to 144 Mbps).
- New models of radio module for the 868 MHz and 900 MHz bands.
- Up to two RF (XBee) modules can be installed in the device, working with the 4G radio at the same time (2.4GHz and 868/900 MHz).
- GPS/GLONASS capabilities for a faster global location.
- New improved design.
- Operating system updated including new versions of programs and system.
- New Meshlium is Microsoft Azure Certified (More info: https://azure.microsoft.com/es-es/marketplace/programs/certified/).

3.1. Capabilities comparison

	Previous Meshlium version	New Meshlium
CPU cores	1	4
CPU architecture	32 bits	64 bits
CPU frequency	500 MHz	1 GHz
RAM	256 MB DDR	2 GB DDR3
Storing	Compact Flash 8 GB	SSD disk 16 GB
Linux Kernel	2.6	3.16
Simultaneous cloud services	2-4	15-20
Boot time	≈ 2 minutes	Less than 1 minute
WiFi	a/b/g (up to 54 Mbps)	a/b/g/n (up to 144 Mbps)
Cellular	Up to 7.2 Mbps downlink (SIM5218) Up to 384 kbps downlink (SIM5215)	Up to 42 Mbps downlink
Antenna connectors	4	6
RF module sockets	1	2
Geolocation	GPS	GPS + GLONASS
Root access	Yes	No
Power consumption	≈ 10 W	≈ 15 W (depending on num. of radios)
Enclosure (mm)	210 x 190 x 55	255 x 225 x 80
Certifications	CE / FCC / IC	CE (Europe) / FCC (US) / IC (Canada) / ANATEL (Brazil) / RCM (Australia) / PTCRB (US) / AT&T (US)

3.2. Compatibility with Waspmote and Plug & Sense! nodes

Old hardware	Compatible	Notes
Plug & Sense! (Waspmote v1.2) 802.15.4	YES	
Plug & Sense! (Waspmote v1.2) ZigBee	NO	Old ZigBee modules are EoL
Plug & Sense! (Waspmote v1.2) DigiMesh	NO	
Plug & Sense! (Waspmote v1.2) 900	NO	Old 900 MHz modules are EoL. Substituted by the new 900HP radios.
Plug & Sense! (Waspmote v1.2) 868	NO	Old 868 MHz modules are EoL. Substituted by the new 868 radios.
Plug & Sense! (Waspmote v1.2) WiFi	YES	
Plug & Sense! (Waspmote v1.2) 3G	YES	
Plug & Sense! (Waspmote v1.5) 802.15.4	YES	
Plug & Sense! (Waspmote v1.5) 900	YES	
Plug & Sense! (Waspmote v1.5) 868	YES	
Plug & Sense! (Waspmote v1.5) WiFi	YES	
Plug & Sense! (Waspmote v1.5) 4G	YES	
Plug & Sense! (Waspmote v1.5) ZigBee	NO	Meshlium does not support this RF module

3.3. Compatibility with current cloud software

Cloud software	Compatible	Notes
Amazon IoT	yes	
Esri	yes	Only ArcGIS online
IBM Bluemix	yes	
IOT-Ticket	yes	
Azure Event Hubs	yes	
Azure Service Bus	no	Obsolete: use Event Hubs
MQTT	yes	
Telefonica	yes	
ThingWorx	yes	
amplia	yes	
Simfony	yes	
Smart City Platform	yes	
B-Scada	yes	
DeviceLynk	no	Obsolete
devicify	yes	
Eagle.io	yes	
ElementBlue	yes	
Extunda	yes	
IoTSens	yes	
Sentilo	yes	
Sofia2	yes	
Solvver	no	Obsolete
Thing+	yes	

Compatibility with other software:

Software	Compatible	Notes
External DB synchronization of sensor data	yes	Some changes in the tables needed, can be done without losing data

3.4. XBee-PRO 868 vs XBee 868LP

The new XBee 868LP module supports some changes:

- The new XBee 868LP operates between 863 and 870 MHz, making it deployable in several regions throughout the world including approved European countries and India by utilizing a software selectable channel masking feature.
- The XBee 868LP is also the industry's first RF module using 868 MHz and surrounding frequencies for LBT + AFA (Listen Before Talk and Adaptive Frequency Agility). This virtually eliminates interference by listening to the radio environment before any transmission starts, and automatically shifting to a new channel when interference is detected. This patent-pending frequency scan occurs automatically and in a matter of microseconds so as not to impact performance.

Features comparison:

Feature	[Old] XBee-PRO 868	[New] XBee 868LP
Frequency band	868 MHz (1 channel)	863 to 870 MHz (32 channels)
RF data rate	24 kbps	10 kbps
Indoor/urban range	Up to 550 m	Up to 112 m
Outdoor/line-of-sight range	Up to 40 km	Up to 8.4 km
Transmit power	25 dBm	14 dBm
Receive sensitivity	-112 dBm	-106 dBm
Transmit current	500 mA	48 mA
Receive current	65 mA	27 mA
LBT + AFA	No	Yes

3.5. XBee-PRO 900 vs XBee-PRO 900HP

The new XBee 900HP modules support some changes:

- The new XBee-PRO 900HP uses greater power transmission compared to the old version. Thus, the ranges achieved by these new modules are larger than before.
- The XBee-PRO 900HP modules are certified for use in multiple countries: Brazil, Australia, US. Through the new channel selection it is possible to enable/disable the preferred frequency channels within the 902-928 MHz band.
- The power consumption has been improved compared to the old modules. Better ranges have been achieved with almost the same TX power. On the other hand, RX power consumption has been reduced.

Features comparison:

Feature	[Old] XBee-PRO 900	[New] XBee-PRO 900HP
Frequency band	902-928 MHz (8 hopping patterns on 12 channels)	902 MHz to 928 MHz (64 channels)
RF data rate	156 kbps	10 kbps
Indoor/urban range	Up to 450 ft (140 m)	Up to 2000 ft (610 m)
Outdoor/line-of-sight range	Up to 1.8 miles (3 km)	Up to 9 miles (15.5 km)
Transmit power	17 dBm	24 dBm
Receive sensitivity	-100 dBm	-110 dBm
Transmit current	210 mA	215 mA
Receive current	80 mA	29 mA

3.6. 3G (SIM5215) vs 4G (LE910)

The new 4G module supports some changes:

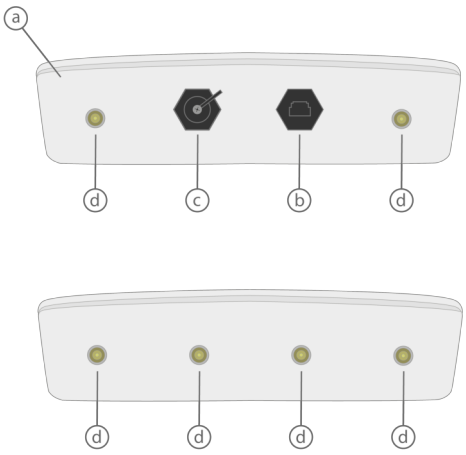
- The new 4G counts with many different models, one specifically designed for each market:
  - LE910-EU (Europe / Brazil): CE, GCF, ANATEL.
  - LE910-NAG (US / Canada): FCC, IC, PTCRB, AT&T Approved.
  - LE910-AU V2 (Australia): RCM, Telstra Approved.
- The GPS module also makes possible perform geolocation services using NMEA sentences offering information such as latitude, longitude, altitude and speed what makes it perfect to perform tracking applications.
- The new 4G module offers the maximum performance of the 4G network as it uses two different antennas (normal + diversity) for RX (MIMO DL 2x2) choosing the best received signal at any time and getting a maximum download speed of 100 Mbps.

Features comparison:

Features	[Old] 3G (SIM5215)	[New] 4G (LE910)
Protocols	3G / GPRS / GSM	4G / 3G / GPRS / GSM / WCDMA / HSPA+ / LTE
Certifications	CE, GCF, FCC, IC, PTCRB	CE, GCF, ANATEL, FCC, IC, PTCRB, AT&T Compliant, KCC, RCM, NTT DoCoMo, KDDi
GPS	No	Yes
Download max speed	384 kbps	100 Mbps
Upload max speed	384 kbps	50 Mbps
Antenna diversity	No	Yes
Cellular carriers	Any	Any + Specially tested with AT&T, SK Telecom, Telstra, NTT DoCoMo or KDDi

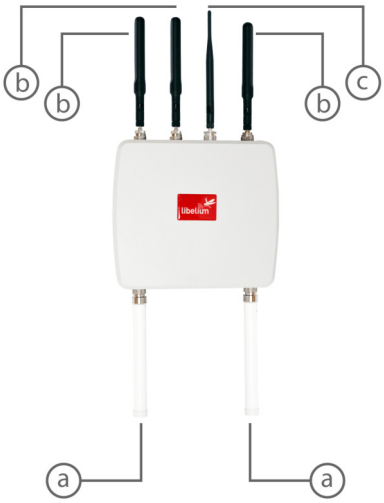
4. Contents of the box

Meshlium



- **a)** IP67 casing.
- **b)** Ethernet connector.
- **c)** nano-SIM + micro-USB connector.
- **d)** Antenna connectors.

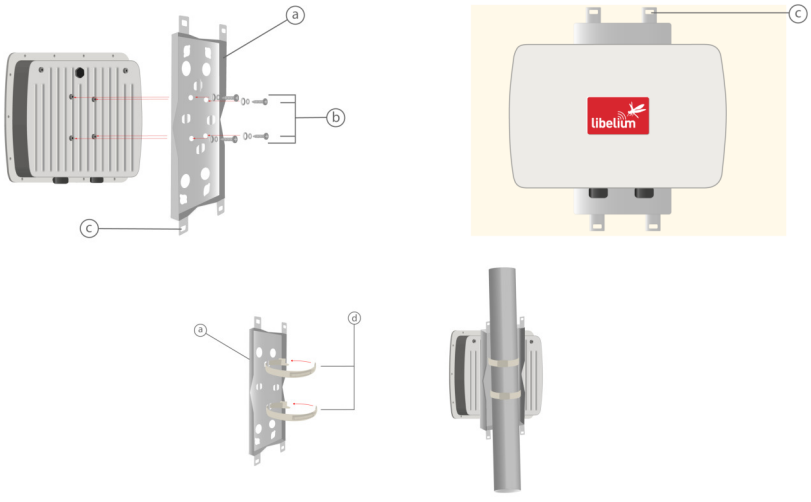
Antennas*



- **a)** Dipole 5 dBi (Bluetooth, WiFi, XBee-PRO 802.15.4).
- **b)** 4G / GPS (3 antennas for EU, US or BR models; 2 antennas for AU models).
- **c)** Dipole 4.5 dBi (XBee 868LP, XBee-PRO 900HP).

(*) Number and type of antennas depend on the model purchased.

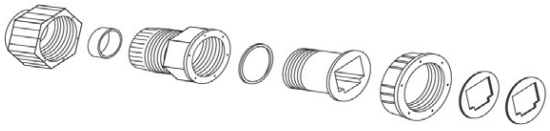
Fixing



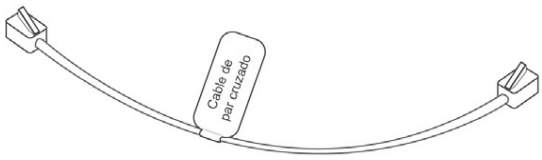
- **a)** Bracket plate.
- **b)** Mounting screws.
- **c)** 4 screw holes for wall.
- **d)** 2 worm-drive clamps.

Ethernet cable

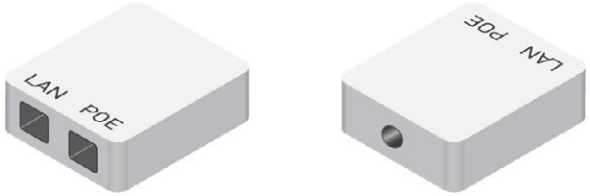
IP67 Ethernet cap



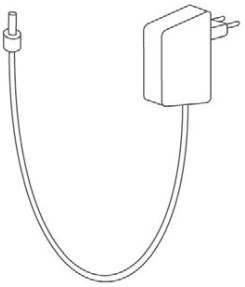
Ethernet crossover cable



POE injector



Ethernet cable



5. Specifications

Figure : Meshlium unit

Processor	1 GHz Quad Core (x86)
RAM memory	2 GB (DDR3)
Disk memory	16 GB
Power	6 to 12 W (12 V)
Power source	PoE (Power Over Ethernet)
Max current supply	2 A
Enclosure	Material: Aluminum Dimensions: 255 x 225 x 80mm Weight: 1.9 kg External protection: IP67
Temperature range	-20°C / 50°C
Response time to Ethernet ping	60 s
Time to have all the services running	60 s
Types of power supply*	AC-220 V (DC-12 V)
System	Linux, Debian based
Management software	Meshlium Manager System (open source)
Security	Authentication WPA, WPA2, HTTPS

(*) Only with the accessories supplied by Libelium.

WiFi (2.4 GHz) radio (Access Point/Scanner)

WiFi radio	
Chipset	Qualcomm Atheros QCA9882
TX power	20 dBm
Range	500 m*
Antenna 5dBi dipole	
Type	Omni-directional, dipole
Gain	5 dBi
Dimensions	224 x 22 mm

(*) Depending on antenna and line of sight.

RF radio modules

Model	XBee-PRO 802.15.4
Frequency	2.4 GHz
TX power	18 dBm (10 dBm for EU models)
Rx sensitivity	-100 dBm
Antenna	5 dBi dipole
Range	1.6 km (750 m in EU models)*

Model	XBee 868LP
Frequency	868 MHz
TX power	14 mW
Rx sensitivity	-106 dBm
Antenna	4.5 dBi dipole
Range	8.4 km*

Model	XBee-PRO 900HP
Frequency	900 MHz
TX power	24 dBm
Rx sensitivity	-110 dBm
Antenna	4.5 dBi dipole
Range	15.5 km*

(*) Depending on antenna and line of sight.

4G/LTE module

Protocols	4G, LTE, 3G, WCDMA, HSPA, UMTS, GPRS, GSM
Frequency bands EU/BR version	LTE - 800 (B20) / 1800 (B3) / 2600 (B7) UMTS - 850 (B5) / 900 (B8) / 2100 (B1) GSM/GPRS - 900 /1800
Frequency bands US version	LTE - 700 (B17) / 850 (B5) / AWS1700 (B4) / 1900 (B2) UMTS - 850 (B5) / 1900 (B2) GSM/GPRS - 850 / 1900
Frequency bands AU version	LTE - 700 (B17) / 1800 (B3) / 2600 (B7) (AU models do not support 3G, GPRS or GSM)
Output power	Class 4 (2 W, 33 dBm) @ GSM 850/900 Class 1 (1 W, 30 dBm) @ GSM 1800/1900 Class E2 (0.5 W, 27 dBm)@ EDGE 850/900 Class E2 (0.4 W, 26 dBm)@ EDGE 1800/1900 Class 3 (0.25 W, 24 dBm) @ UMTS Class 3 (0.2 W, 23 dBm) @ LTE
RX rate	Up to 100 Mb/s
TX rate	Up to 50 Mb/s
Antenna	4 dBi
SIM card	Access via the External nano-SIM socket

GPS Module

Modes*	Assisted GPS (A-GPS), Standalone mode (NMEA frames)
Antenna	4 dBi

(*) The AU models do not have a GPS receiver.

Bluetooth Scanner

Protocol	Bluetooth 2.1 + EDR Class 2
TX power	3 dBm
Antenna	5 dBi dipole
Range	20-30 m*

(*) The AU models do not have a GPS receiver.

6. How to use Meshlium

6.1. Power supply

Meshlium needs a 220 V power connection. The device must be powered with the power source provided by Libelium.

How to connect Meshlium to 220 V (110 V compatible):

1. Unscrew the Ethernet connector cap in Meshlium.
2. Join the end that has the IP67 protection of the Ethernet cable to the connector and screw the cap on to fix it.
3. Connect the free end of the cable to the PoE injector input marked as PoE. As explained in the section "Before using Meshlium", make sure that the PoE is indoors.
4. Take the supplied power adapter and plug it into the corresponding PoE injector connector, labeled as DC.
5. Plug the other end of the adapter into the 220 V socket and your Meshlium is now ready to operate.

***Note**: For equipment powered by an electric outlet, a power outlet must be installed near the equipment, and it must be easily accessible.*

***Note**: To avoid electrical arcs which could damage the equipment, we advise to follow the order described: just connect the AC/DC adapter to the PoE (4) before you plug it into 220 V mains (5).*

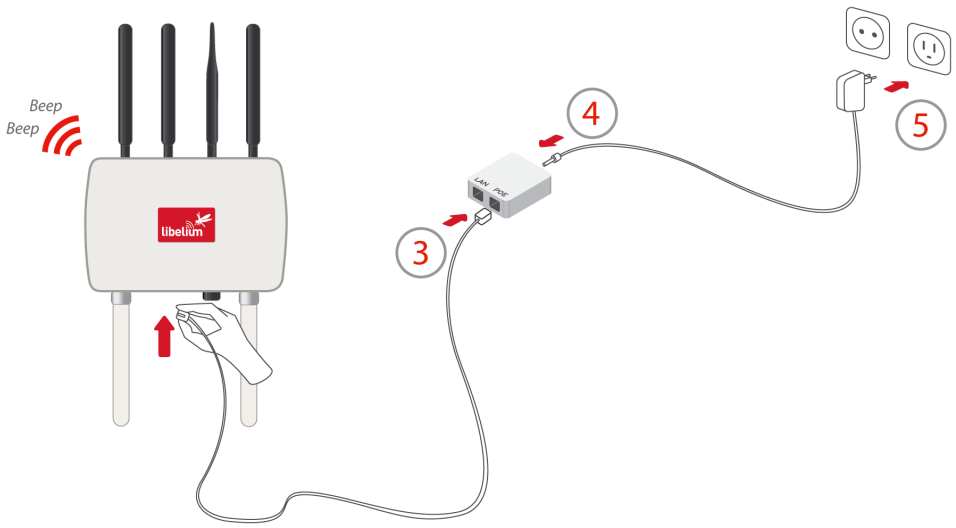


Figure : Connecting Meshlium to 220 V

Important:
The nominal voltage for Meshlium is 12 V. Use only the elements provided by Libelium. Specifically, note that other power inputs will damage the device: other PoE systems may be 24 or 48 V, so they will destroy Meshlium.

How to connect Meshlium in order to get access by the Ethernet interface:

1. Connect the network crossover cable (it has an identifying label) included in the box to the PoE injector input marked LAN and to the network socket of your PC as shown in the diagram*.

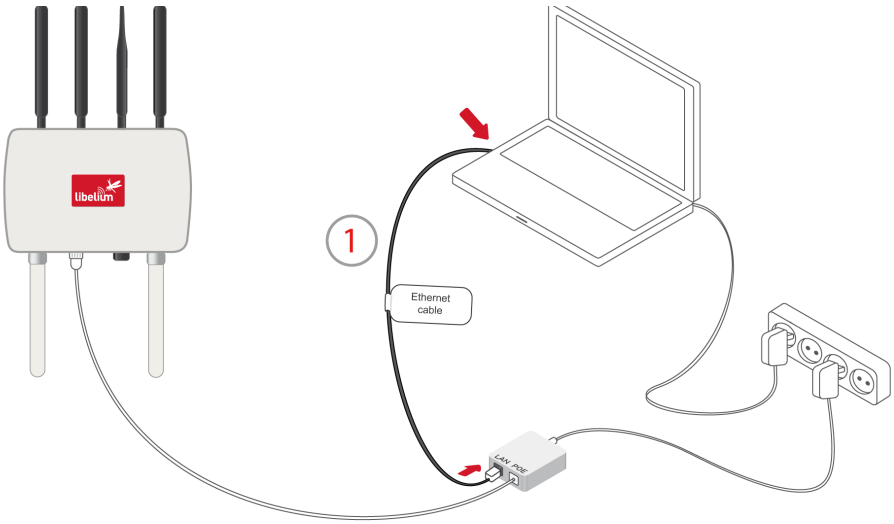


Figure : Connecting LAN cable to a PC

You can also carry out this connection through a switch (not supplied with Meshlium):

1. Connect the Ethernet cable (not the crossover) to the PoE input marked LAN and to one of the switch inputs.
2. Connect another Ethernet cable to another one of the switch inputs and the opposite end to the network socket of your PC*.

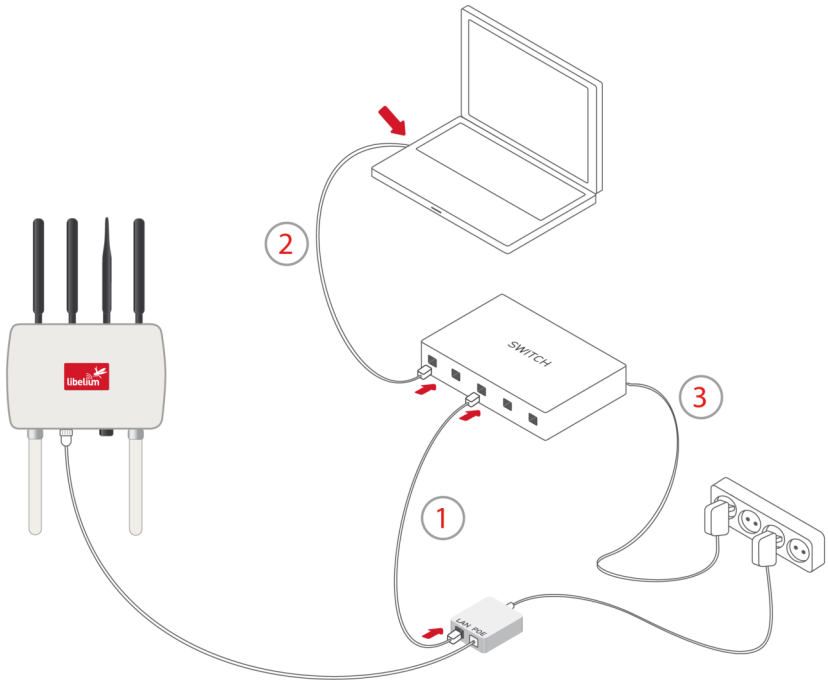


Figure : Connecting LAN cable to a switch

(*) See the "Accessing Meshlium - make it easy!" section in order to see how to get access via wireless.

6.2. External SIM/USB socket

The External SIM socket is composed of 2 connectors:

- nano-SIM card.
- micro-USB (type B

Figure : External SIM socket in a Meshlium with 4G/3G/GPRS/GSM module

The nano-SIM card connector allows the user to connect the SIM card. You can ask your Mobile Network Operator for a nano-SIM card.

The nano-SIM card connector has a push-push mechanism, so it is really easy to remove the card using one nail or a small tool. To insert the SIM, press until a click is heard. To release the card, press until a click is heard and the spring will push the card free.

Figure : Push-push mechanism External SIM/USB socket

Please mind the correct orientation of the nano-SIM card: the side of the chip must look towards the micro-USB connector, and the 45°-angled corner must face the device.

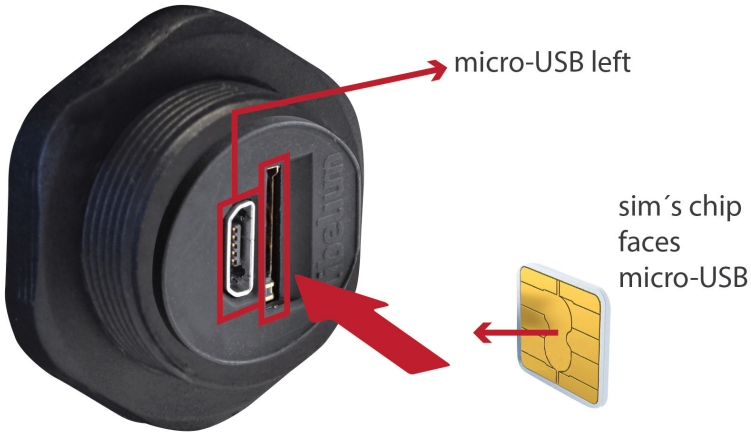


Figure : Correct orientation of the nano-SIM card

It is very important to turn off Meshlium in a secure way before inserting a nano-SIM card, or removing an existing SIM card. The user can damage the device if this operation is done with the device on.

Make sure the External SIM/USB socket is closed with its protection cap tightly screwed before an outdoors deployment.

The operation with the micro-USB socket is just the same than with a normal USB socket. A USB OTG cable can be used to plug in standard A USB connector (like pendrives).

Take into account that the External SIM/USB socket has a limited resistance so please be gentle and push with care.

***Note**: From February 2018, Libelium has redesigned the External SIM/USB Socket, now it is more resistant and we have updated it using the most popular SIM card standard, nano-SIM.*

6.3. How to install the antennas

Every antenna for each technology has a defined position in which it has to be installed. The different positions are:

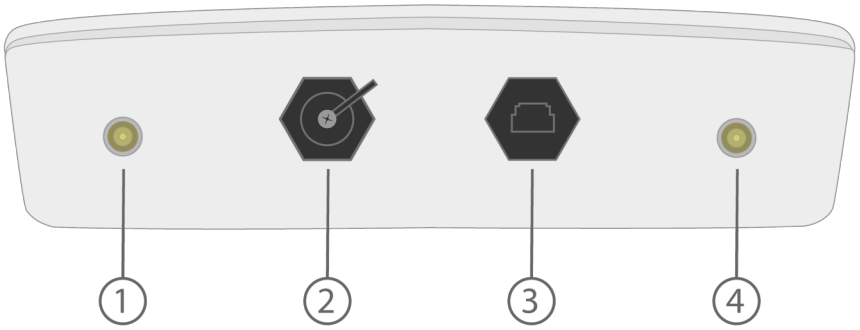


Figure : Antenna socket numbers (front)

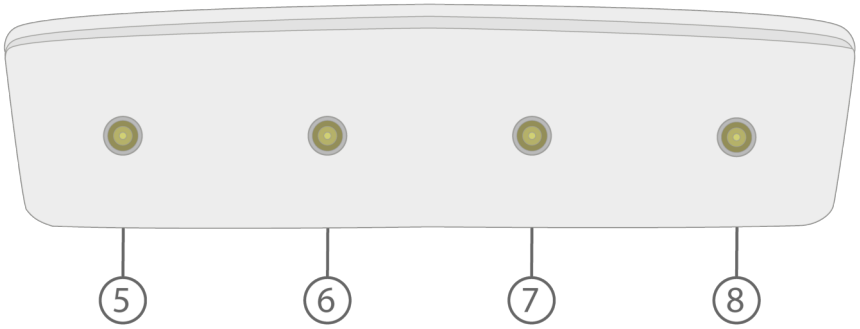


Figure : Antenna socket numbers (rear), check the antenna order

- WiFi AP – Socket 1 (always present).
- 4G with adapter n-to-SMA – Socket 5 and 8 (always present).
- GPS with adapter n-to-SMA – Socket 7.
- RF module 802.15.4 – Socket 4.
- RF module 868 MHz / 900 MHz with adapter n-to-RP-SMA – Socket 6.
- Bluetooth Scanner – Socket 4.
- WiFi Scanner – Socket 6.

The antennas have to be gently screwed on the connector. Do not force the antenna, if you need too much strength to screw it is probably being installed in a wrong position.

If you have any reception issue with 4G or GPS, you can try bending the affected antenna in order to improve isolation.

6.4. Installation of the IP67 Ethernet cable

Installation of the IP67 cap:

In order to install the IP67 cap you will need a connector-free RJ45 cable. This cable is NOT included in the Meshlium box.

Important: Make sure that you have a cable long enough to connect Meshlium from its definitive location to the PoE located indoors. It is not recommended to install Meshlium too far from the PoE injector due to the power loss in the cable. Always test the device with a cable of the same length before installing.

The Ethernet cable can be used for indoors and outdoors deployments. Just note that its resistance is limited, so in order to maximize its lifetime in harsh conditions (direct sunlight, extreme temperatures, very wet climate), we advise to protect the cable with some isolating tube or heat-shrink sleeve. This is also important for installations where insects, birds, rats or other animals could try to bite the cable.

1. Take from the Meshlium box the bag containing the parts for installing the IP67 cap. Check that you have all the parts that appear in the picture.

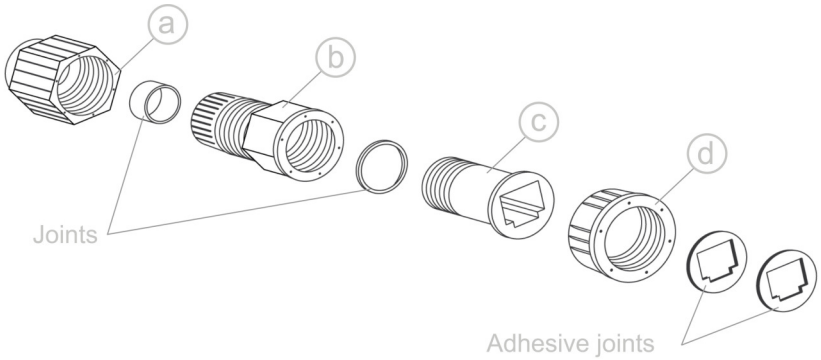


Figure : Cap parts

2. Stick one of the supplied adhesive joints to part C.

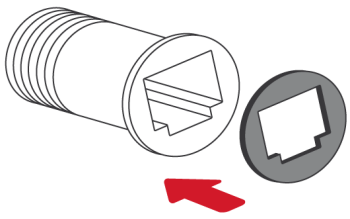


Figure : Stick joint

3. Introduce the joints into part B as shown in the drawing.

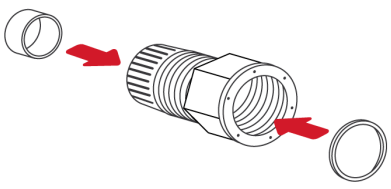


Figure : Introduce joints

4. Insert part C into part D.

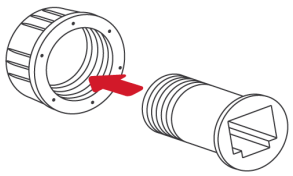


Figure : Insert part C

5. Screw both sets of parts in the direction shown in the diagram.

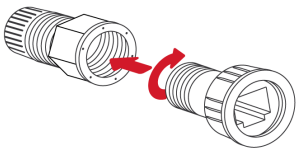


Figure : Screw both parts

6. Partially screw part D to the end.

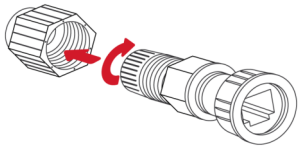


Figure : Screw part D

7. Pass the cable through the fitted cap.

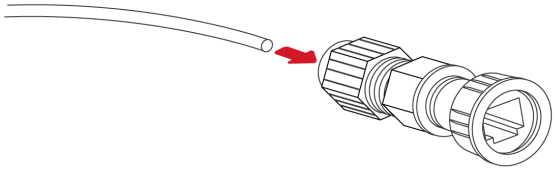


Figure : Pass the cable

8. Crimp the RJ45 connectors at the ends of the cable (the crimping tool is not supplied with Meshlium).

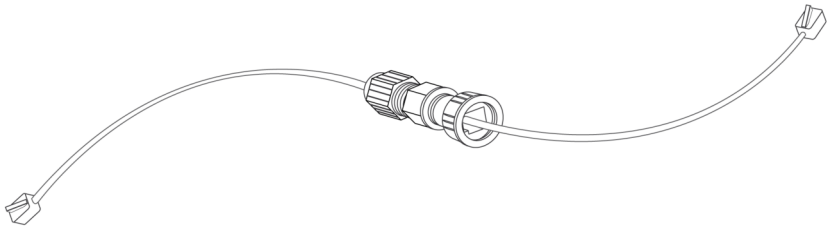


Figure : Crimp RJ45

Your IP67 Ethernet cable is now ready for use.

How to connect the IP67 Ethernet cable to Meshlium:

1. Take the adhesive joint that has not been used for fitting the cap and stick it to the Meshlium Ethernet connector.

Figure : Stick joint

2. Connect the end of the Ethernet cable to the Meshlium Ethernet socket.

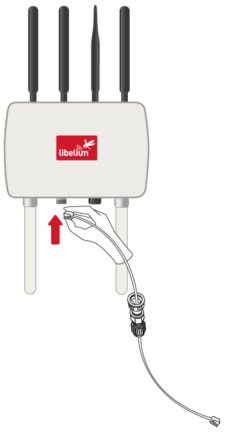


Figure : Connect RJ45

3. Screw part C onto the Meshlium connector. Screw tighter part D to fix the cable too. Your Meshlium is now ready to work outdoors.

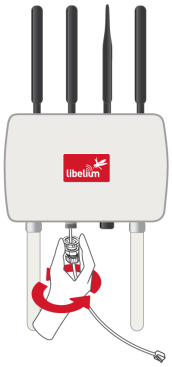
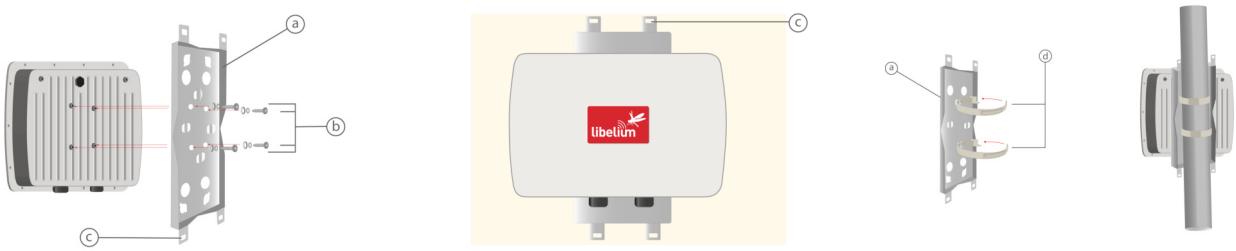


Figure : Screw connector and tighten part D

6.5. Installing Meshlium

Meshlium has been designed to operate in a vertical position with the 2 plastic connectors facing down. You will find the required bracket to mount Meshlium in a pole or in a wall.



Fix the bracket to the Meshlium:

1. Set the bracket **(a)** in the back of the Meshlium with the screw holes for wall facing out **(c)**.
2. Secure the 4 mounting screws **(b)**.

To fix the bracket to a wall:

1. Attach the bracket to the wall securing the screws in the screw holes for wall **(c)**. Screws provided are for general use and could not be valid for every surface. Use hardware adapted for the surface you are installing Meshlium on.

To fix the bracket to a mast:

1. Feed the supplied worm-drive clamps **(d)** through the bracket **(c)** and around the mast. Worm-drive clamps provided are for 78-101mm circumference masts. If your mast is thicker, use clamps with the proper metric.
2. Tighten the worm-drive clamps with a flat head screwdriver.

Finalize the installation:

1. Secure the Ethernet cable to avoid accidental pulls, do not let it loose. If the cable gets stretched, the joint of the cable with its connector could be damaged.

6.6. Initialization, restart and shutdown

In order to allow Meshlium to close correctly all the daemons and applications it is important to use the buttons **Restart** and **Shutdown** placed in the upper right corner in the Manager System. This way you will keep maximum the performance and lifetime of the system.

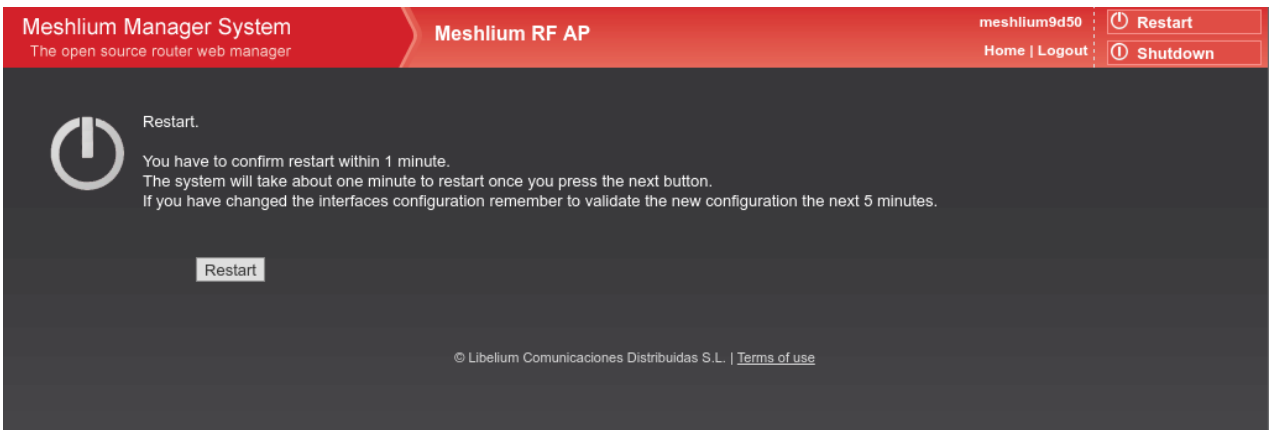


Figure : Restart screen

Once you click on the **Restart** or **Shutdown** button of manager system you have one minute to confirm the operation. If you do not confirm in that time, you will need to click in the button again to perform the operation.

Beep! System

Meshlium includes an internal speaker which will emit “beep!” sounds when initializing, rebooting and shutting down in order to inform about the state of the process.

Initialization beeps:

- Long beep when Meshlium has finished starting and it is ready to be used.

Reboot beeps:

- Long beep when the reboot order is executed.
- Initialization beeps when Meshlium starts again.
- Do not remove the power cable during this process is carried out.

Shutting-down beeps:

- This process could take up to one minute.
- Long beep when Meshlium is about to shut down. A few seconds after the beeps, Meshlium can be unplugged.
- Do not remove the power cable until this process is totally completed.

***Note**: The “beep!” sound is not really loud so you will have to take attention and be close to the Meshlium box in order to hear them clearly.*

***Note**: If Meshlium is unplugged before the acoustic signal of shutdown, internal memory could be damaged. Be sure to wait for several minutes if you are not sure the beeps sounded.*

***Note**: The duration of the reboot or shut-down processes may vary. Make sure you heard the corresponding beeps and be patient.*

***Note**: If the user does not follow these instructions, the risk is very high. Meshlium will become unresponsive and inaccessible. This problem is out of the warranty scope, because it is produced by bad use. The only possible solution will be a repair process in Libelium’s facilities, paid by the user.*

6.7. Setting the time

In order to get all the data stored in the Meshlium local database with the right timestamp, you must adjust the System time.

To do so, go to the Time synchronization section, inside the System Information chapter in the current guide.

6.8. Accessing to the logs

The different processes running on Meshlium produce logs that are self-maintained (the user does not need to delete them).

The user can obtain the logs by connecting to the Meshlium using an FTP client and these credentials:

- user: **log**
- password: **libelium2014**

7. Understanding Meshlium

7.1. Concepts

Meshlium is an IoT gateway that may contain up to 4 different radio interfaces: a WiFi 2.4 GHz (Access Point), a 4G/3G/GPRS/GSM and 2 XBee/RF radios. Meshlium also integrates a GPS module for mobile and vehicular applications and may include Bluetooth and WiFi radios too for scanning applications. These features along with an aluminum IP67 enclosure allows Meshlium to be placed outdoors.

Meshlium can work as:

- an RF (XBee) to Ethernet router for Waspmote nodes*.
- an RF (XBee) to 4G/3G/GPRS/GSM router for Waspmote nodes*.
- a WiFi Access Point.
- a WiFi to 4G/3G/GPRS/GSM router.
- a GPS – 4G/3G/GPRS/GSM real-time tracker.
- a smartphone scanner (detects iPhone and Android devices).

* More info about Waspmote at: https://www.libelium.com/waspmote

All the networking options can be controlled from the **Manager System**, a web interface which comes with Meshlium. It allows you to control all the interfaces and system options in a secure, easy and quick way.

7.2. Meshlium models

There are different Meshlium models depending on the radios integrated:

Meshlium model	Ethernet	WiFi AP	4G/3G/GPRS /GSM	802.15.4	868/900	WiFi & Bluetooth scanners
Meshlium 4G 802.15.4 AP 868 EU	✓	✓	EU/BR version	EU version	868	
Meshlium 4G 802.15.4 AP 900 US	✓	✓	US version	World version	900 US	
Meshlium 4G 802.15.4 AP 900 BR	✓	✓	EU/BR version	World version	900 BR	
Meshlium 4G 802.15.4 AP 900 AU	✓	✓	AU version	World version	900 AU	
Meshlium 4G AP 868 EU	✓	✓	EU/BR version		868	
Meshlium 4G AP 900 US	✓	✓	US version		900 US	
Meshlium 4G AP 900 BR	✓	✓	EU/BR version		900 BR	
Meshlium 4G AP 900 AU	✓	✓	AU version		900 AU	
Meshlium 4G 802.15.4 AP EU	✓	✓	EU/BR version	EU version		
Meshlium 4G 802.15.4 AP US	✓	✓	US version	World version		
Meshlium 4G 802.15.4 AP BR	✓	✓	EU/BR version	World version		
Meshlium 4G 802.15.4 AP AU	✓	✓	AU version	World version		
Meshlium 4G AP Scanner EU/BR	✓	✓	EU/BR version			✓
Meshlium 4G AP Scanner US	✓	✓	US version			✓
Meshlium 4G AP Scanner AU	✓	✓	AU version			✓

Each model with RF modules can have XBee-PRO 802.15.4 and XBee 868LP or XBee-PRO 900HP (depending on the region).

7.3. Storage

The size of the Meshlium hard disk is 16 GB. The Operating System and the Manager System take ≈ 3 GB. This means the space which can be used to store the data captured and to be used by the applications loaded by the user is:

- 16 GB - 3 GB = 13 GB

Some of this space (7.2 GB) is assigned to the user partition: **"/mnt/user"**.

The local database files can be found in: **"/mnt/user/mysql/MeshliumDB"**.

7.4. Application model by model

Meshlium RF 4G AP

Meshlium can take the sensor data which comes from a Wireless Sensor Network (WSN) made with Waspmote sensor devices equipped with RF (XBee) radios and send it to the Internet using the Ethernet interface or the 4G/3G/GPRS/GSM interface. Besides, Waspmotes with GPRS, GPRS+GPS, 3G, 4G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol. Users can connect directly to Meshlium using the WiFi interface to control it and access to the sensor data. Users can also connect to Meshlium via WiFi with laptops and smart phones and get access to the Internet (as a common Access Point).

(*) https://www.libelium.com/waspmote

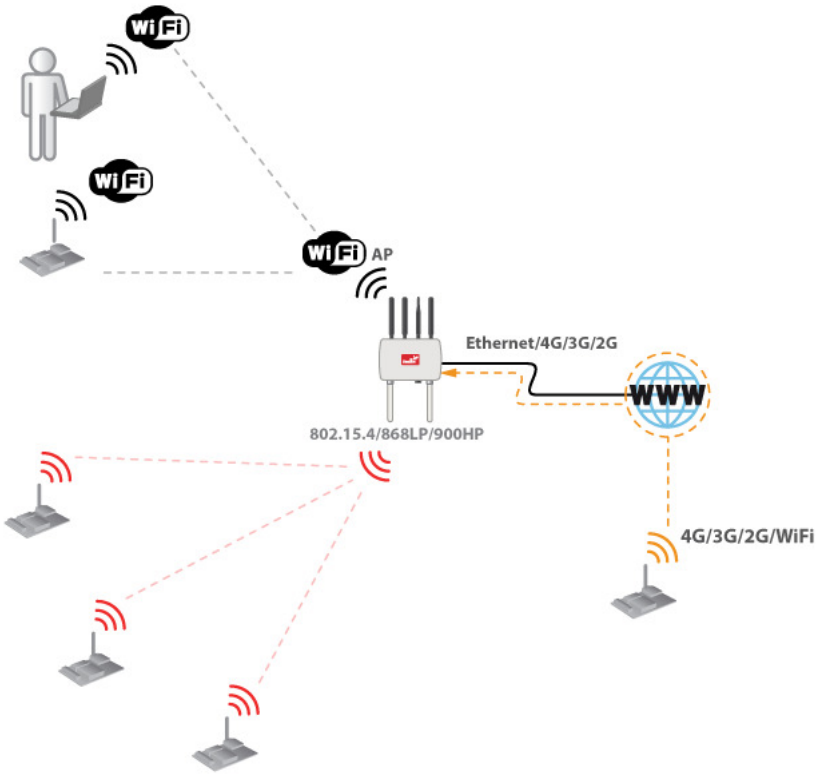


Figure : Meshlium RF 4G AP

Meshlium Scanner 4G AP

It allows to detect Smartphones (iPhone, Android) and in general any device which works with **WiFi** or **Bluetooth** interfaces. The collected data can be send to the Internet by using the Ethernet interface or the 4G/3G/GPRS/GSM connectivity. Besides, Waspmotes with GPRS, GPRS+GPS, 3G, 4G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol. Users can connect directly to Meshlium using the WiFi interface to control it and access to the sensor data. Users can also connect to Meshlium via WiFi with laptops and smart phones and get access to the Internet (as a common Access Point).

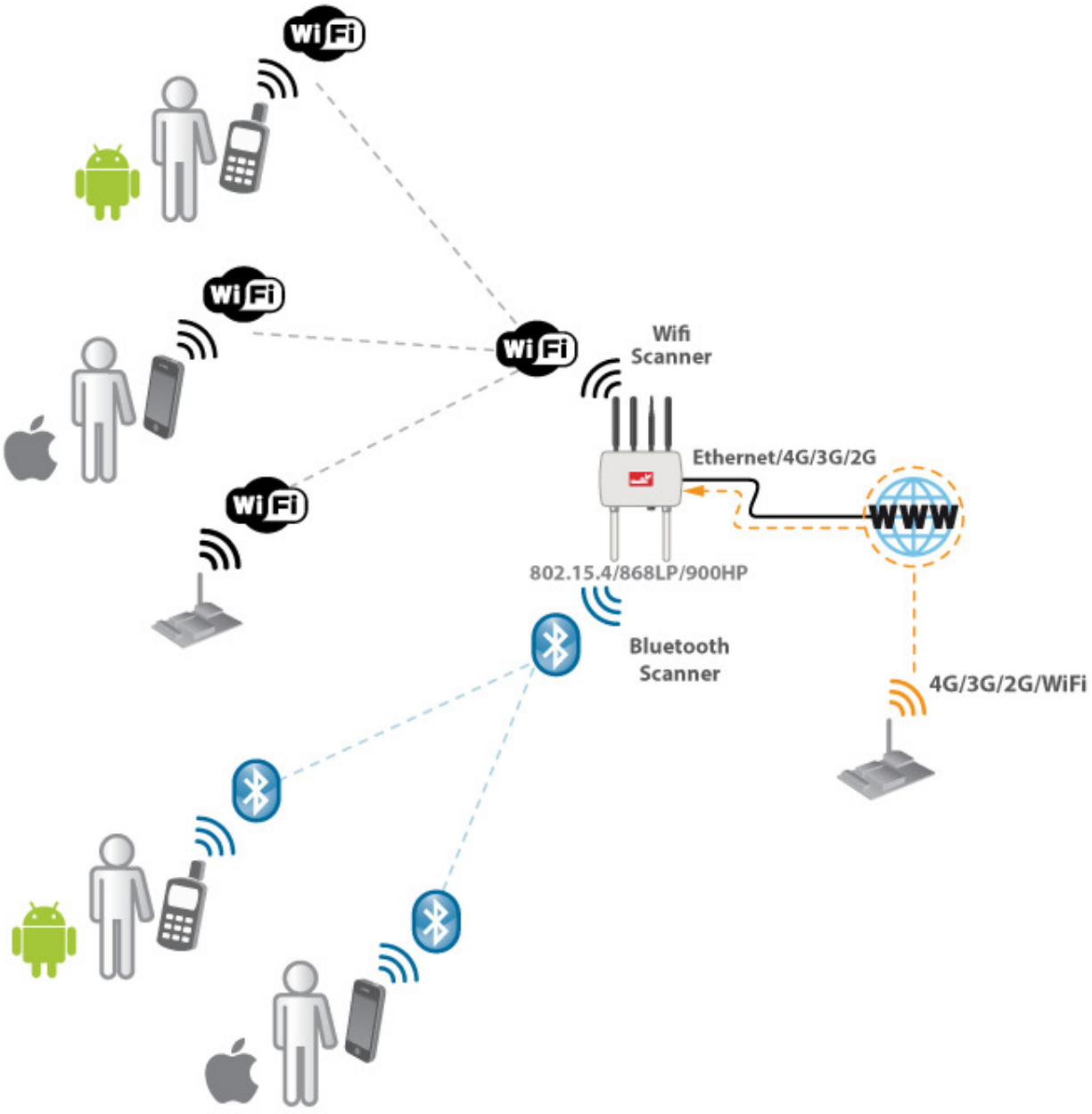


Figure : Meshlium Scanner 4G AP

8. Accessing Meshlium - make it easy!

Meshlium comes with all the radios ready to be used. All the Meshlium units come with the WiFi Access Point ready, so that users can connect using their WiFi devices. Connect the Ethernet cable to your network hub, restart Meshlium and it will automatically get an IP address from your network using DHCP.

Then access Meshlium through the WiFi connection. First of all, search the available access points and connect to "MeshliumXXXX". The four digits at the end allow to identify different Meshliums when working near each other.

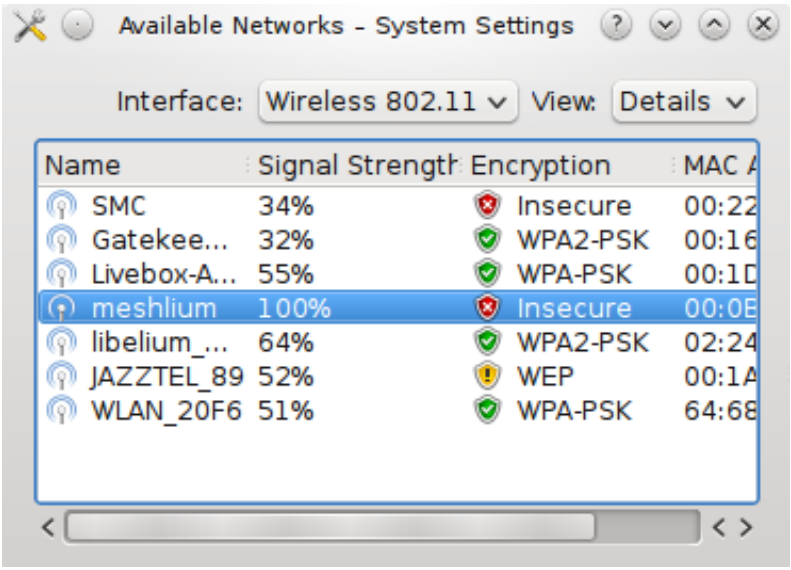


Figure : List of AP with Meshlium network

No password is needed as the network is open (you should change it later in the WiFi AP interface options). When you select it, Meshlium will give an IP address from the range 10.10.10.10 - 10.10.10.250.

Now you can open your browser on your PC and type the URL https://10.10.10.1/ManagerSystem.

Manager System is now securized with HTTPS. Accessing Manager System requires configuring your browser with a certificate. Please read the next section to learn the process.

8.1. Configure your browser to trust the Meshlium Manager System's self-signed certificate

HTTP Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP) for secure communication (protocol is encrypted using TLS or SSL) over a computer network, and is widely used on the Internet. Web browsers know how to trust HTTPS websites based on security certificates that come pre-installed in their software. Those certificates have been created by a Certificate Authority (CA). Certificate authorities are in this way being trusted by web browser creators to provide valid certificates.

Most web browsers alert the user when visiting sites that have certificates issued by authorities not pre-installed. The communication is still secured, but the user needs to install the CA certificate in the browser (Firefox, Chrome, etc) to avoid the warning message. Meshlium devices are provided with a self-signed certificate to serve Manager System, providing protection of the privacy and integrity of the exchanged data while in transit. That protects against man-in-the-middle attacks, eavesdropping and tampering for all the communications between the user and Meshlium.

Therefore, you will have to accept the Meshlium's self-signed certificate in your browser. Follow the instructions below for adding the Meshlium certificate in the most widely used browsers.

8.1.1. Firefox

A window showing the message “Your connection is not secure” will appear.

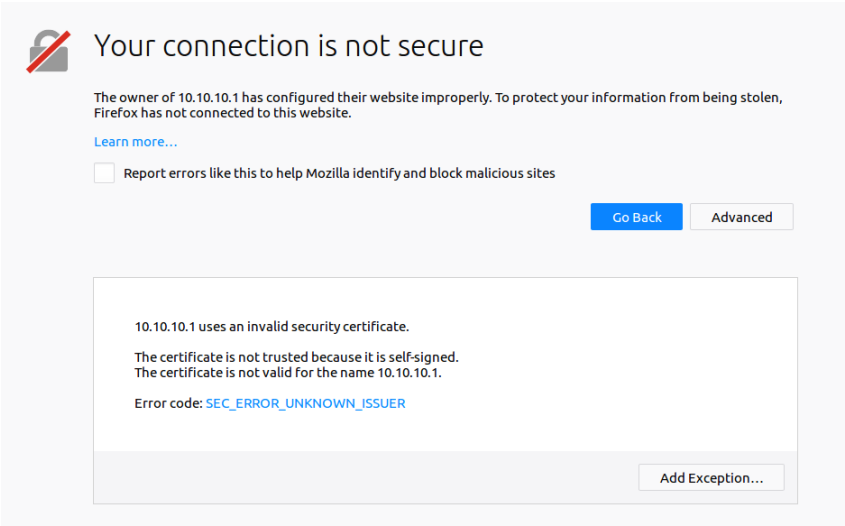


Figure : Manager System’s certificate screenshot on Firefox

Press the button **“Advanced”** to check the certificate details, then press **“Permanently store this exception”** and finally **“Confirm Security Exception”**.

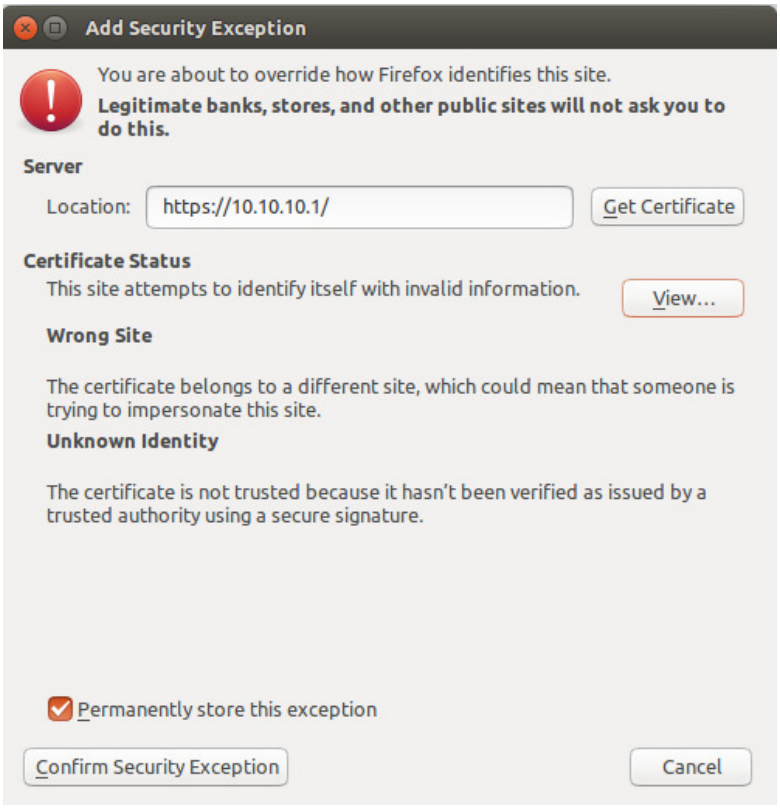


Figure : Accept the self-signed certificate in Firefox

Finally, you will see a lock icon with an exclamation symbol in the URL bar. These symbols mean that the address could not be verified by the certificate (an IP address cannot be validated by a certificate), but your connection is ciphered and cannot be intercepted by an attacker.

8.1.2. Chrome

A window showing the message “Your connection is not private” will appear.

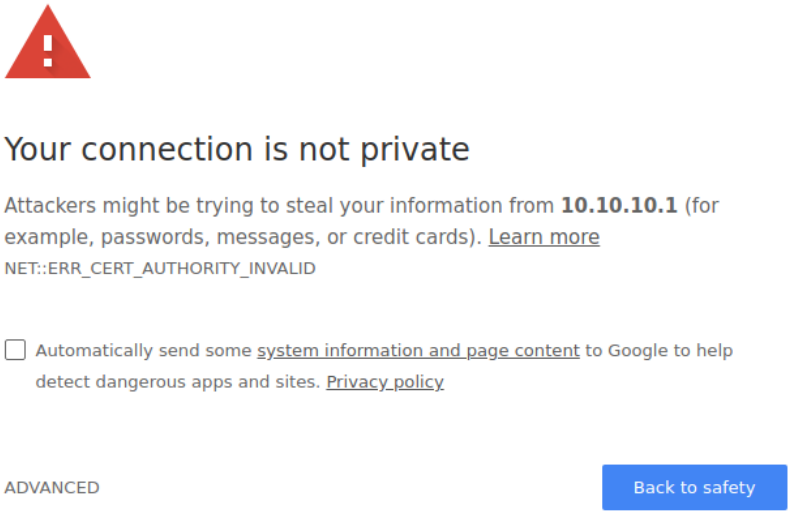


Figure : Manager System’s certificate screenshot on Chrome

For removing this warning, you have to install the Manager System’s certificate as a trusted certificate in your system.
First of all, export the Manager System’s certificate to your computer:

1. Right-click with the mouse in the address bar on the “Not Secure” words.

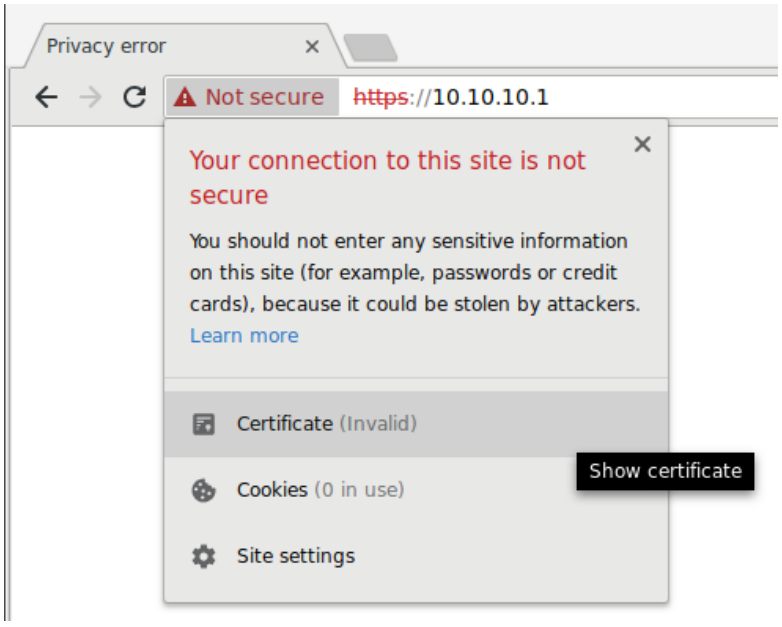


Figure : Show the self-signed certificate in Chrome

2. Click “Certificate” → “Details” → “Export” and save the certificate on your disk.

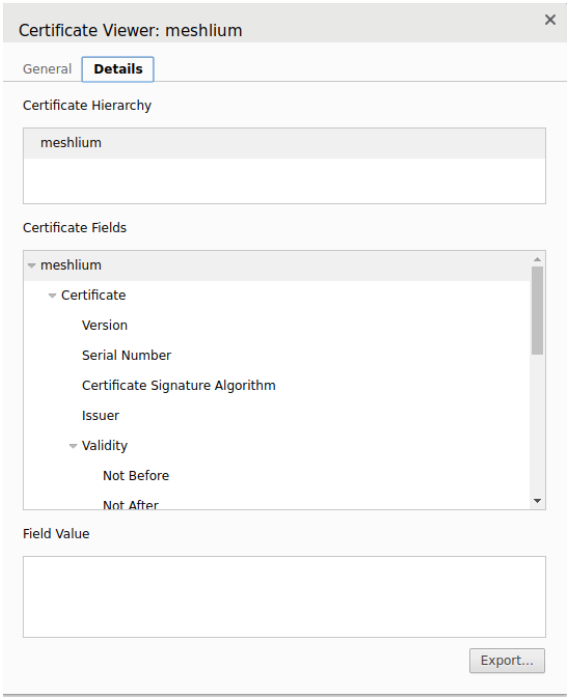


Figure : Export the self-signed certificate from Chrome

Now, install the certificate in your system:

1. On Chrome, go to → “Settings” and search for SSL (chrome://settings/search#ssl).
2. Press “Advanced Configuration”.
3. Click on “Manage certificates”.
4. Go into the “Authorities” tab.
5. Import the certificate previously stored.
6. Restart Chrome.

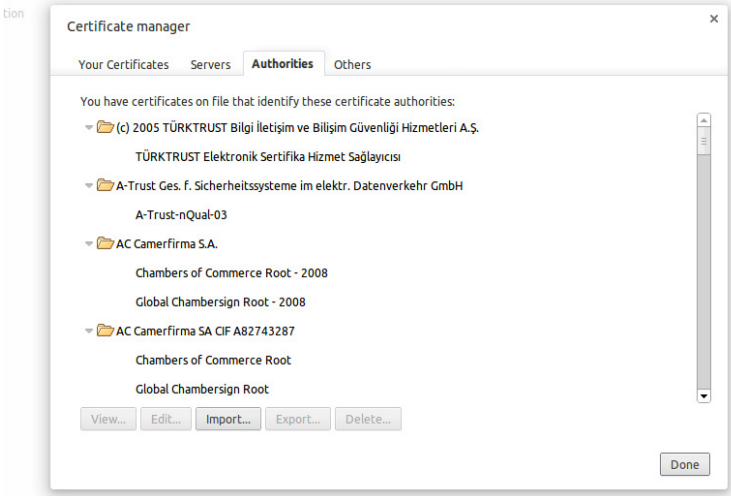


Figure : Add the self-signed certificate in your system

8.1.3. IExplorer (only Microsoft Windows)

A window showing the message “There is a problem with this website’s security certificate” will appear.

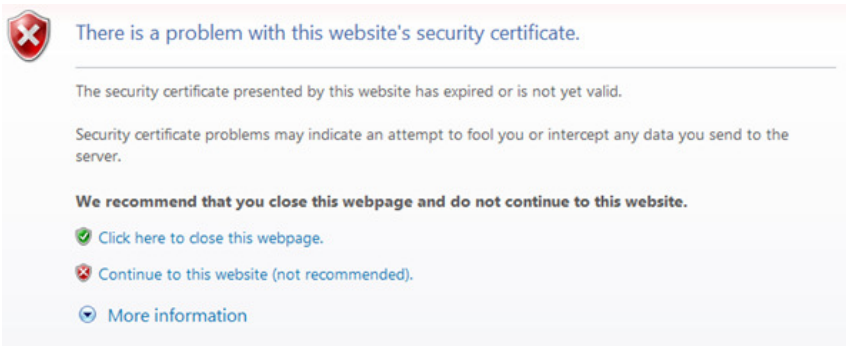


Figure : Manager System’s certificate screenshot on IExplorer

For removing this warning, you have to install the Manager System’s certificate as a trusted certificate in your system:

1. Click on “Continue to this website (not recommended)”.
2. Click on the gear icon → “Internet Options”.
3. The dialog for Internet Options appears. Select the “Security” tab.

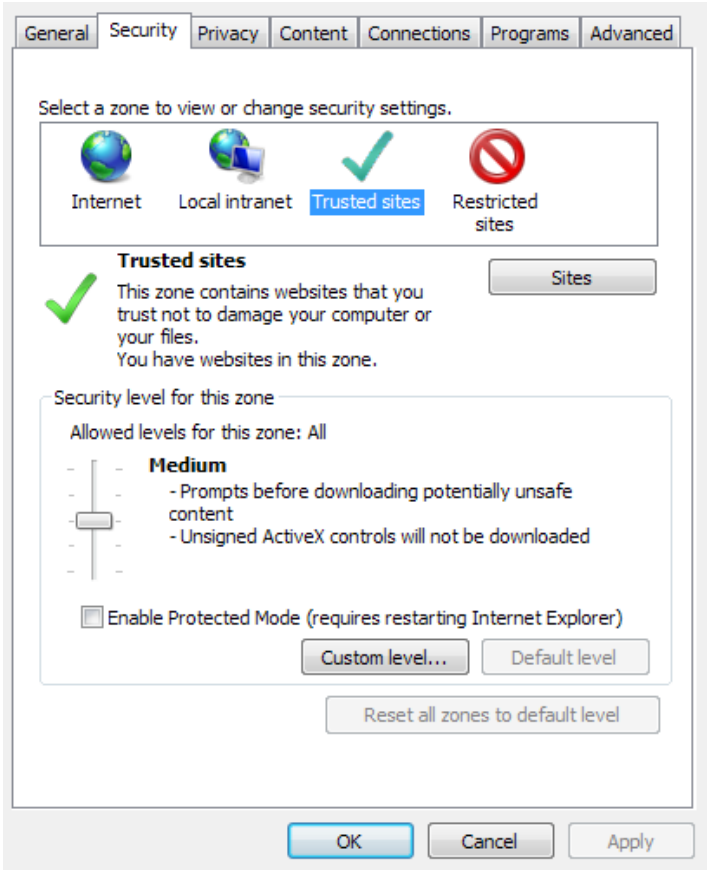


Figure : IExplorer Security tab

4. Select the zone “Trusted sites” and then click on the “Sites” button.

5. A list of trusted sites will appear, with the current URL to add.
6. Click on the “Add button”. The current URL will be added to the list.

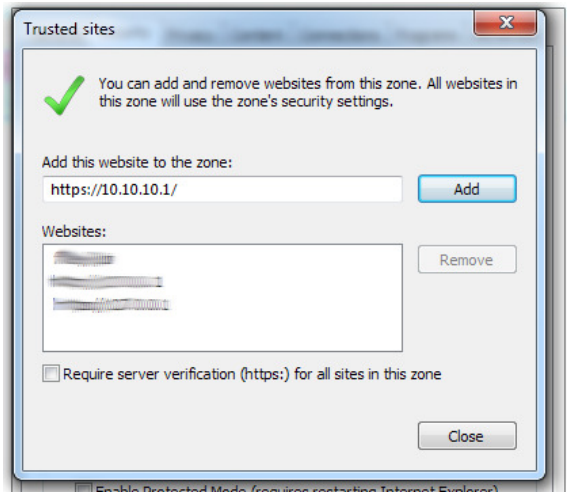


Figure : Add the 10.10.10.1 as trusted site

7. Close the dialogs and go back to the main browser window.
8. Refresh the page (pressing F5 or clicking on the refresh icon).
9. Click on “Continue to this website (not recommended)”.
10. Click on Certificate Error in the red colored address bar and click on “View certificates”.
11. In the Certificate dialog, press “Install Certificate”.
12. In the “Certificate Import Wizard”, click Next
13. On the 2nd step of the wizard, select “Place all certificates in the following store” and click “Browse”.

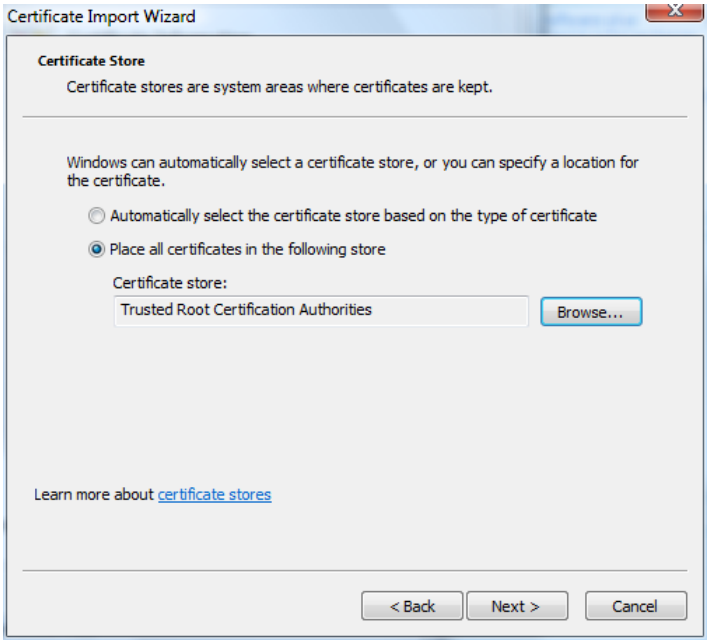


Figure : Add the self-signed certificate in your system

14. In the “Select Certificate Store” dialog, select “Trusted Root Certification Authorities” and click “OK”.
15. In the wizard, click “Next”, then click “Finish”.
16. If a security message pops up, choose “Yes”.
17. Close the dialogs and restart IExplorer.

8.1.4. Safari (only MacOS)

A window showing the message “This Connection Is Not Private” will appear.

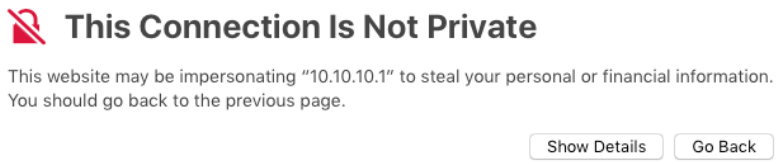


Figure : Manager System’s certificate screenshot on Safari

For removing this warning, you have to install the Manager System’s certificate as a trusted certificate in your system.

1. Click on “Show Details” → “View Certificate”.

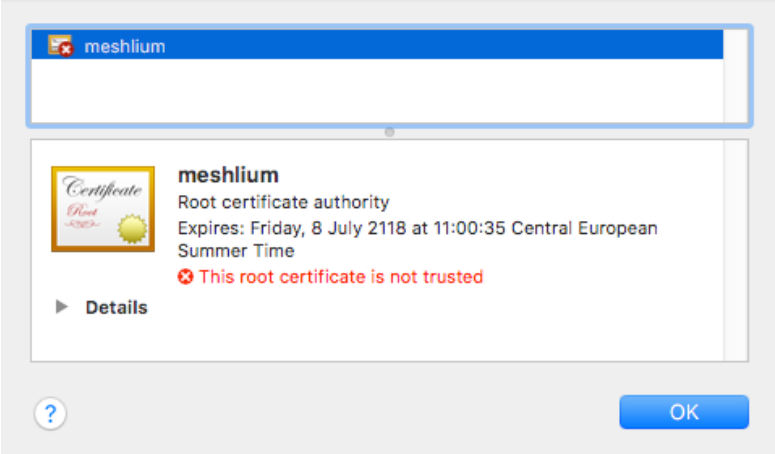


Figure : View the self-signed certificate on Safari

2. Click and drag the image to your desktop. It looks like a little certificate.

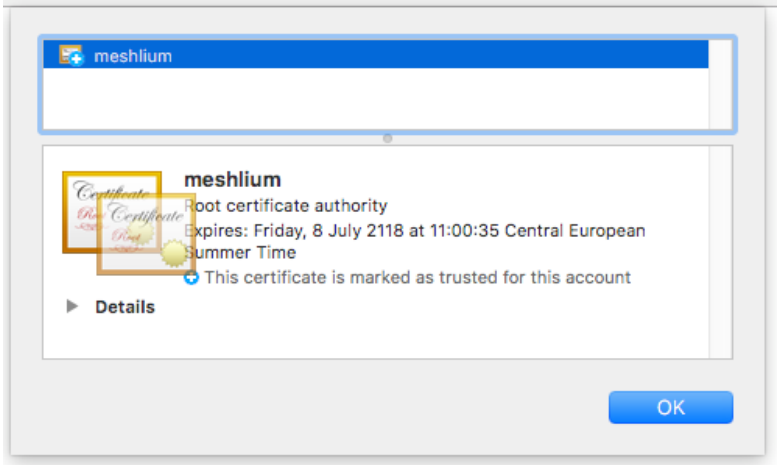


Figure : Export the self-signed certificate to your disk

3. Double-click it. This will start the Keychain Access utility. Enter your password to unlock it.

4. Add the certificate to the “System” keychain (not the login keychain) and press “Add”.

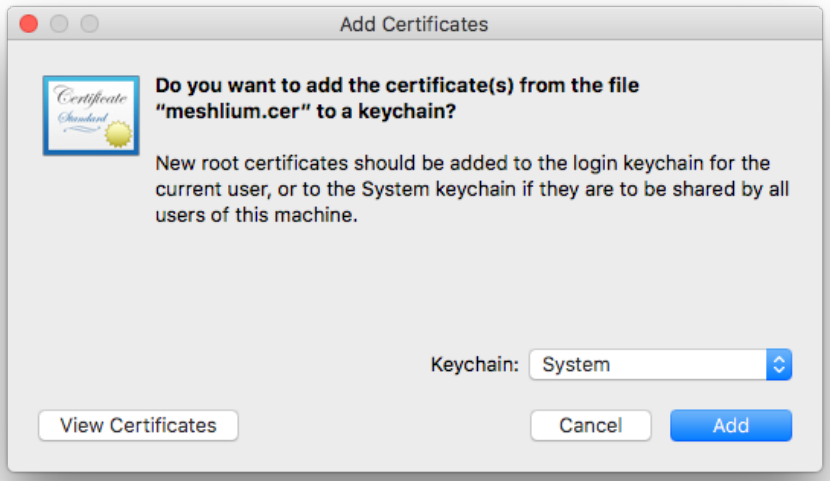


Figure : Select “System”

5. Click "Always Trust" even though this does not seem to have any effect.
6. After it has been added, double-click it. You may have to authenticate again.
7. Expand the "Trust" section.
8. Set "When using this certificate" to "Always Trust".

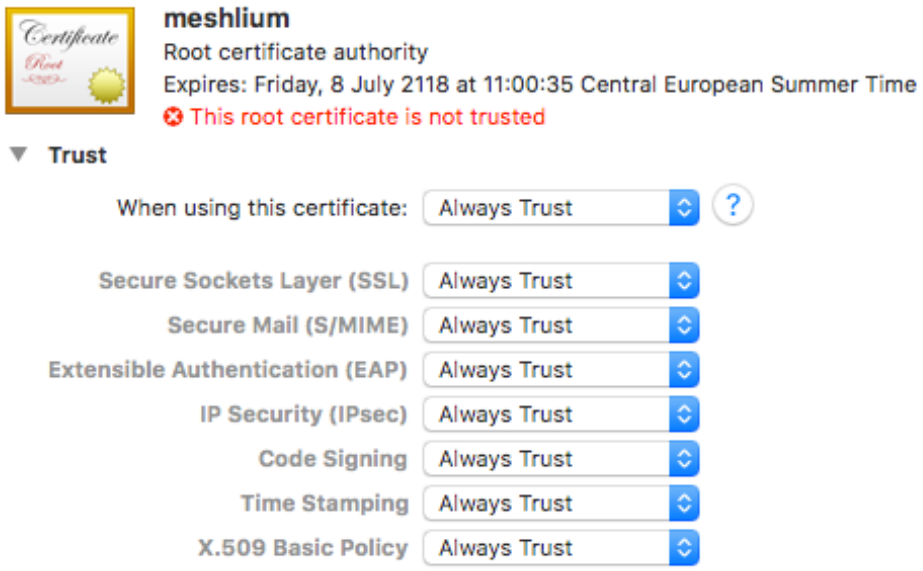


Figure : Select “Always Trust”

9. Close Keychain and restart Safari.

8.2. Access to the Meshlium Manager System

Now you can access the Meshlium Manager System:

- **user**: admin
- **password**: libelium

Figure : Manager System login screen

Figure : Manager System landing page

If your network does not offer DHCP service, Meshlium starts with a default IP address (192.168.1.100). In this case you can connect Meshlium through the WiFi connection (which is always available) or with the crossover cable provided with Meshlium.

If you want to access to the Manager System using the crossover Ethernet cable go to:

- **URL**: https://192.168.1.100/ManagerSystem
- **user**: admin
- **password**: libelium

Important:
*We recommend to **change the default passwords** of the different Meshlium services. Go to "User Manager" section for more information about how to change passwords.*

9. Network interfaces setup

Access the network interfaces setup clicking on the button “Interfaces”:

Figure : Interfaces setup plugin

9.1. Ethernet setup

By default Meshlium comes with the Ethernet interface activated to get dynamically the IP using the DHCP service. In case a static configuration is required the next parameters can be configured:

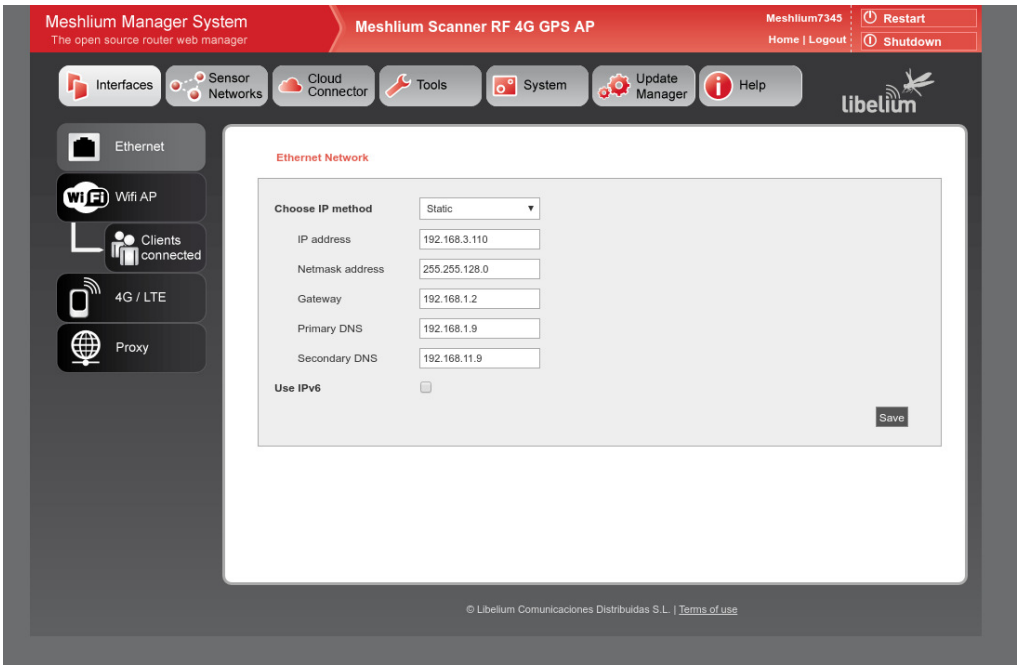


Figure : Ethernet setup

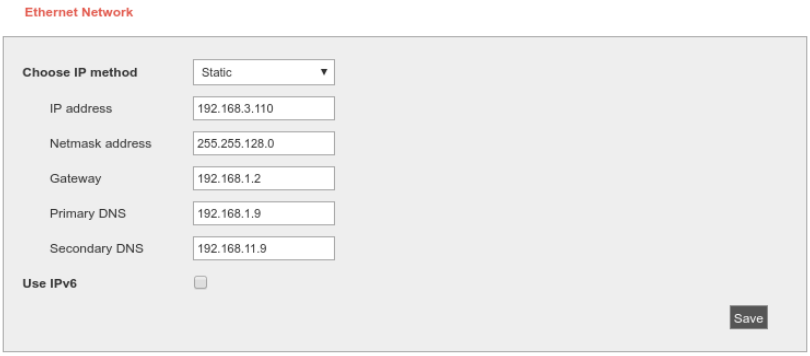


Figure : Ethernet setup form

You can also use IPv6 (Internet Protocol version 6) by setting the check box “Use IPv6”. IPv6 is a version of the Internet Protocol (IP) intended to replace IPv4. The next parameters can be configured:

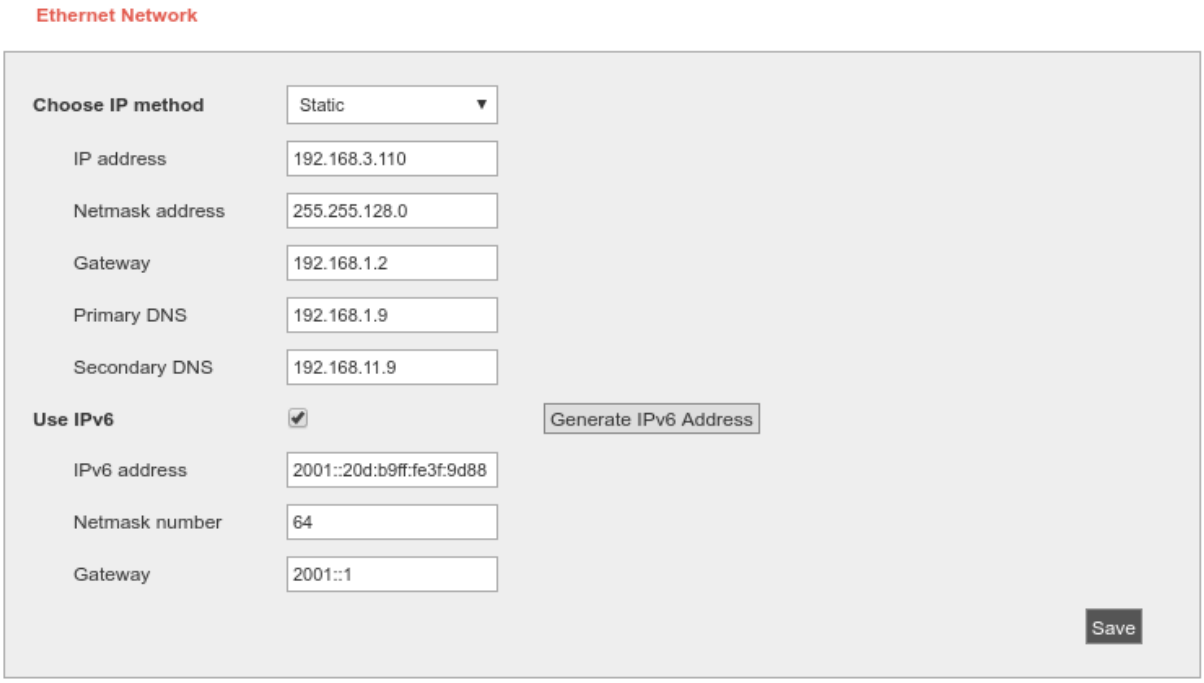


Figure : IPV6 setup

In many cases, IPv6 addresses are composed of two logical parts: a prefix of 64-bit (2001::) and a 64 bit part that is generated automatically from the MAC address of the interface.

The button “Generate IPv6 address” performs this task.

After saving the new options and once you have restarted Meshlium you have to validate the new configuration before the next 5 minutes, if not, the factory default configuration will be restored to avoid leaving Meshlium without connectivity. See section “Network setup confirmation” for more information.

To check IPv6 configuration, after save and restart Meshlium, go to Tools → Ping . Select Ethernet (IPv6), by default ipv6.google.com appears as destination host.

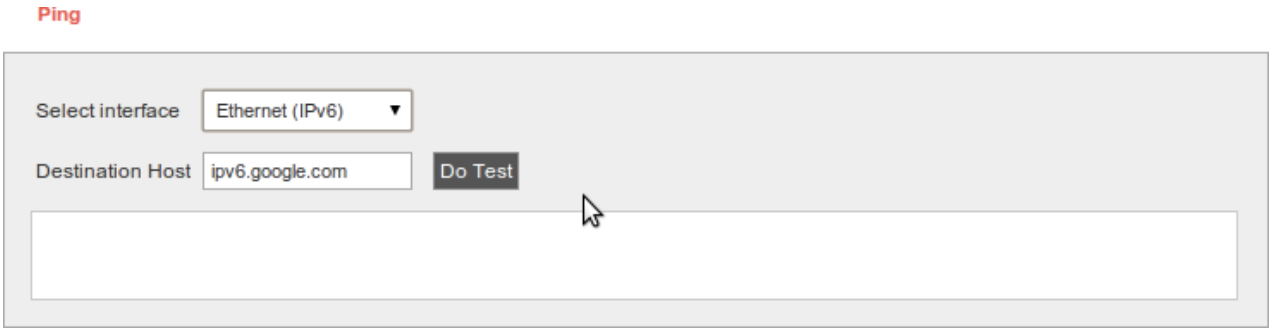


Figure : Ping IPv6 with name

If your Internet Service Provider does not support external IPv6 addresses yet, you can change it to a local address.

Figure : Ping IPv6 with address

Then press “Do Test”. If something like next image appears, you have IPv6 correctly configured.

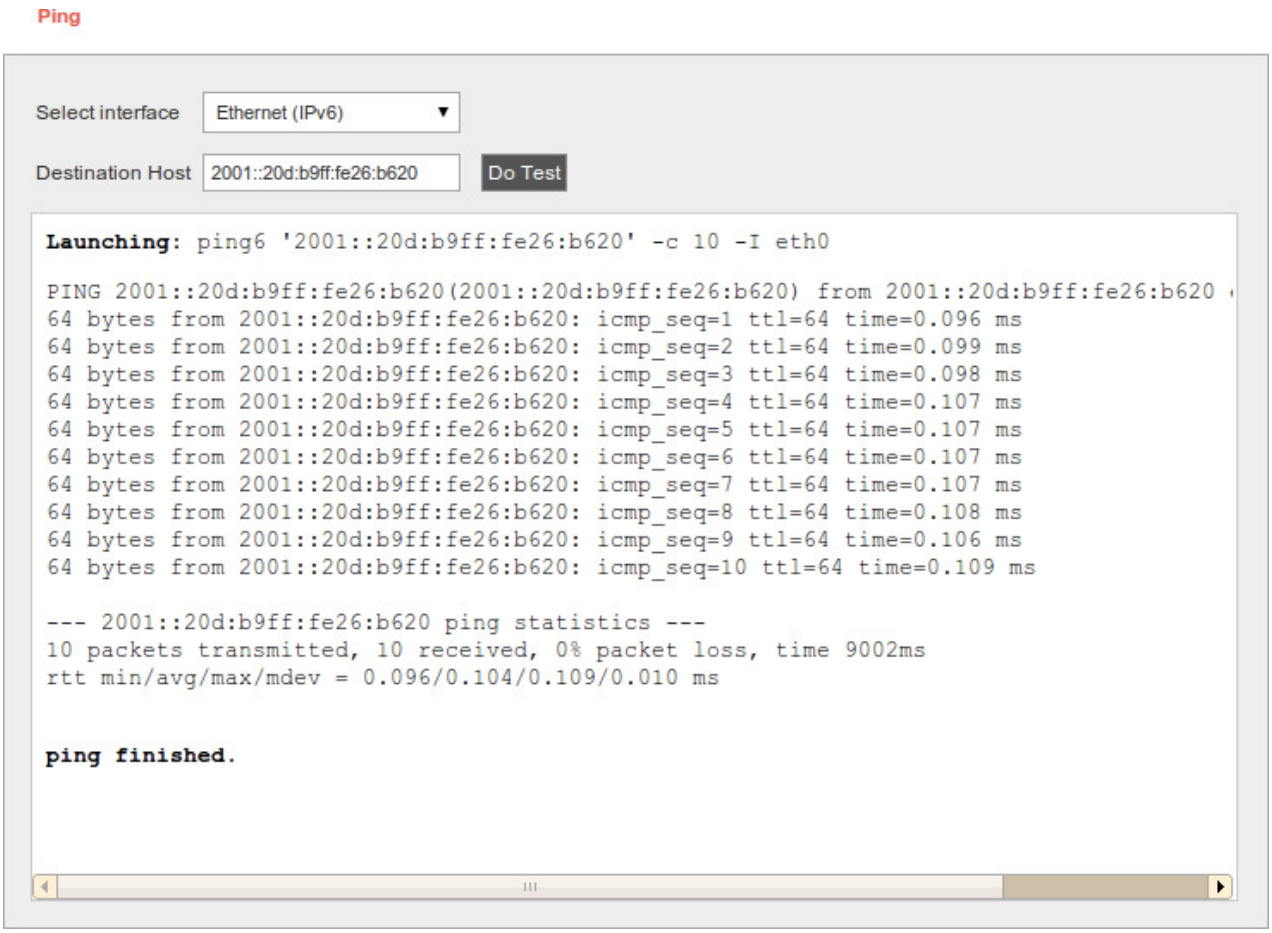


Figure : Ping results

9.2. WiFi Access Point setup

Meshlium is a WiFi Access Point and can supply network connectivity trough WiFi. The most useful feature of the AP is to provide access to Manager System from a tablet or laptop without any physical connection with Meshlium.

By default the AP hast the ESSID “meshliumXXXX” where XXXX are the last four digits of Ethernet MAC. This allows to identify different Meshliums installed nearby.

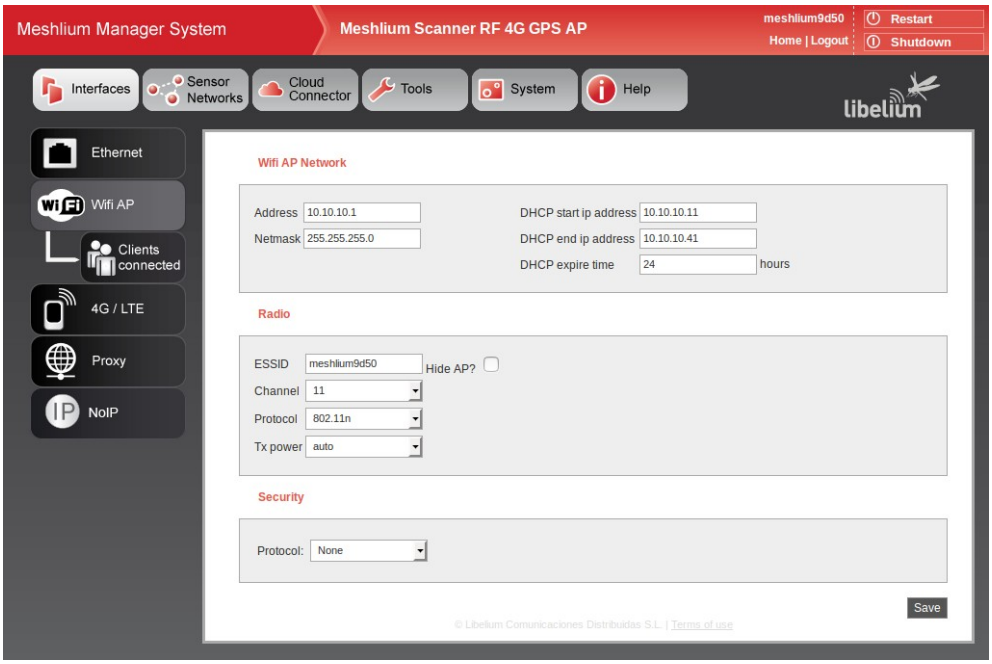


Figure : WiFi Access Point setup

9.2.1. Configuration

There three sections in the configuration page: Network, Radio and Security.

Network

Here you can change the IP of the device in the network and the DHCP setup. Here can be setup:

- IP address of the AP.
- Netmask of the Address.
- DHCP range. The address range in the DHCP setup must be inside the network defined by the IP address and netmask of the AP.
- DHCP lease time.

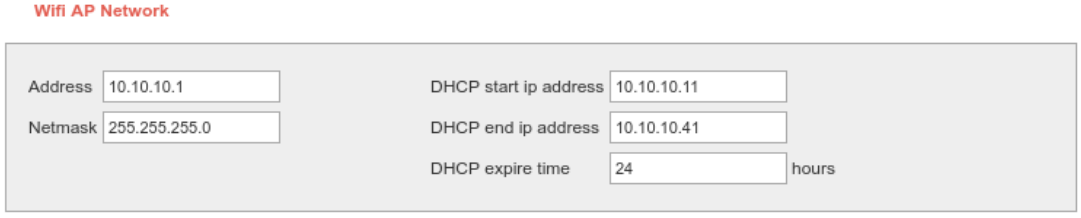


Figure : WiFi AP Network setup

Radio

These are specific WiFi parameters. Here can be setup:

- ESSID of the network. This is the name that appear in the devices that are searching WiFi networks. It can be public or hidden, allowing only connections manually started.
- Channel. It is possible to change the radio channel which is used for transmission, according to the next diagram.

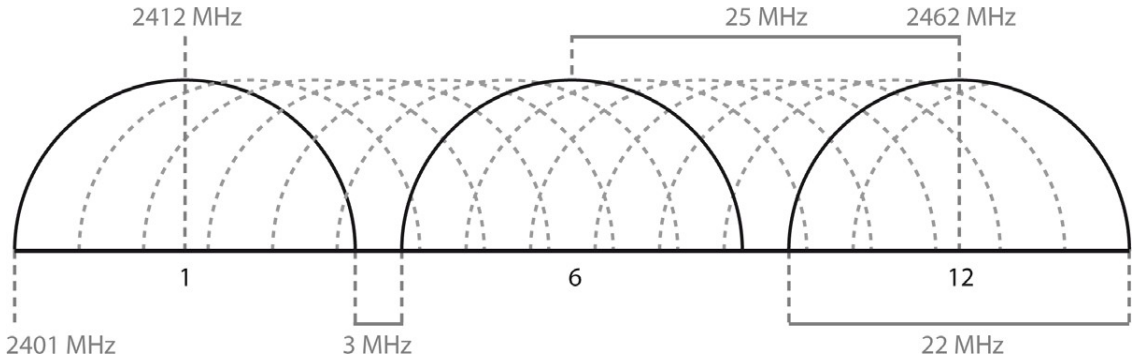


Figure : WiFi radio channels

- Protocol. It is possible to use 802.11g and 802.11n.
- Tx power. It allows to control the transmission power, thus the range of the AP.

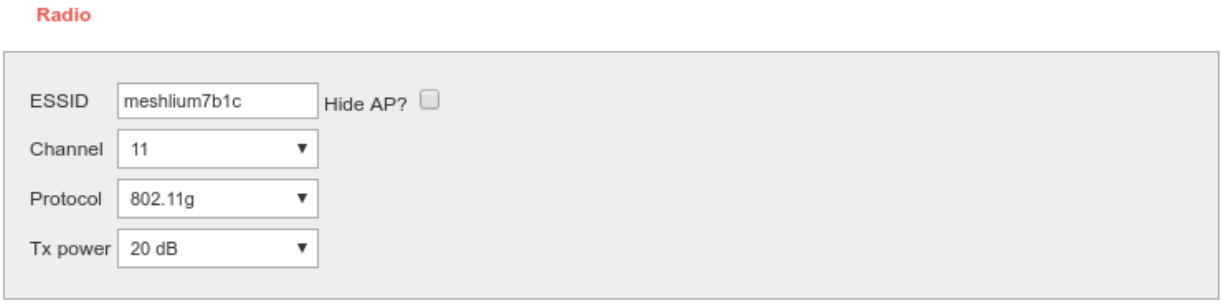


Figure : WiFi radio settings

Security

The WiFi AP can be protected with encryption. WPA and WPA2 are available.

WPA-PSK can be used with a password from 8 to 63 characters.

We recommend use WPA2 in order to get the a good security in the network.

Figure : WiFi AP WPA2

Saving

After saving the setup, a message will warn the user about setup confirmation. A reboot is needed to apply new settings. The setup has to be confirmed within 5 minutes after reboot. More info in **"Network setup confirmation"**.

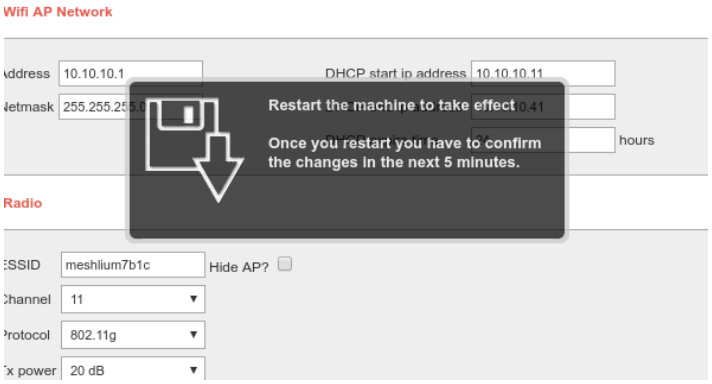


Figure : Confirmation warning

9.2.2. Clients connected

This section shows the list of clients connected to the WiFi AP, showing information like the MAC address and the IP assigned. It is a quick way to know how many devices are connected and who are they.

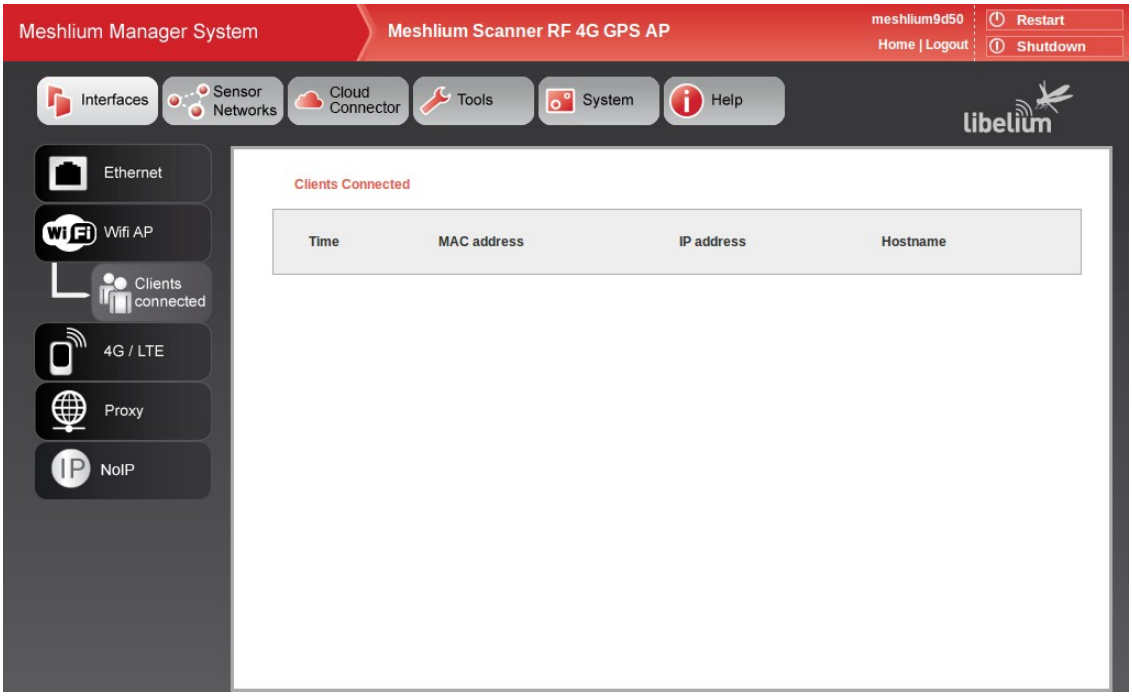


Figure : Clients connected

9.3. Network setup confirmation

After changing Ethernet or WiFi AP setup, a reboot is needed to apply new settings. After this reboot, the user has to confirm the settings in order to definitely apply them. If after 5 minutes of the reboot the user has not confirmed the new settings, last validated settings will be applied again. If there are no validated settings, default settings will be applied.

In the confirmation screen the user can select to confirm new settings, change to last validated settings or change to default settings. All the information of every setup will be shown. After the confirmation is done, the new settings will be stored as last validated settings for future confirmations.

The system will show a confirmation window for every setting changed, one for Ethernet setup and one for WiFi AP setup, so it can be independently confirmed.

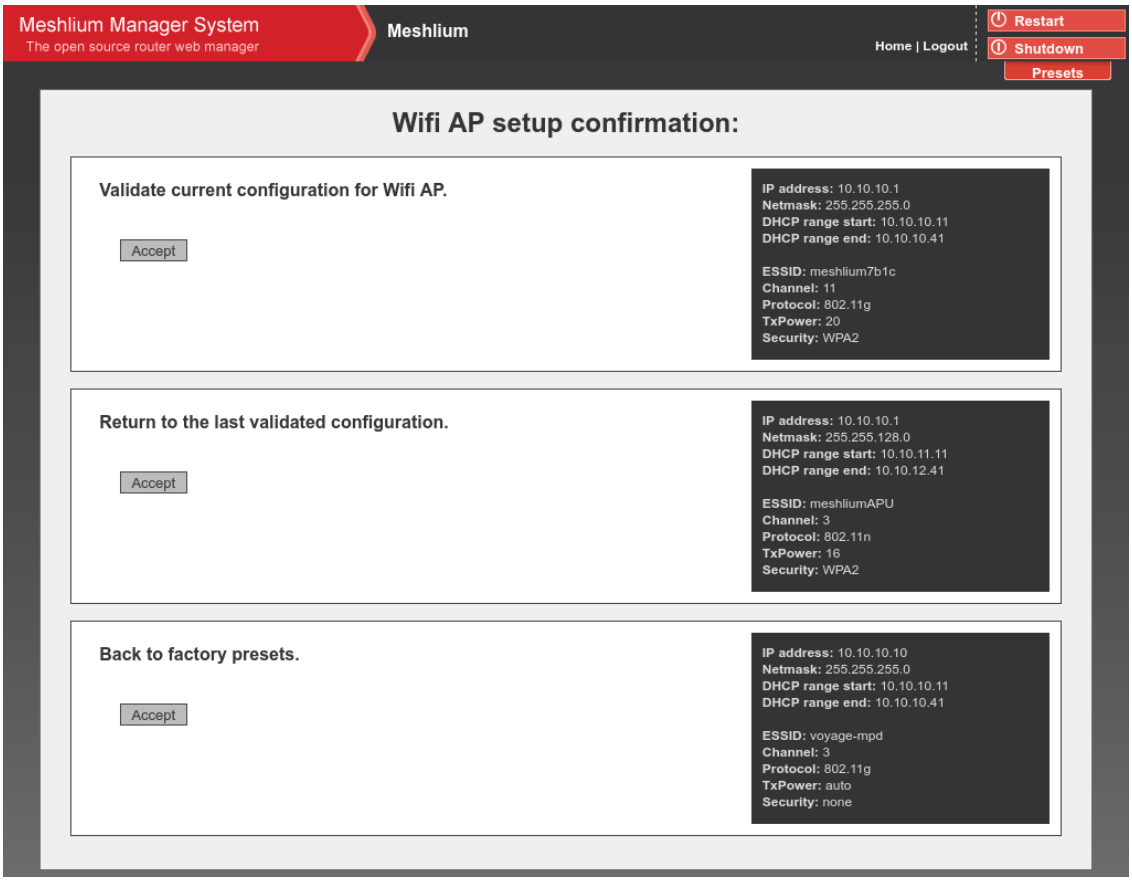


Figure : Confirmation screen

9.4. 4G Setup

This plugin allows to set-up the parameters of the modem connection. There is a list with some initial configurations depending on the country and the operator. However, this list may not be updated with the last valid configuration of your mobile provider. Ask your mobile company for the information required to connect (APN, Username, Password) and add the PIN code of the SIM card used (leave empty if there is no PIN).

We recommend to disable the PIN in the SIM card as this will make easier the test and validation process and will avoid to block the SIM card.

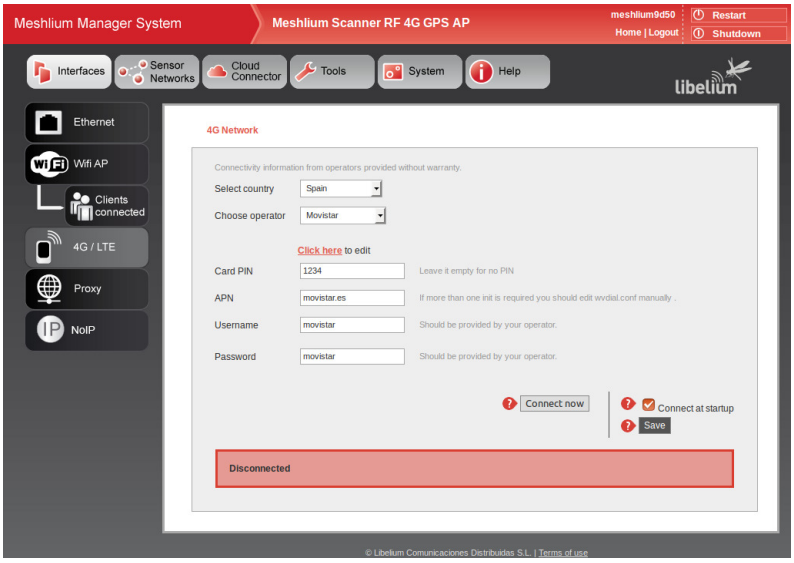


Figure : 4G setup plugin

After setting the 4G parameters, you have to save them by pressing the “Save” button. Then you can test your connection through the “Connect now” button. It will try to connect to your carrier and get a valid IP. Once the connection has been made the default gateway of the machine is changed so all the clients connected through WiFi will reach the Internet via 4G.

Important: Once you get a valid 4G IP through the “Connect now” button, you will not be able to access Meshlium via Ethernet unless you are connected through the same Local Area Network. For this reason we recommend to make all the tests using the WiFi connection.

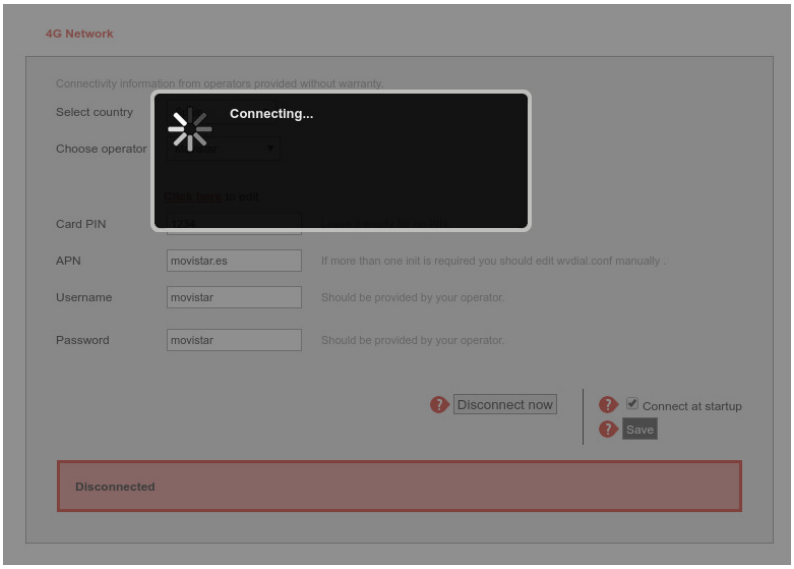


Figure : 4G connecting

If connection is established, the IP will be shown in the interface. Once the modem is connected, a process will check the connection every 15 minutes and will try to reconnect in case of disconnection.

Figure : 4G successfully connected

If you want the 4G to be the Default Gateway of Meshlium each time it starts just activate the service in the "Set as Default Gateway" check box before saving. Setting this on will connect to the Internet using the 4G radio each time Meshlium restarts.

If any problem is preventing the device to connect at boot or to reconnect after a connection fall, a message will be displayed in the plugin. The user can manually stop automatic reconnection by pressing the "Disconnect now" button.

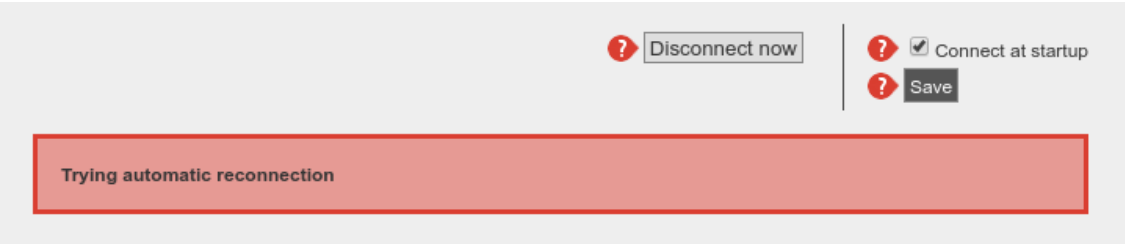


Figure : 4G trying to reconnect

9.5. Proxy setup

This plugin allows to setup an HTTP proxy for some features of Meshlium. Here can be configured the proxy address, the port and the credentials (leave blank if not authentication needed).

Figure : Proxy setup plugin

The proxy can be enabled or disabled from the control of the interface.

Figure : Proxy enable control

Figure : Proxy disable control

***Note**: Currently the proxy feature is only available for visualizer plugin. This feature will be gradually included in other services.*

9.6. No-IP setup

This plugin allows to setup a No-IP account (https://www.noip.com) for dynamic IP remote access.

Configure the following parameters with the information of a valid and active No-IP account, previously created in the No-IP platform:

- **Hostname**: Name of the host to link with the Meshlium IP.
- **Username**: No-IP account username.
- **Password**: No-IP account password.
- **Interval**: Update interval in minutes.

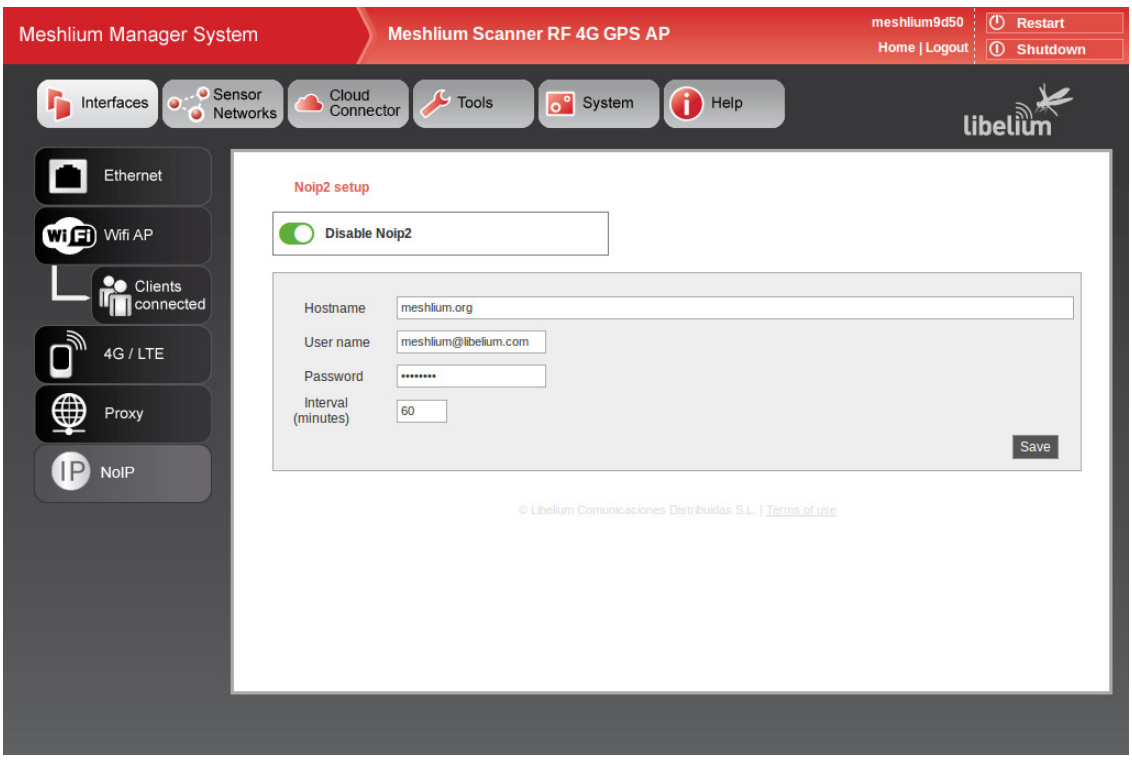


Figure : NoIP setup plugin

Please, refer to the interface configuration section to use a proxy.

Figure : NoIP enable control

Figure : NoIP disable control

10. Wireless Sensor Networks

10.1. Meshlium and Waspmote

One of the main applications of Meshlium is being a gateway for Wireless Sensor Networks based on Waspmote and Plug & Sense! devices. These are sensor nodes that can work with different communication technologies like WiFi, 4G or XBee among others. More than 70 sensors are already available and a complete open source IDE (API libraries + compiler) make really easy to start working with the platform.

More info at:

https://www.libelium.com/products/waspmote/

https://www.libelium.com/products/plug-sense/

In the main page of Sensor Networks tab will be shown the devices in the system showing the last received data.

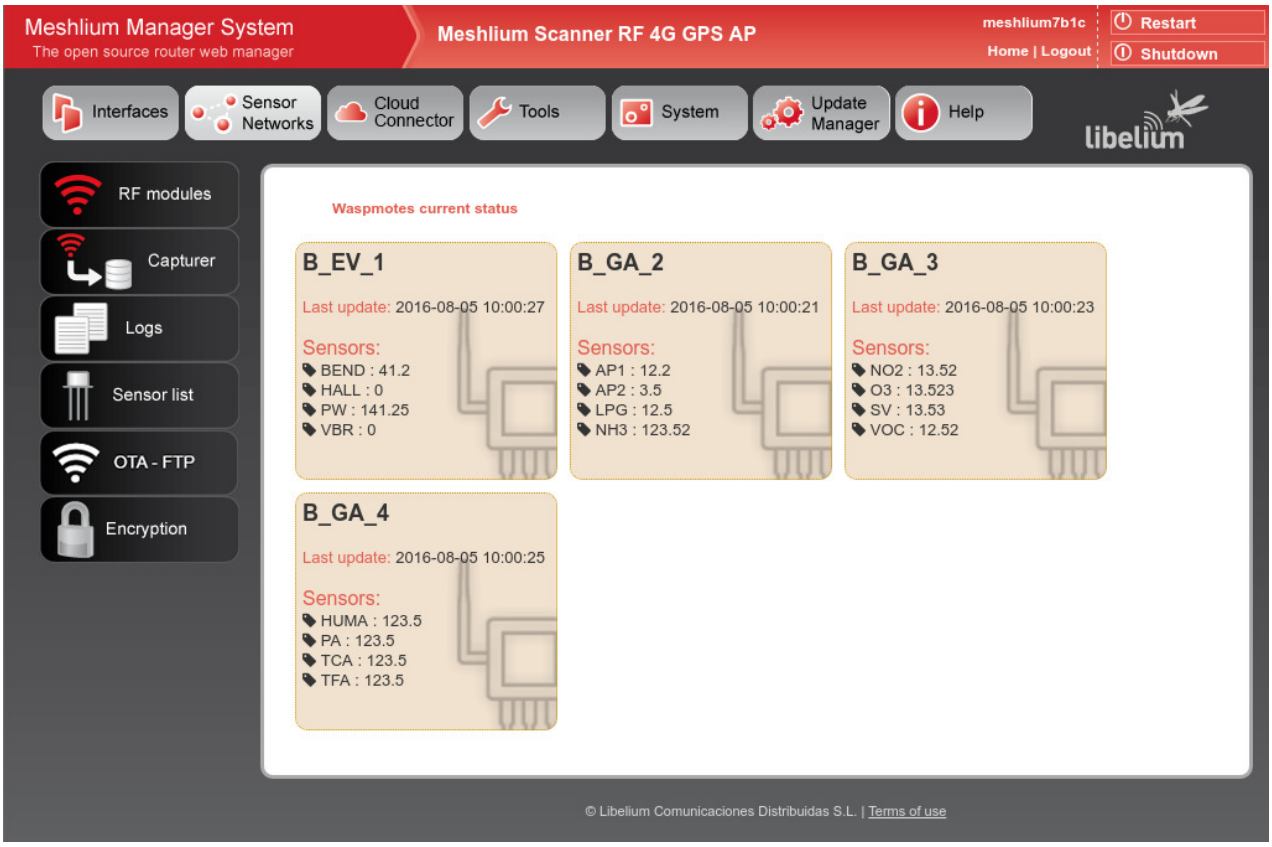


Figure : Nodes with last data

10.2. Receiving and storing data

10.2.1. Receiving trough RF communications

RF module setup

Meshlium can integrate 3 different RF modules: XBee-PRO 802.15.4 (2.4 GHz), XBee 868LP (868 MHz) and XBee-PRO 900HP (900 MHz). It can have up to 2 RF modules at the same time.

RF modules setup can be found in:

Sensor Networks → RF modules

The plugin will show:

- one tab for each module detected in the device.
- switch button to enable/disable the sensor parser service.
- menu for changing logging levels in HTTP and RF parser.

Figure : RF communications

XBee-PRO 802.15.4 radio setup

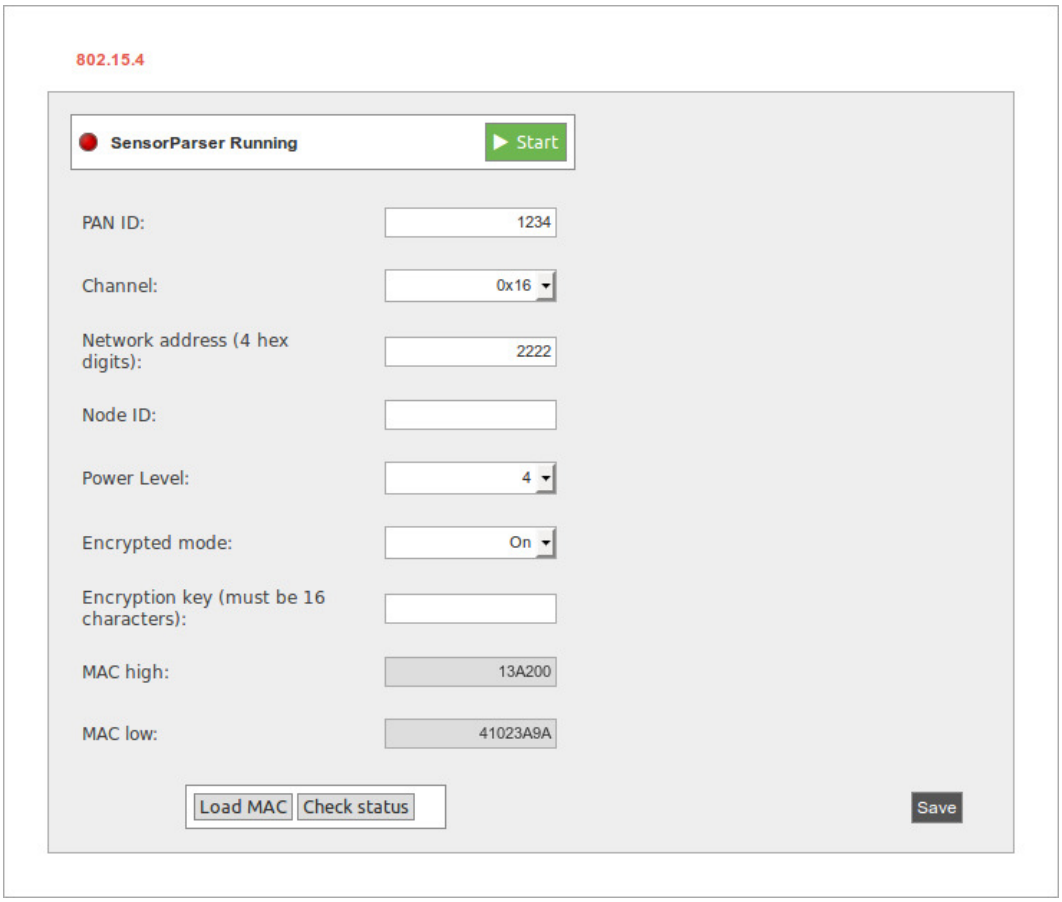


Figure : XBee-PRO 802.15.4 setup

In this module the parameters to setup are:

- **PAN ID**: Personal Area Network ID (also known as Network ID). It is the identifier of the network. It has to be the same in all the nodes in order to be able to send data to this Meshlium.
- **Channel**: Frequency channel used for transmissions.
- **Network Address**: User defined identifier for the node in the network. 4 hexadecimal digits (MY).
- **Node ID**: readable name set for the device, by default "Meshlium". Up to 20 characters.
- **Power level**: [0-4] By default 4.
- **Encrypted mode**: Internal XBee AES 128 bits encryption. Disabled by default.
- **Encryption key**: 16 characters.
- **MAC**: 64 bits hardware address of the module. It is a read-only value divided in two parts:
  - MAC-high: 32 bits (8 hexadecimal digits).
  - MAC-low: 32 bits (8 hexadecimal digits).

This setup must be consistent with those set on the Waspmote and Plug and Sense nodes.

In the bottom part of the interface, the button "**Check status**" allows to check if the module setup is concordant with values shown in the interface. The button "**Save**" will write the parameters in the module.

Both process ("Save" and "Check status") require the sensorParser daemon to be stopped. This means no frames will be received while executing this actions. Be patient this can take up to 1 minute to finish.

XBee 868LP radio setup

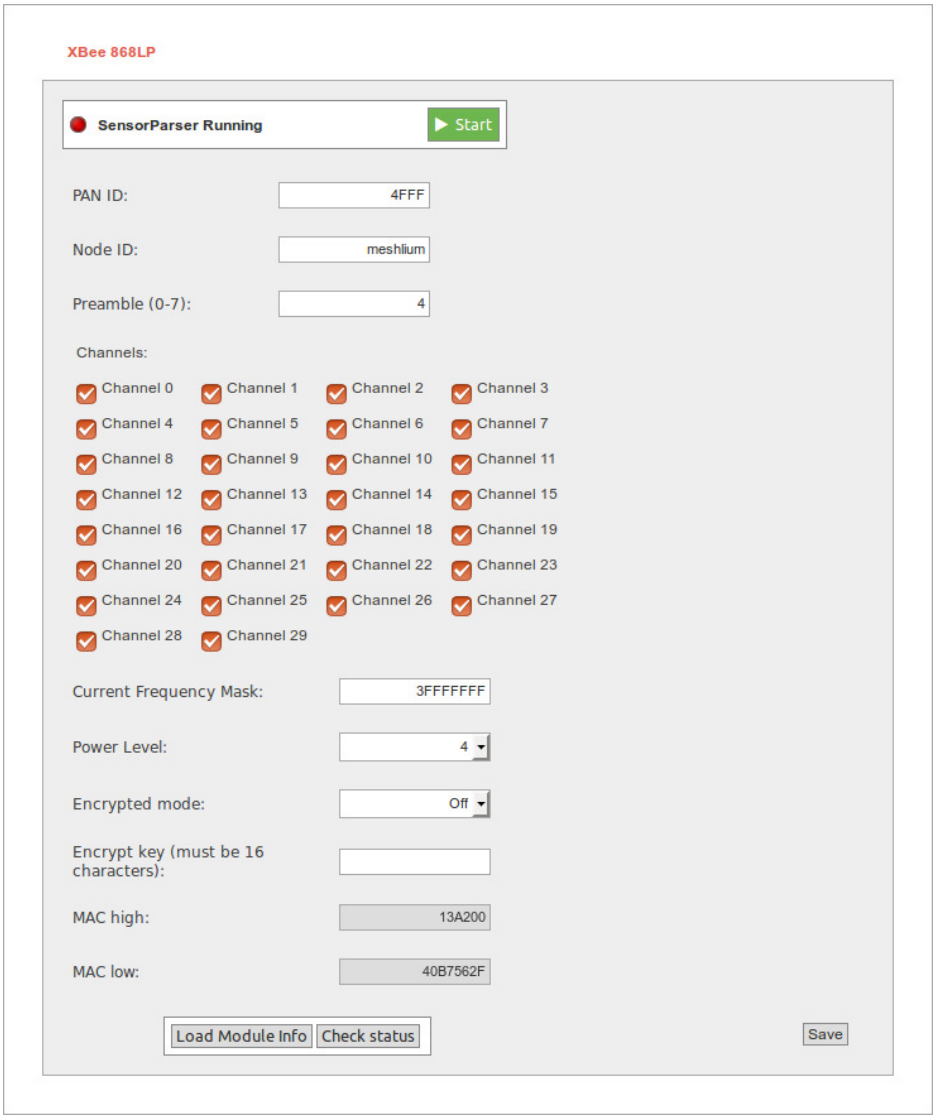


Figure : XBee 868LP setup

In this module the parameters to setup are:

- **PAN ID**: Personal Area Network ID (also known as Network ID). It is the identifier of the network. It has to be the same in all the nodes in order to be able to send data to this Meshlium.
- **Node ID**: readable name set for the device, by default "Meshlium". Up to 20 characters.
- **Preamble**: An extension to PAN ID. It needs to be the same in the nodes too.
- **Channel**: This module allow to select the channels that can be used. The module automatically selects the channel for the communication between available ones. Once the channels are selected, the plugin generates the "**Channel Frequency Mask**" (read-only 8 hex digits) that the needs to be set in the nodes.
- **Power level**: [0-4] By default 4.
- **Encrypted mode**: Internal XBee AES 128 bits encryption. Disabled by default.
- **Encryption key**: 16 characters.
- **MAC**: 64 bits hardware address of the module. It is a read-only value divided in two parts:
  - MAC-high: 32 bits (8 hexadecimal digits).
  - MAC-low: 32 bits (8 hexadecimal digits).

XBee-PRO 900HP radio setup

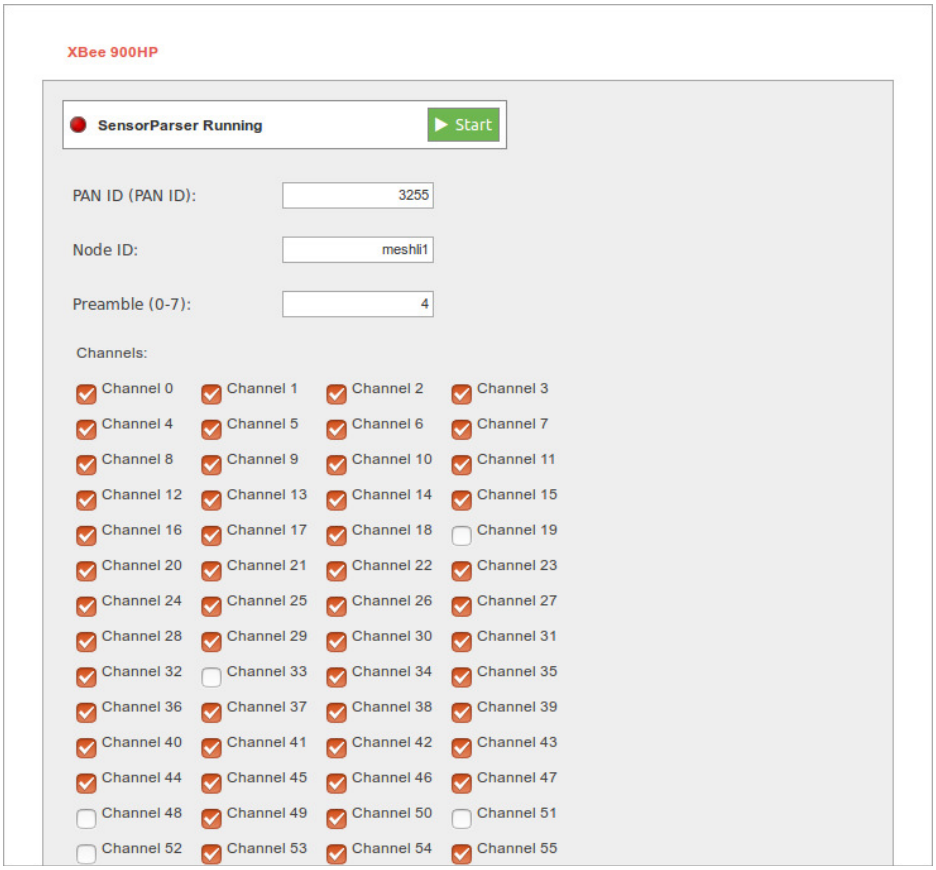


Figure : XBee-PRO 900HP setup

In this module the parameters to setup are:

- **PAN ID**: Personal Area Network ID (also known as Network ID). It is the identifier of the network. It has to be the same in all the nodes in order to be able to send data to this Meshlium.
- **Node ID**: readable name set for the device, by default “Meshlium”. Up to 20 characters.
- **Preamble**: An extension to PAN ID. It needs to be the same in the nodes too.
- **Channel**: This module allow to select the channels that can be used. The module automatically selects the channel for the communication between available ones. Once the channels are selected, the plugin generates the “**Channel Frequency Mask**” (read-only 16 hex digits) that the needs to be set in the nodes. In the bottom part of the interface is shown the minimum number of channels that have to be selected.
- **Power level**: [0-4] By default 4.
- **Encrypted mode**: Internal XBee AES 128 bits encryption. Disabled by default.
- **Encryption key**: 16 characters.
- **MAC**: 64 bits hardware address of the module. It is a read-only value divided in two parts:
  - MAC-high: 32 bits (8 hexadecimal digits).
  - MAC-low: 32 bits (8 hexadecimal digits).

Encryption setup

Link layer key management (AES-128)

This feature is provided by XBee modules.

Encryption is this layer provided through the AES algorithm. Specifically through the type AES-CTR. In this case the Frame Counter field has a unique ID and encrypts all the information contained in the Payload field which is the place in the link layer frame where the data to be sent is stored. The way in which the libraries have been developed for module programming means that encryption activation is as simple as running the initialization function and giving it a key to use in the encryption.

```
{
  xbee.encryptionMode(1);
  xbee.setLinkKey(key);
}
```

In the Manager System, on Sensor Network section, users can encrypt messages on link layer. It can be achieved by setting the parameters:

- **Encrypted mode**: true/false (by default false).
- **Encryption Key**: Must be 16/24/32 characters depending on the AES encryption type (128/192/256 bits).

See section “RF module setup” for more details about setting encryption.

Application layer key management

Meshlium is capable to properly receive encrypted data from Waspmote. The coding process is made in the application layer, so it is Waspmote and Meshlium processor and not XBee module who encrypts and decrypts the messages.

The user have to set a key for the encryption in Waspmote and Meshlium.

In the Manager System, the menu for managing the encryption options is found in: Sensor Networks → Encryption

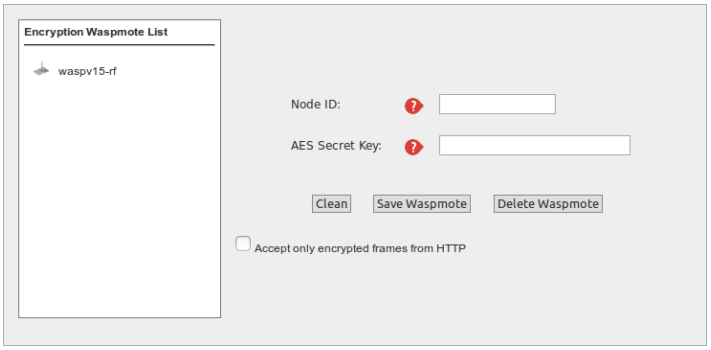


Figure : Encryption key setup

For each Waspmote unit which is able to send frames to Meshlium, Waspmote keys can be added to an encryption Key file. In this interface the user must specify the node ID and the Waspmote AES secret key (128, 192 or 256 bits).

After defining the above fields, press the button “Add Waspmote”. A new entry will be generated in the left side list.

For deleting a specific Waspmote unit from the list, select the Waspmote unit and press “Delete Waspmote”. The encrypted frames received from this node will not be able to be decrypted anymore.

The option “Accept only encrypted frames from HTTP” is also available. Select this option if you want to discard all the *not encrypted* frames received by the HTTP parser.

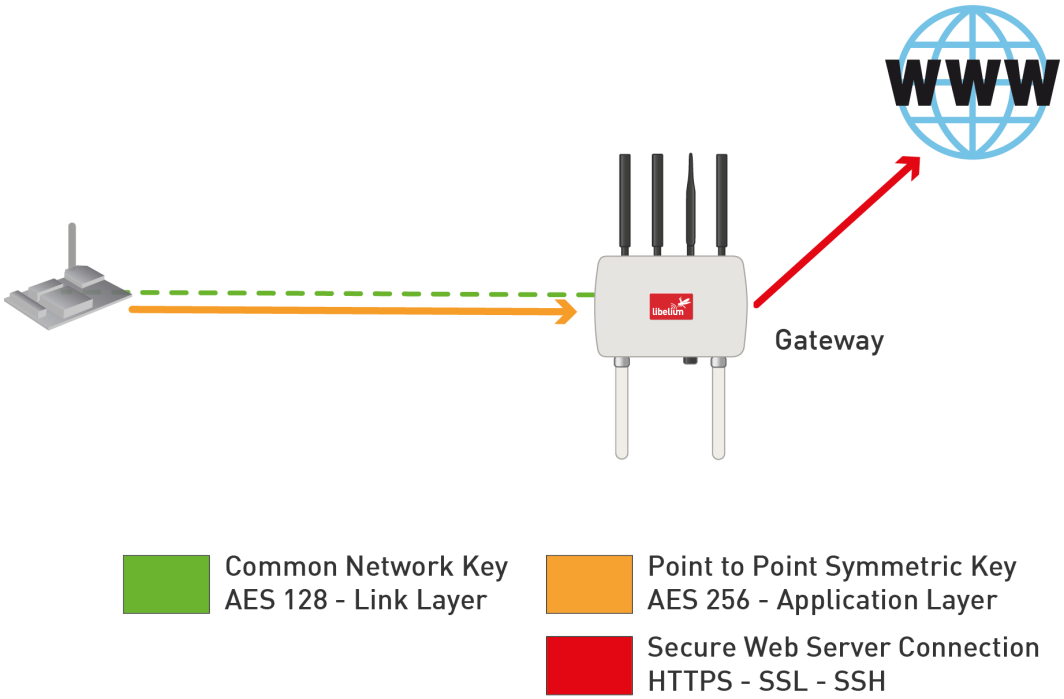


Figure : Encryption in communications

Once the user has properly set the AES keys associated to every Waspmote unit, receiving AES encrypted frames in Meshlium is a straightforward process.

As an encrypted frame arrives to Meshlium, sensorParser program takes the appropriate key for the Waspmote ID. The frame is decoded with the key and the information is extracted to the sensor database.

Bear in mind that to use this feature, the frame have to be created with the Waspmote libraries for AES frames. You can see further information about this in the Waspmote guides.

https://www.libelium.com/waspmote

Capturing and storing sensor data from RF module

Meshlium will receive the sensor data sent by Waspmote and Plug and Sense using the RF radio and it will store the frames in the local database. Data is stored with the timestamp of the reception in the Meshlium unit. The timestamp is always stored in UTC to avoid inconsistencies (regardless of the time zone selected in Meshlium).

That can be done in an automatic way thanks to the Sensor Parser.

The Sensor Parser is a software system which is able to do the following tasks in an easy and transparent way:

- receive frames from XBee modules (with the Data Frame format).
- parse these frames.
- store the data in the local database.

Besides, the user can add his own sensors, and the data will be parsed in the database too. In order to add your own sensor frames properly go to the section “Sensor list”.

We can perform two different storage options with the frames captured:

- Local database.
- External database.

All the data is stored in the local database in the first place, then it can be synchronized to an external database as per user needs.

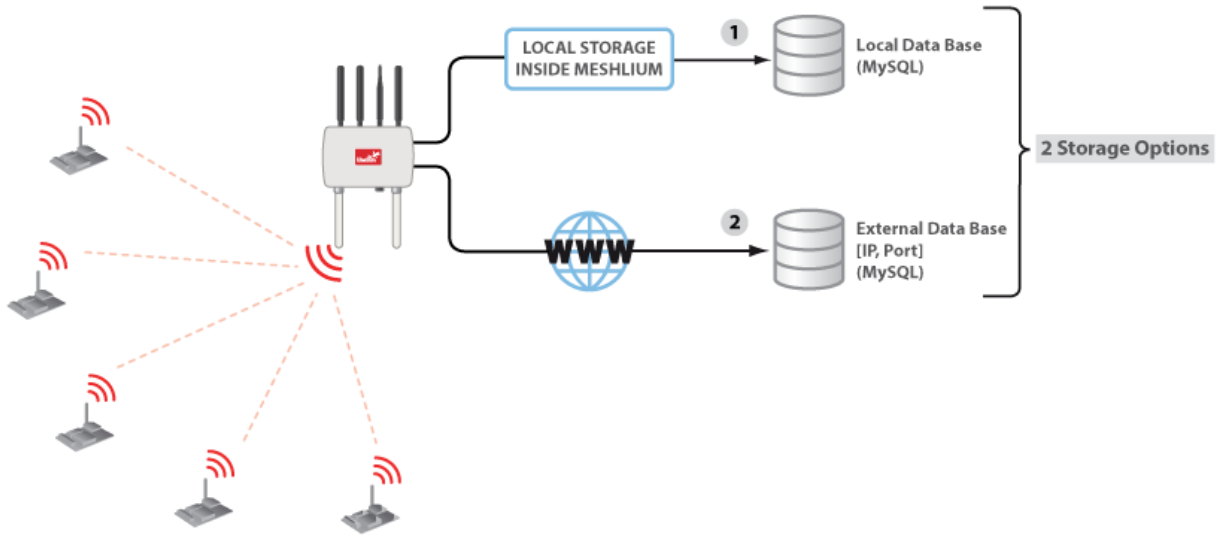


Figure : Storage options

The data stored can be synchronized too to external services using the Internet connection.

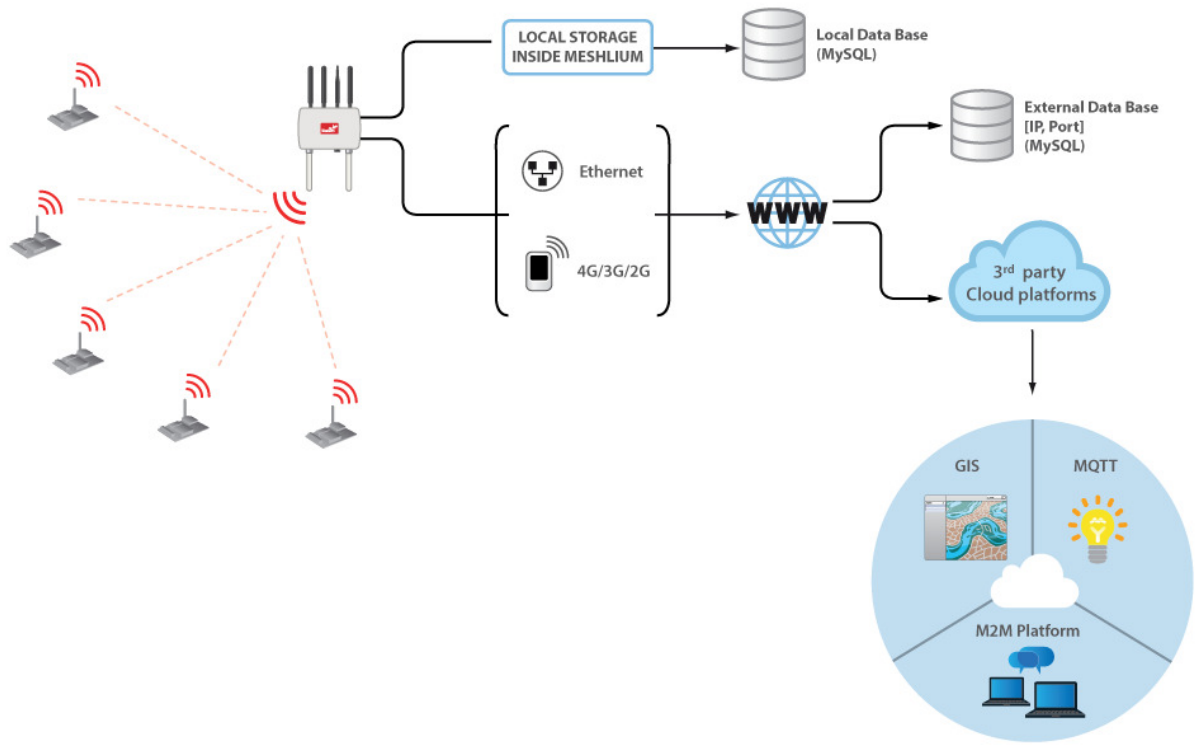


Figure : External synchronization options

10.2.2. Receiving trough 4G / WiFi / Ethernet (HTTP)

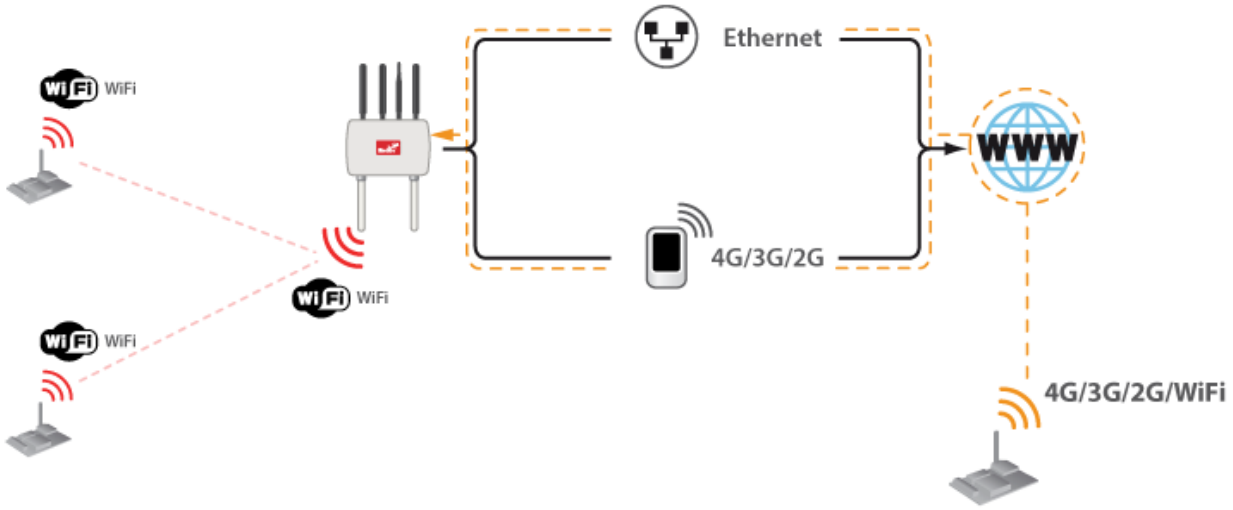


Figure : HTTP data reception

Meshlium accepts POST and GET requests in any of its interfaces so Waspmotes are capable of sending frames, through GPRS, 3G, 4G or WiFi modules, via HTTP requests. Meshlium, through HTTP requests is capable of:

- receive frames from 4G/3G/GPRS/GSM, WiFi or Ethernet via HTTP.
- parse these frames.
- store the data in local Database.
- synchronize the local Database with an external database.

Frames received by this method are stored the same way that RF frames, and are identically processed at synchronization stage.

No configuration of any kind is needed to use HTTP. If HTTPS is needed, certificate configuration would be needed in many cases (self signed certificate is included with Meshlium).

Like the case of RF modules reception, the user can add his own sensors.

10.3. Capturer

The Capturer plugin is where the user can check most recent data received in order to check if the nodes are sending information.

It can be found in:

Sensor Networks → Capturer

Capturer plugin have several tabs where the user can see recent data received, manage external database synchronization and perform some local database operations.

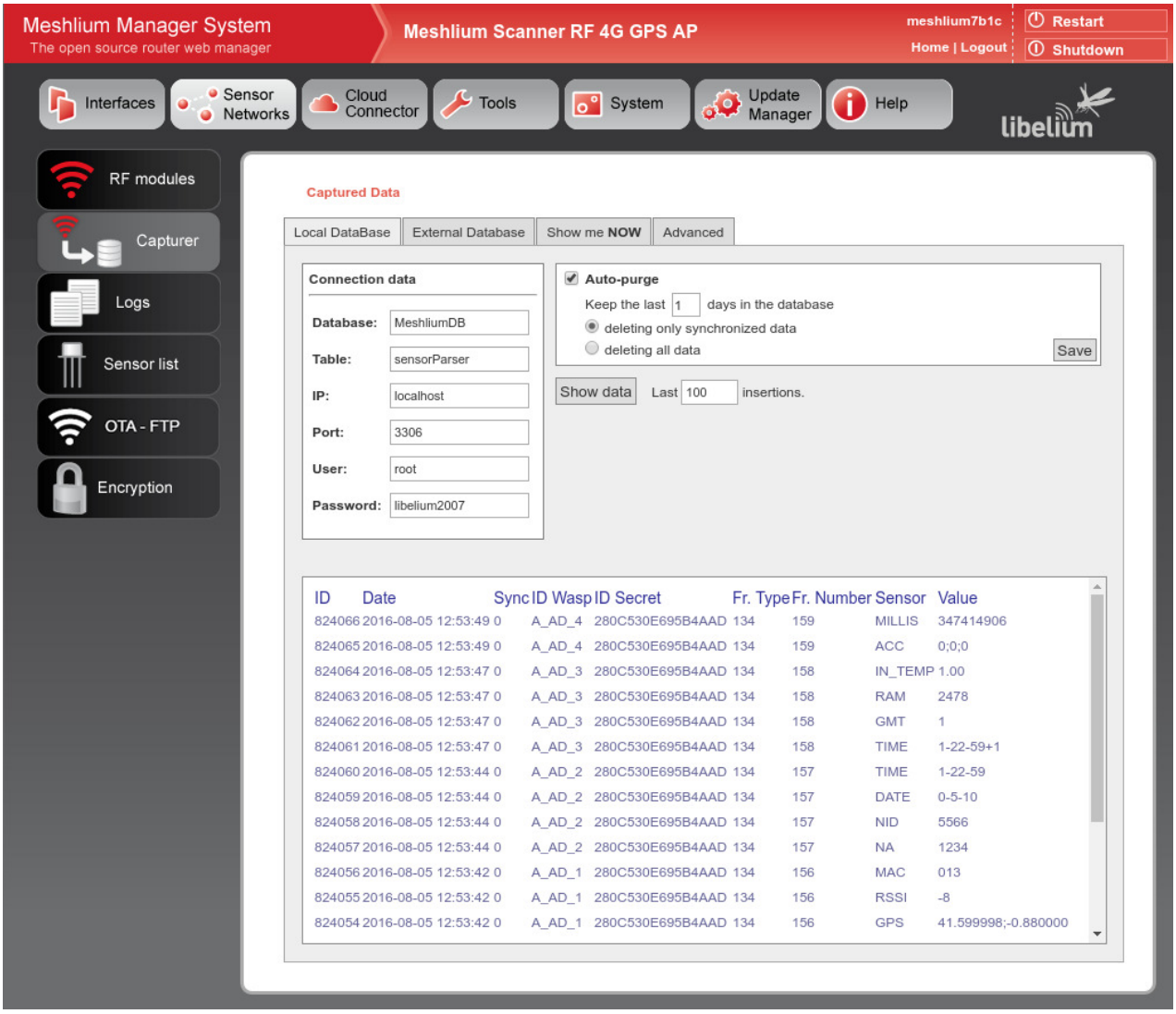


Figure : Capturer plugin

10.3.1. Local database

Meshlium has a MySQL database up and running which is used to locally store the information captured. In the "Local Data Base" tab the user can see the default connection parameters.

- **Database**: MeshliumDB
- **Table**: sensorParser
- **IP**: localhost
- **Port**: 3306
- **User**: root
- **Password**: libelium2007

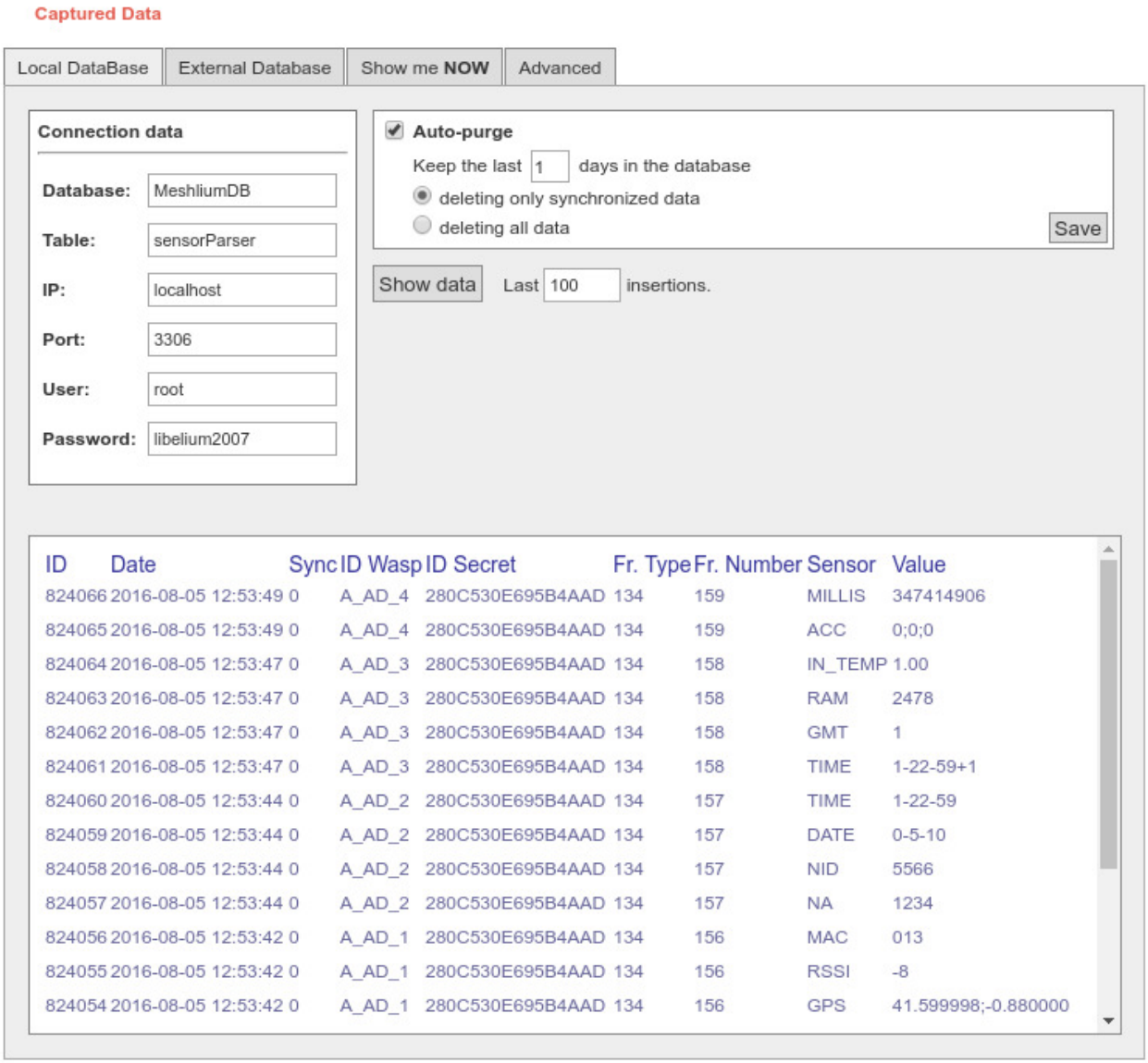


Figure : Local database tab

In this tab the user can:

- Show last insertions, up to 500.

Figure : Show last data

- Setup Auto-purge. This function allow to program a daily maintenance in the local database that deletes old data, keeping only the number of days configured, and allowing to delete synchronized data (only external database) or all data.

Figure : Autopurge setup

10.3.2. External Database

Meshlium can synchronize all the sensor information stored in the local database to an external MySQL database managed by the user.

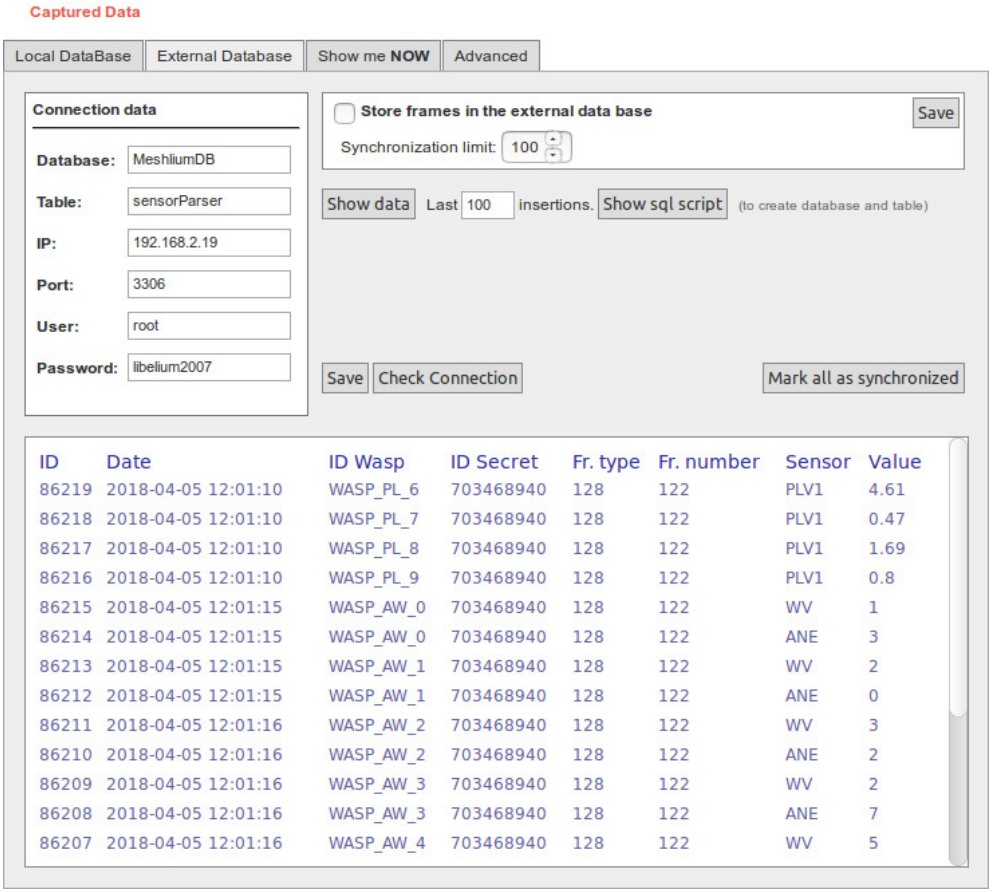


Figure : External database tab

In this tab the user can:

- Setup the parameters of the external database and check the connection.

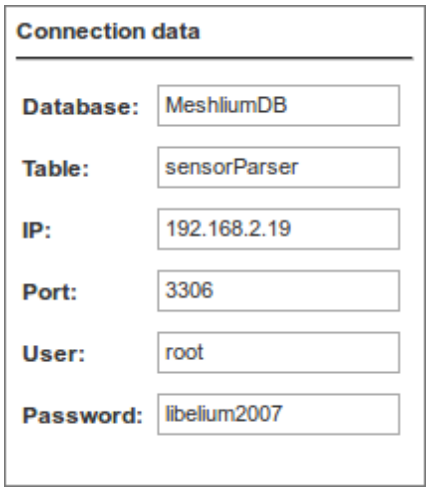


Figure : External database setup

- Enable or disable the synchronization and select the number of fields sent per synchronization iteration.

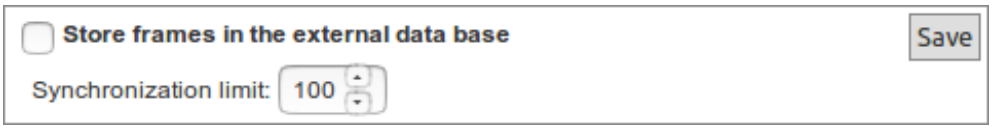


Figure : Control to enable or disable synchronization

Show last data inserted in the external database (up to 500 data).

Figure : Show last inserted data

Show the SQL script used to create the database and table needed for the synchronization.

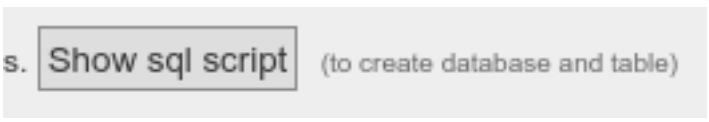


Figure : Show SQL script

Mark all data in the local database as synchronized so it will not be sent to the external database.

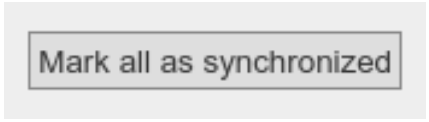


Figure : Mark as synchronized button

The steps to setup the synchronization are:

- Before configuring anything, make sure you have a MySQL database working under your control. Make sure the database listen to connections in an external IP.
- Press the “Show SQL script” button, copy the SQL code. You can modify user, password, database name and table, as long as you change the setup of the connection to match.

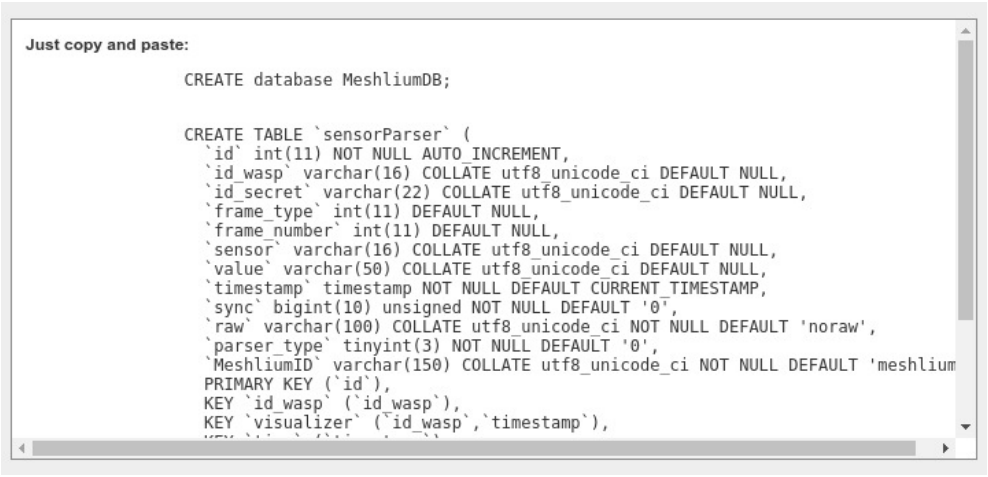


Figure : SQL script

- Enter the connection settings and press “Save” button. You can check the connection now to ensure the settings are correct.
- Enable the service with the checkbox and save.

The synchronization service runs every 60 seconds and synchronizes up to 100 data every loop. The service synchronizes first newer data, as it is more relevant for decision making. This could make data in external database to be out of order. As every data has a timestamp, this should not be a problem for using the data in any external application.

10.3.3. Show me Now

In this tab the user can show the last frame received. The user can show only last frame or can specify if the information will be updated periodically with the defined interval just checking the “Use the Defined Interval” button.

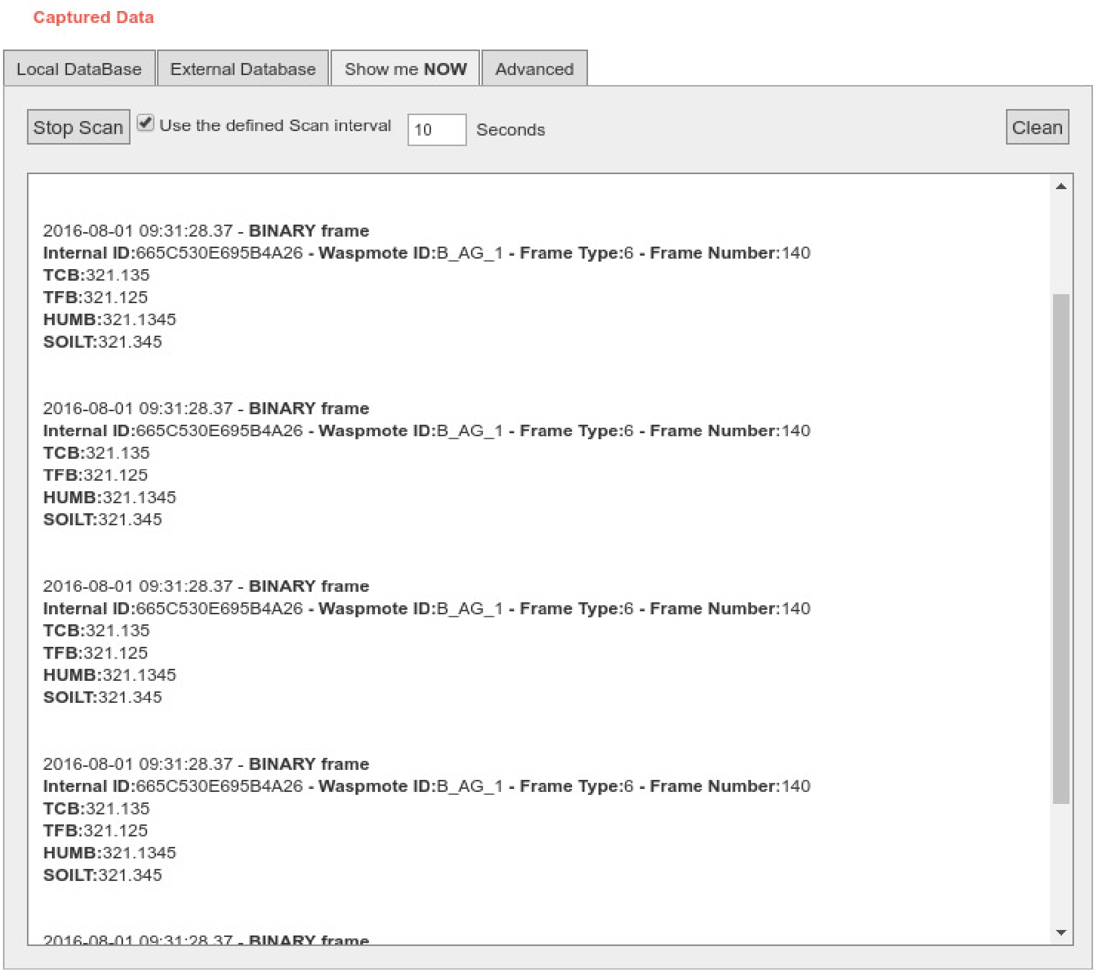


Figure : Show me Now tab

The screen can be cleaned with the button in the top right.

10.3.4. Advanced database options

This tab shows information about local database.

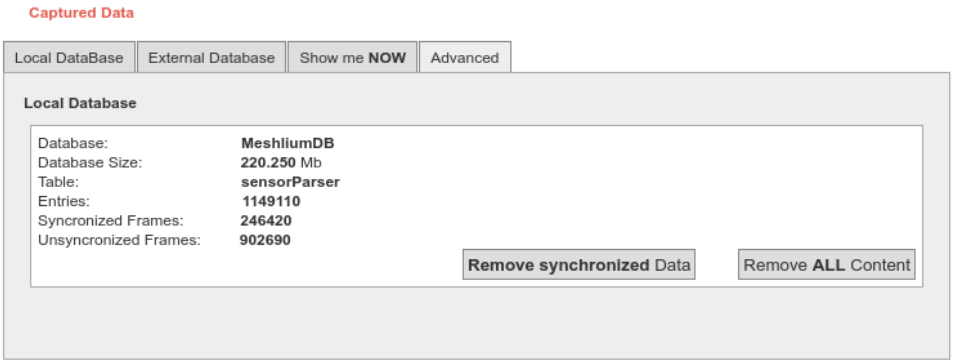


Figure : Mark as synchronized button

It shows:

- Database name.
- Database size.
- Database table used.
- Number of total sensor entries.
- Number of frames already synchronized with external services.
- Number of unsynchronized frames.

There are too two controls to:

- Remove synchronized data. It removes from the database all the frames already synchronized with external database. Be careful as this could give unexpected results if you are using several cloud or external services. A confirmation will be prompted.
- Remove ALL content. This removes all the sensor information from the database. A confirmation will be prompted.

Important:
The sensor data will be permanently deleted from the database and will be impossible to recover. Be sure to have a backup of the database before deleting the content.

10.4. Logs

In this section the user can see the last lines of the logs of frames and sensor data received.

Figure : Logs visualizing plugin

- The “Refresh” button will load again the log files.
- The “Delete logs” button will delete the files, allowing to clean some space in the device.

10.5. Sensor list

In this section, the user can view the list of available sensors in the system and add or delete user custom sensors.

By default, Meshlium recognizes all Libelium official sensors. The button “Update sensors” updates the Meshlium unit with the latest Libelium’s official sensor list: Meshlium will connect to Libelium’s servers and will download the latest configuration files.

All additional sensors (not officially integrated by Libelium) must be specified by the user. Users can add and remove custom sensors in an easy and simple way on the Manager System. The update process will not change the “User sensors” list.

To add a new sensor the user must complete the fields:

- ASCII ID: sensor id for ASCII frame.
- Fields: This field specifies the number of sensor fields sent in the frame. This helps to calculate the frame length.
- Type: type of fields:
  - uint8_t
  - int
  - float
  - string
  - ulong
  - array (ulong)
- Units: Units for the sensor added.

Once all fields are filled in, click on the button “Add sensor”.

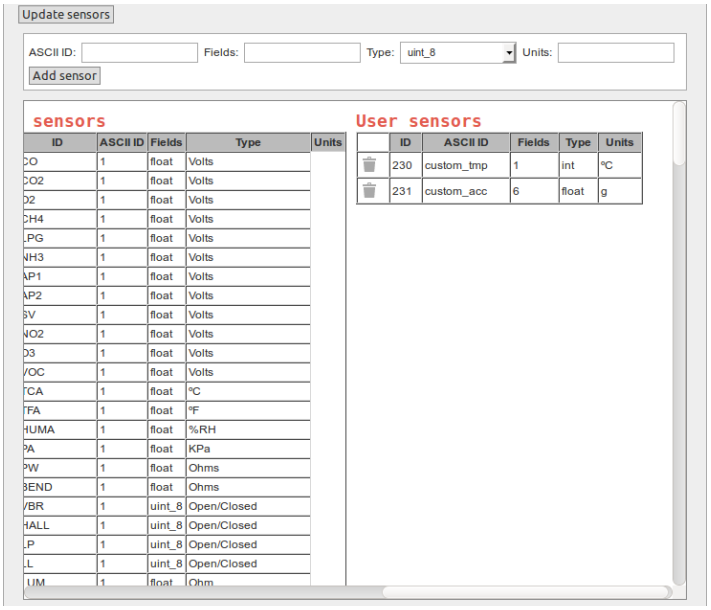


Figure : Sensor list plugin

***Note**: Extensive information about how to build the frame is available on the “Waspmote Data Frame Guide”.*

To delete sensor the user must press the garbage can that appears to the left of the description of the sensor. To complete the action should accept a confirmation message.

10.6. OTA via FTP

Meshlium can also be used as an FTP server to prepare the binary files to be downloaded by Waspmote.

For more info about Over the Air Programming go to:
https://www.libelium.com/development/waspmote/documentation/over-the-air-programming-guide-otap/

This feature allows reprogramming Waspmote using an FTP server (inside Meshlium) and FTP client (Waspmote itself).

There are two basic steps involved in OTA procedure:

- **Step 1**: Waspmote requests a special text file which gives information about the program to update: program name, version, size, etc.
- **Step 2**: If the information given is correct, Waspmote queries the FTP server for a new program binary file and it updates its flash memory in order to run the new program.

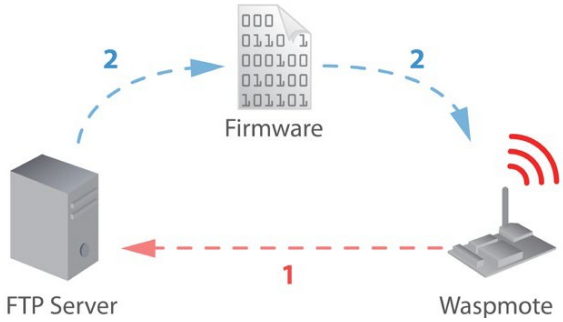


Figure : OTA via FTP protocol

Besides, a default user is configured in Meshlium FTP Server with the following settings:

- user: ota
- password: libelium

This user directly connects to the following path in Meshlium's system directory where the application creates all the binary and UPGRADE.TXT files:

/mnt/user/ota

Inside "Sensor Network" there is the section OTA - FTP. Users can prepare the binary files to be downloaded by Waspmote. Then, the user can generate UPGRADE.TXT text file necessary to do OTA with 4G/3G/GPRS/GSM/WiFi via FTP.

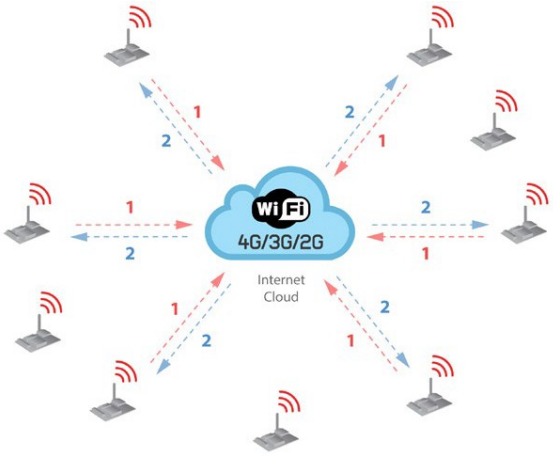


Figure : OTA-FTP in Meshlium

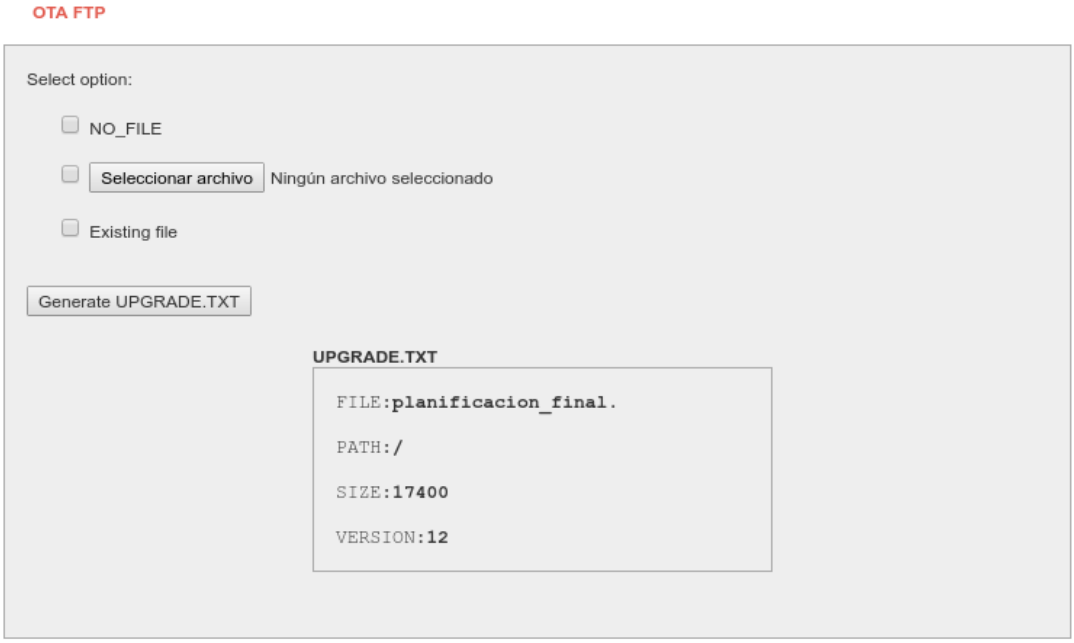


Figure : OTA-FTP plugin

Firstly, there are three possibilities to be chosen:

- Select NO_FILE to inform Waspmote that no OTA is necessary.
- Select a new file generated by the Waspmote platform IDE so as to update the Waspmote's program.
- Select an existing binary if the user needs to update to an older program. The files are stored in the following path: /mnt/user/ota.

Secondly, the program version is always set by the user before generating the new UPGRADE.TXT file. There is a specific input to indicate the program version. It must be defined as a 1-unsigned-byte number (range: from 0 to 255).

Finally, there is a button to generate the new UPGRADE.TXT file.

Once these steps have been completed, the binary file and the proper UPGRADE.TXT file will be ready for the Waspmote devices deployed which try to perform OTA via FTP transmission. This file is shown in the window of the application representing the actual binary prepared for OTA.

11. Meshlium Visualizer

Meshlium Visualizer is a plugin which plots graphs and maps with the data stored in the database. It can also export data in common formats. Meshlium Visualizer is a special software feature only available in the Meshlium units included in the IoT Vertical Kits (Smart Cities IoT Vertical Kit, Smart Water IoT Vertical Kit, etc) and Solution Kits with Meshlium. The service is valid for Libelium standard sensors (it displays the "Catalog sensors").

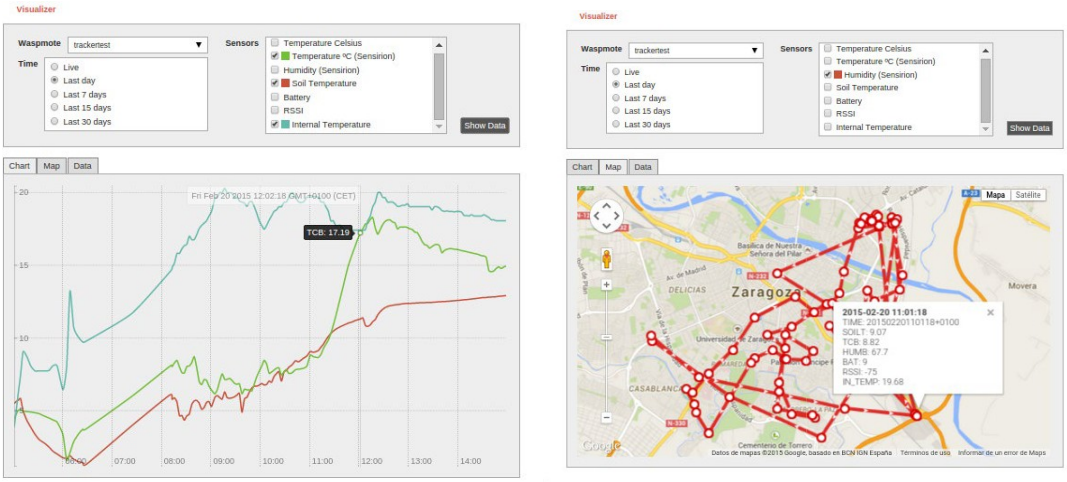


Figure : Meshlium Visualizer can plot graphs and geo-locate data on maps

Please note that this is a paid service. Each Meshlium comes with 100 visualizations. After 100 visualizations, users can contact Libelium Sales Department (sales@libelium.com) if they want to continue using the service.

11.1. Working with the Visualizer

On the top of the page you can use a simple form to make all your queries. To do so, just follow these steps:

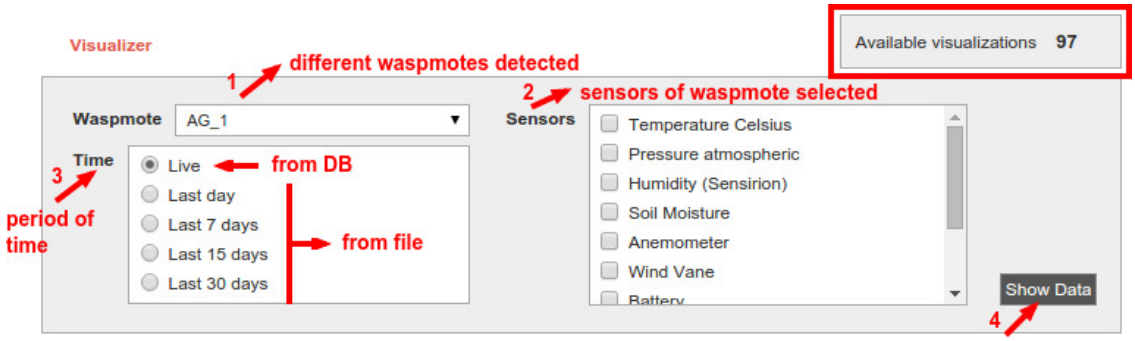


Figure : Filling Meshlium Visualizer's form

1. Select one Plug & Sense! from the list. All Plug & Sense! units with frames in the database will be shown.
2. Once a Plug & Sense! Is selected, all its sensors will be loaded. This process is repeated each time you change the selected Plug & Sense!.
3. Select the period of time you want to see in the chart. The "Live" option reads directly from the database, while the rest options read from a file generated everyday by the service cron. For each Plug & Sense!, cron generates 4 files each day, one for the last day, other for last 7 days, other for last 15 days and other for the last 30 days.
4. Hit on the "Show Data" button and, if your query has results to show, Meshlium Visualizer will show them. The remaining visualization number will decrease in one unit. If the query does not have any results, a message will appear notifying the situation; the available visualizations remain without changes.

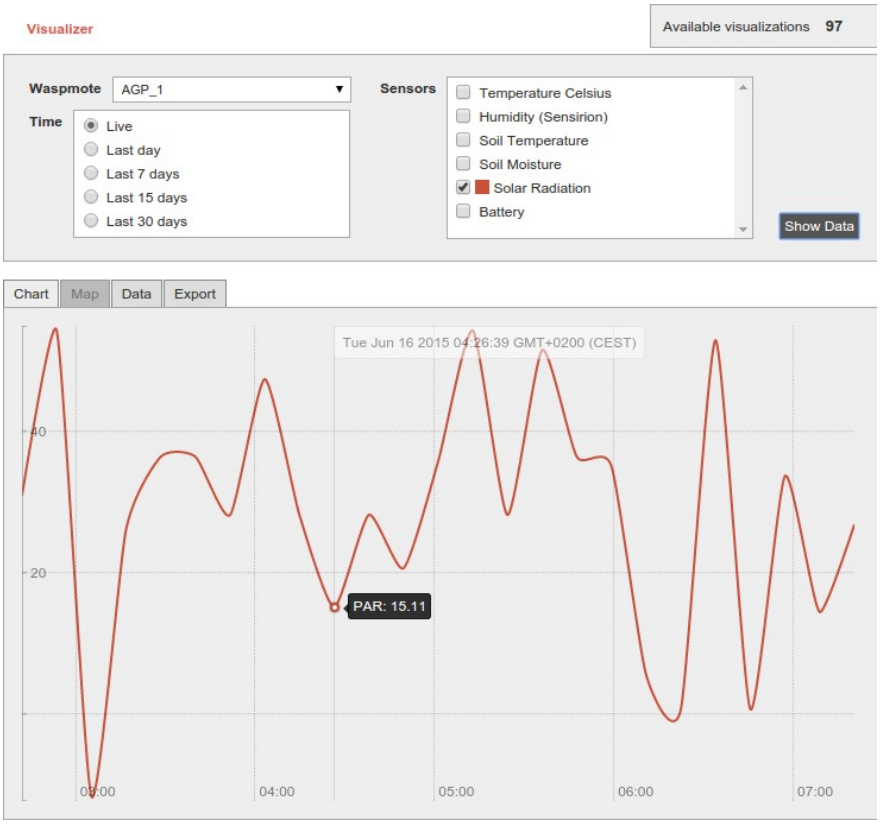


Figure : Meshlium Visualizer showing one graph

If your query has GPS results (data frames with GPS information), the “Map” tab will be shown. If it is not the case, like in the previous picture, this tab remains disabled.

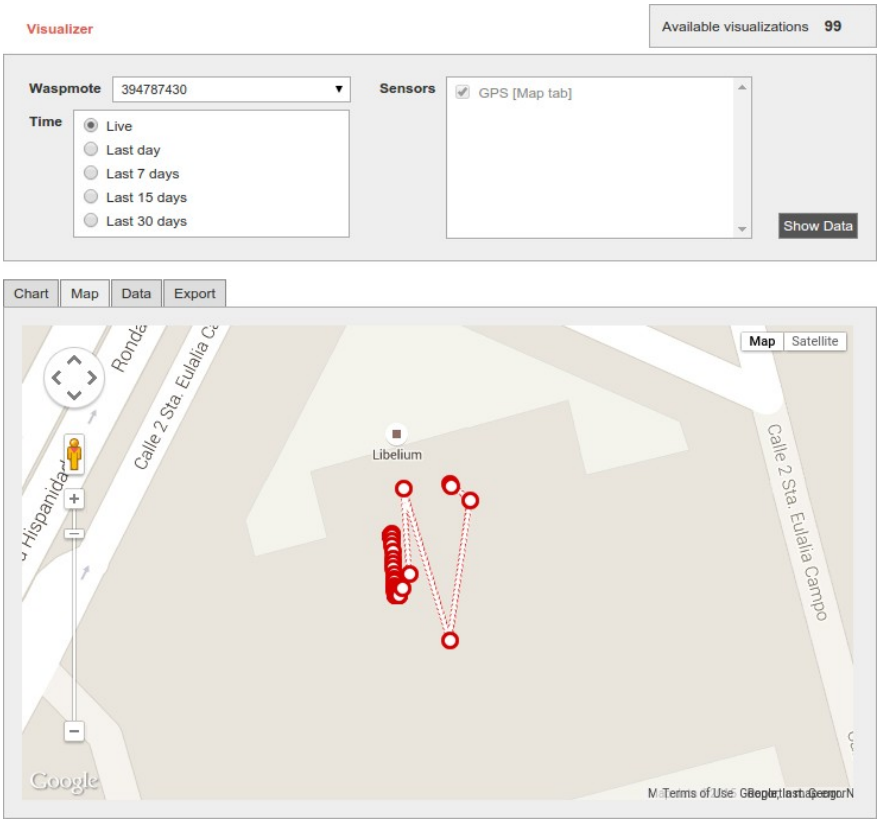


Figure : Locating nodes on a map thanks to Meshlium Visualizer

The “Data” tab shows a list of sensor values, ordered by time.

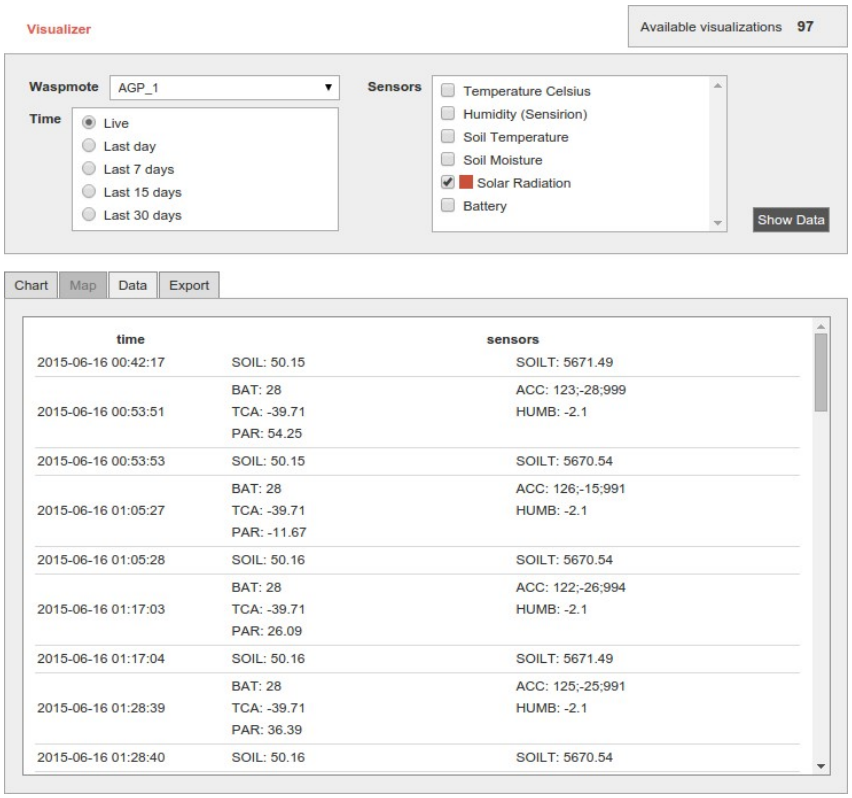


Figure : Meshlium Visualizer showing the Data tab

The “Export” tab shows two calendars to select the initial and final date. This feature does not take into account the block on the top of the page, it will export all data from all Plug & Sense! units between these dates. Data can be exported in 5 formats (CSV, SQL, XML, TXT and HTML) and compressed in ZIP.

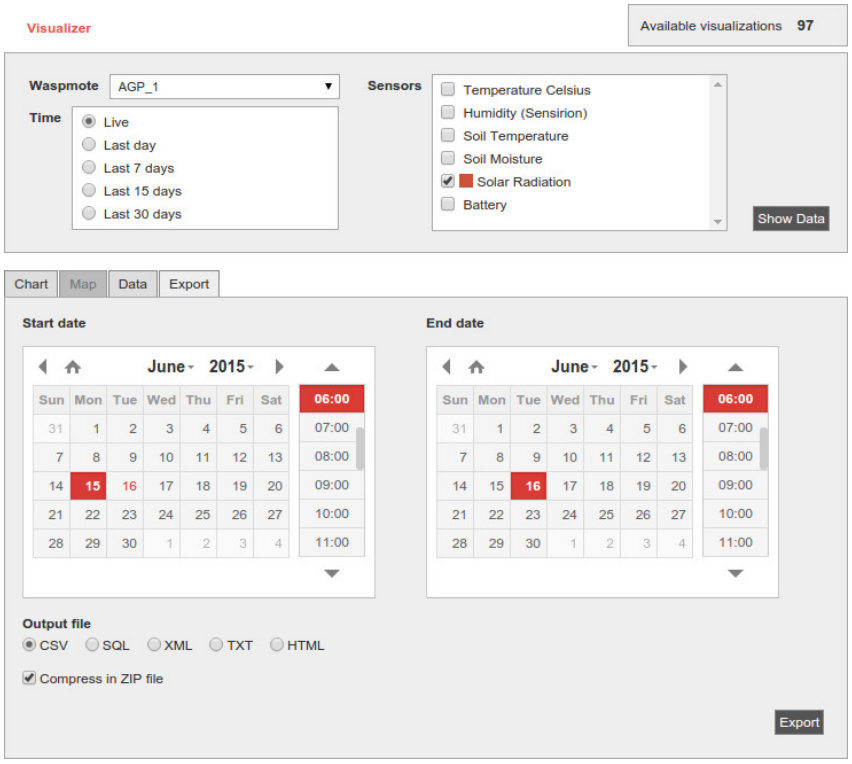


Figure : Configuring Meshlium Visualizer to export data

12. Cloud Connectors

The aim of this chapter is to introduce the user to the Meshlium Cloud Connector functionality. This section will help you to connect your Meshlium to a third party cloud platform.

Only sensor data can be sent to the cloud services.

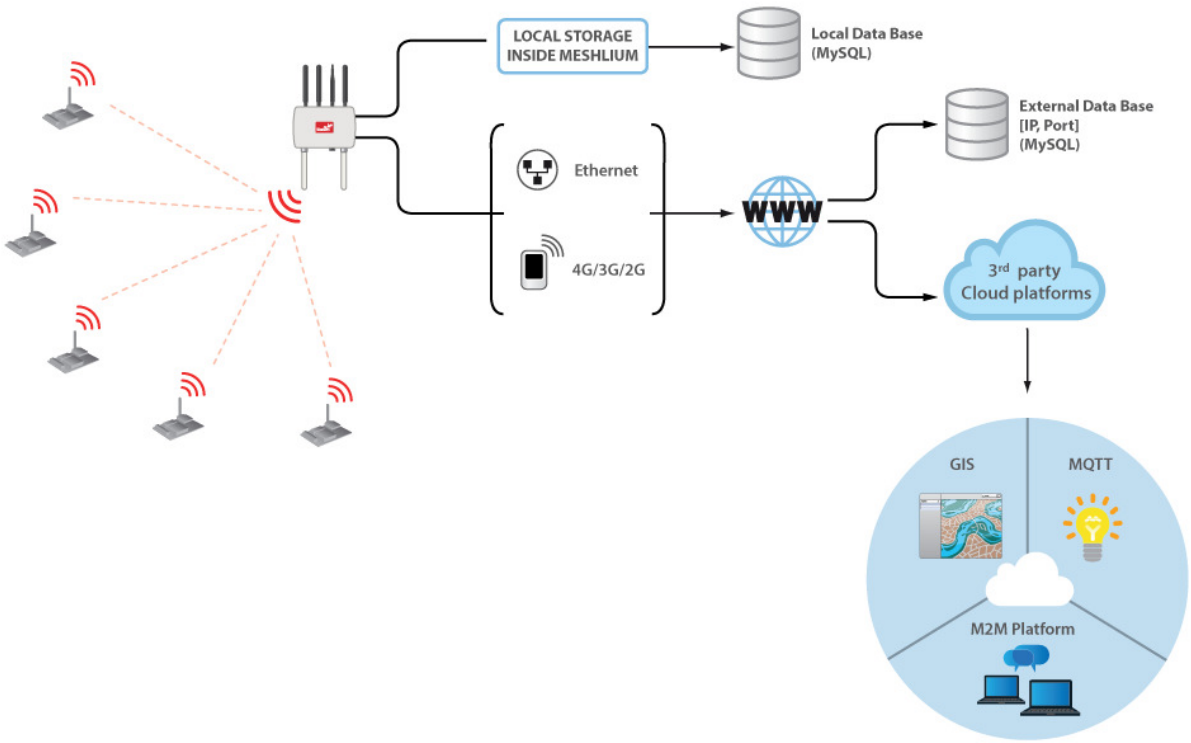


Figure : Cloud connector diagram

Interfacing Meshlium with 3rd party cloud services should be the last step the user develops in any project. The user should analyze if the use of clouds is needed, and if so, that will be the last step in the project. Before trying clouds, make sure all the Waspmote units are sending frames to Meshlium, and Meshlium is receiving and inserting them on the local database properly.

What is a cloud platform?

Cloud computing is a major change in our industry. One of the most important parts of that paradigm are cloud platforms. This kind of platforms let developers write applications that run in the cloud, use services provided from the cloud or both.

Meshlium Cloud Connector

Meshlium runs a set of scripts for implementing the data synchronization from its internal database “to the cloud”. In other words, those scripts send data to webservers where the cloud service providers host their clouds. Those scripts are called Cloud Connector.

We have divided the Cloud Connector into 3 groups: “Premium”, “Advanced” and “Basic”.

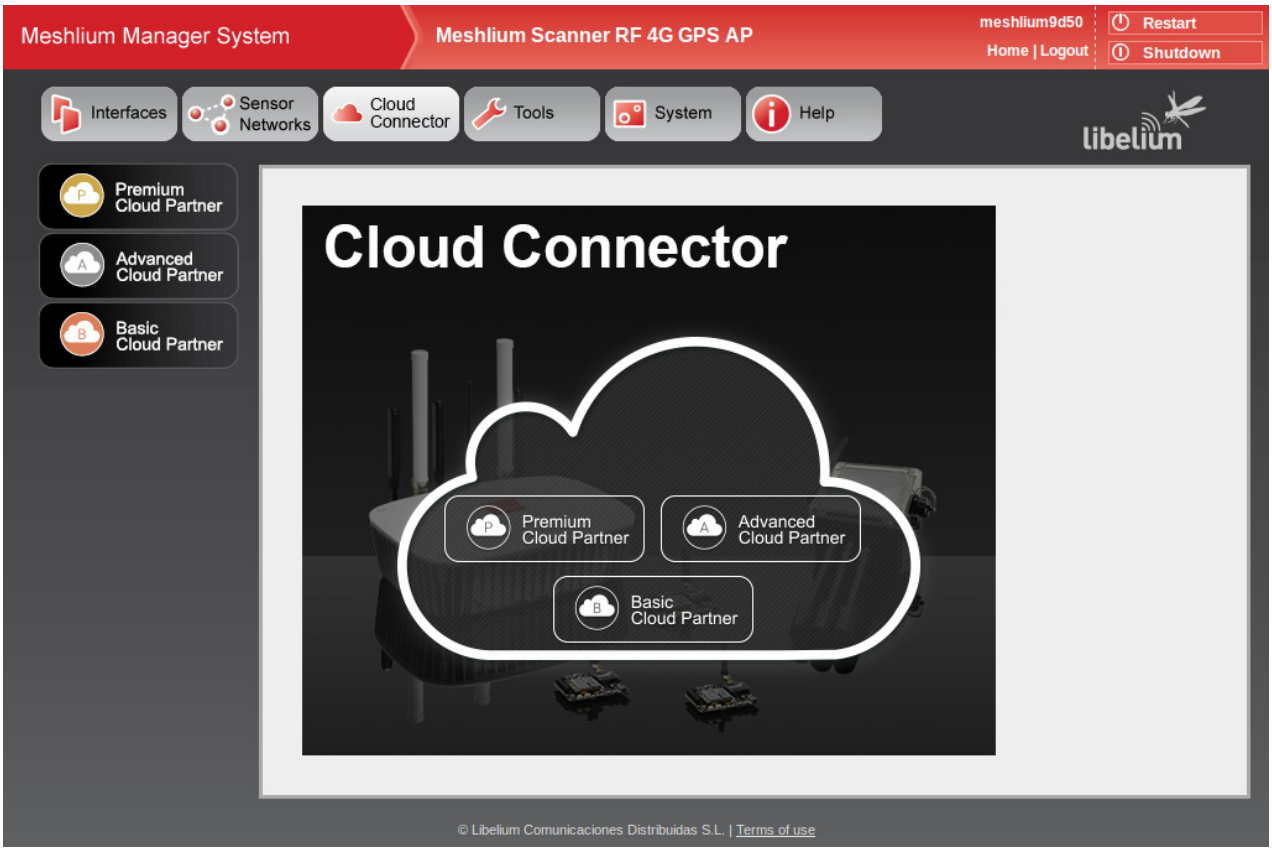


Figure : Cloud Connector main menu on the Manager System

12.1. Premium Cloud Partners

12.1.1. Arrow

Arrow Connect IoT Cloud Platform

Arrow Connect is a software solution that plumbs the data from the edge to the cloud. Developed by Arrow Electronics and designed with security, scale, flexibility, device management, open APIs and extensibility as core tenants enabling broad use cases across multiple industries. You can provision, control, import, assign, activate, update, suspend, replace, deactivate, and more, all from a single platform. Arrow Connect makes device management easy so you can focus on driving business value via data analytics and machine learning tools.

Visit http://iot.arrow.com/developer.html for more information.

Register Meshlium in Arrow Connect

To request for a developer account in Arrow Connect platform, follow the Developer Registration process on https://portal.arrowconnect.io/#/signup or please email to tnguyen@arrow.com including the following information:

- Company name.
- Full name.
- Title.
- Email address.
- How did you hear about us?
- Short description of your project.

Log on to the portal at https://portal.arrowconnect.io using your developer account.

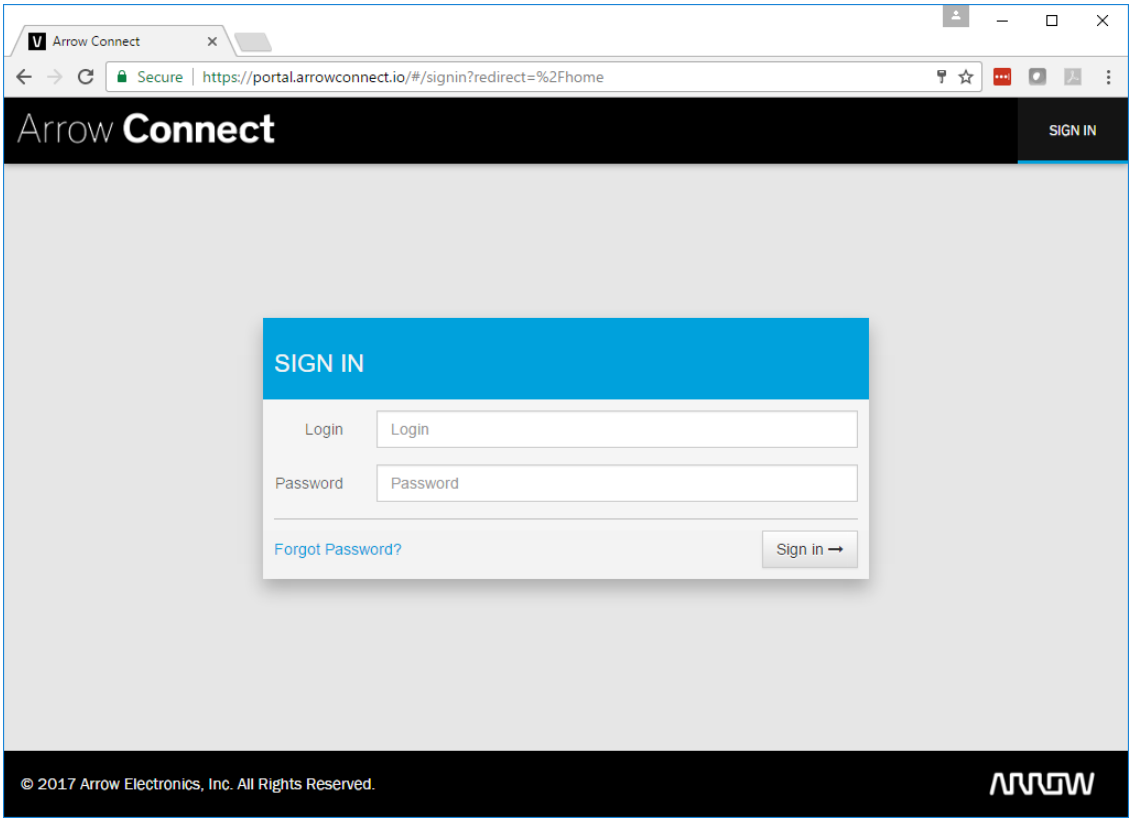


Figure : Arrow portal

With the developer account, you have a private application instance with full admin access. Upon logging in, click on Administration > Access Keys, then click on the Master Key. You can save the “Client Encrypted API Key” and “Client Encrypted Secret Key” to the notepad. This pair of keys will be used to configure the Meshlium to connect to your cloud account.

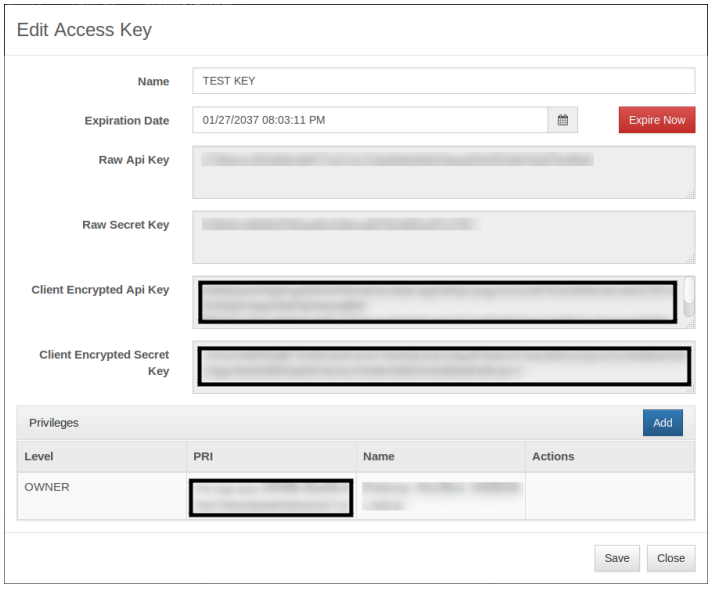


Figure : Access Key

Configuration

Paste the keys above into the 2 text areas as screen-shot below.

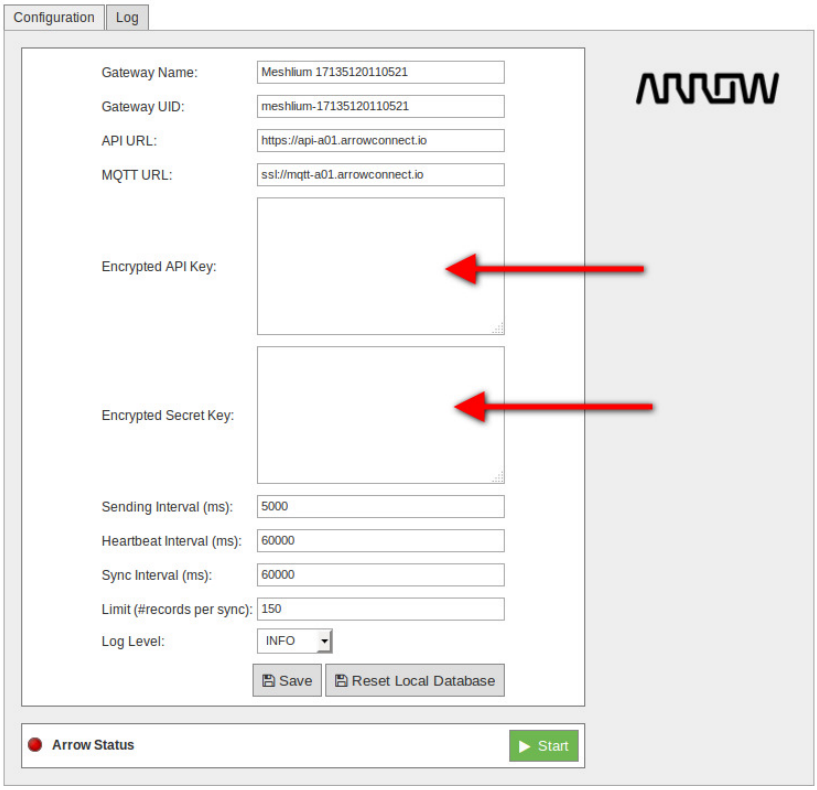


Figure : Configure Arrow

All the default configuration values should be appropriate and you do not need to change them. Click on the "Save" button for storing the configuration fields.

- API URL.
- MQTT URL.
- Encrypted API Key.
- Encrypted Secret Key.

After clicking the "Save" button, you also need to click on "Reset Local Database", otherwise your changes will not be affected.

In all cases, to prevent local database corruption, the configuration should never be updated while the Arrow Connect IoT Cloud Connector is running. Use the Start/Stop buttons as described below to stop it before making changes, restart it after saving changes.

Controlling synchronization

Launch the Meshlium Arrow Connect IoT Cloud background process (Start button). The program will search for the received frames on the local database, and will send them to the Arrow Connect IoT Cloud Platform. The status indicator displays the current state. Green colour means running.

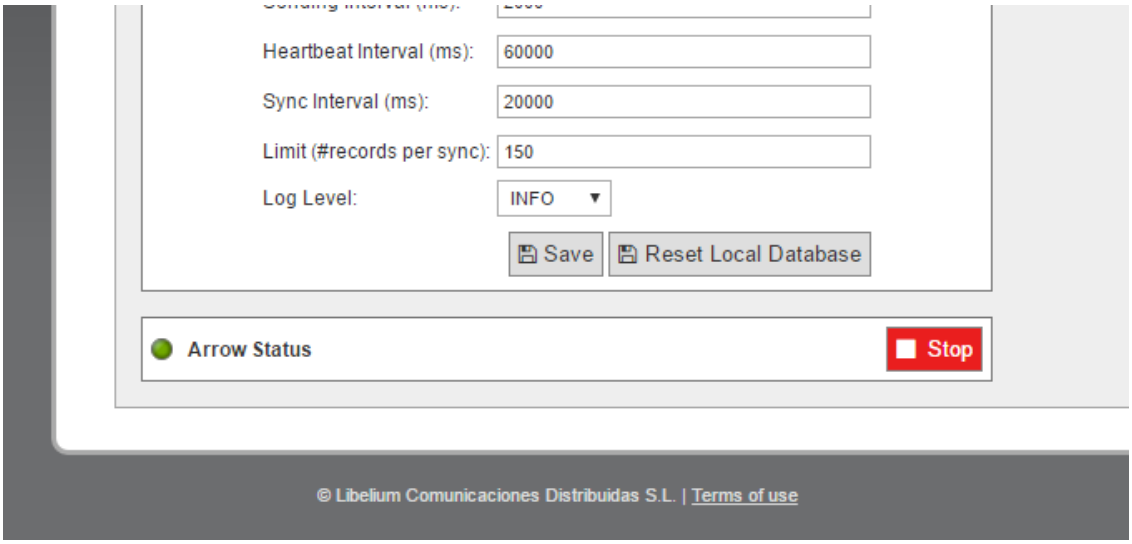


Figure : Synchronization stopped

Stop the Arrow Connect program anytime by clicking on the "Stop" button. Red colour means stopped.

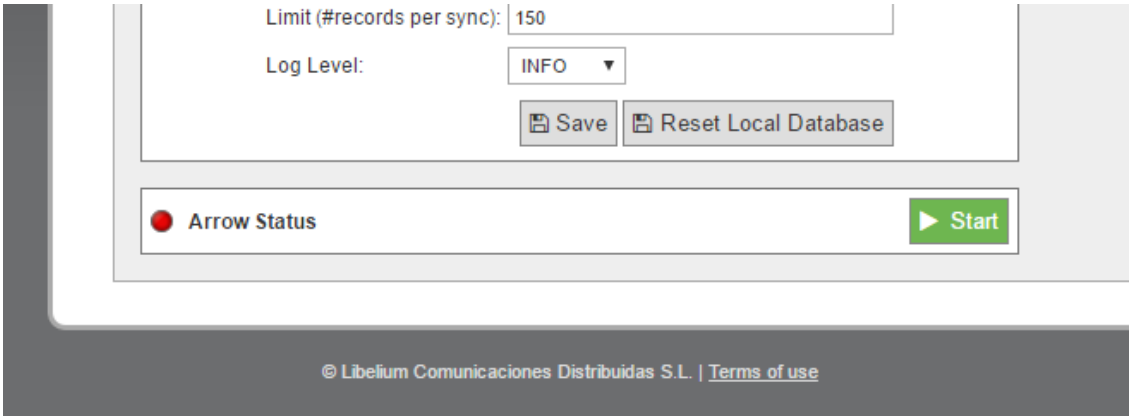


Figure : Synchronization running

While the program is running, log on to the Arrow Connect portal to check that the Meshlium and Waspmotes have been properly registered and are sending telemetry data. See some example screen-shots below:

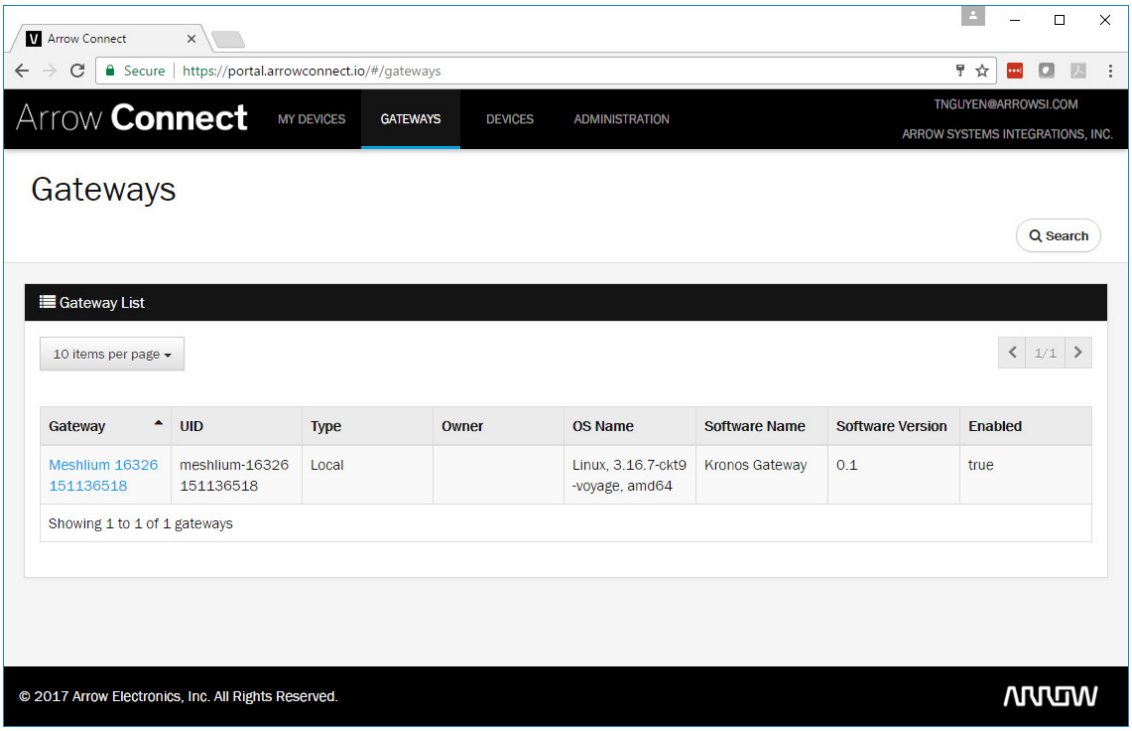


Figure : Arrow gateways display

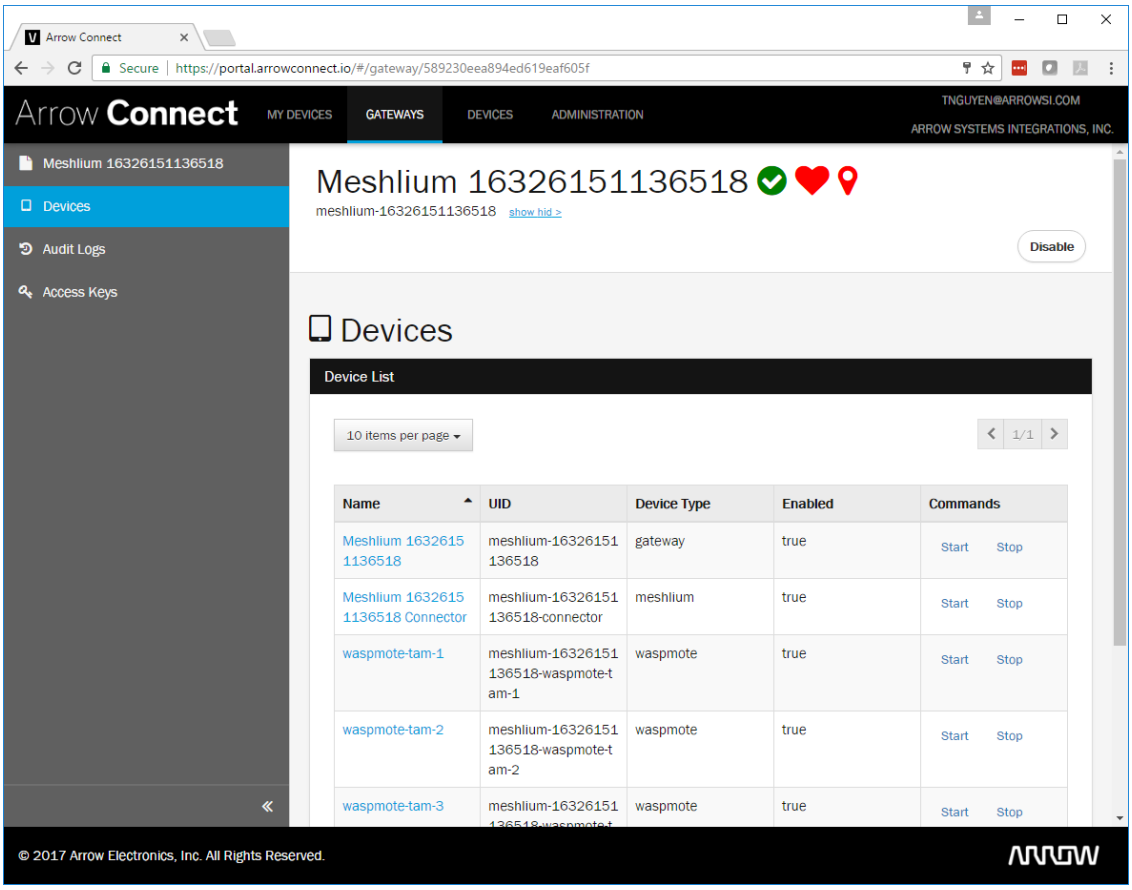


Figure : Arrow Meshlium display

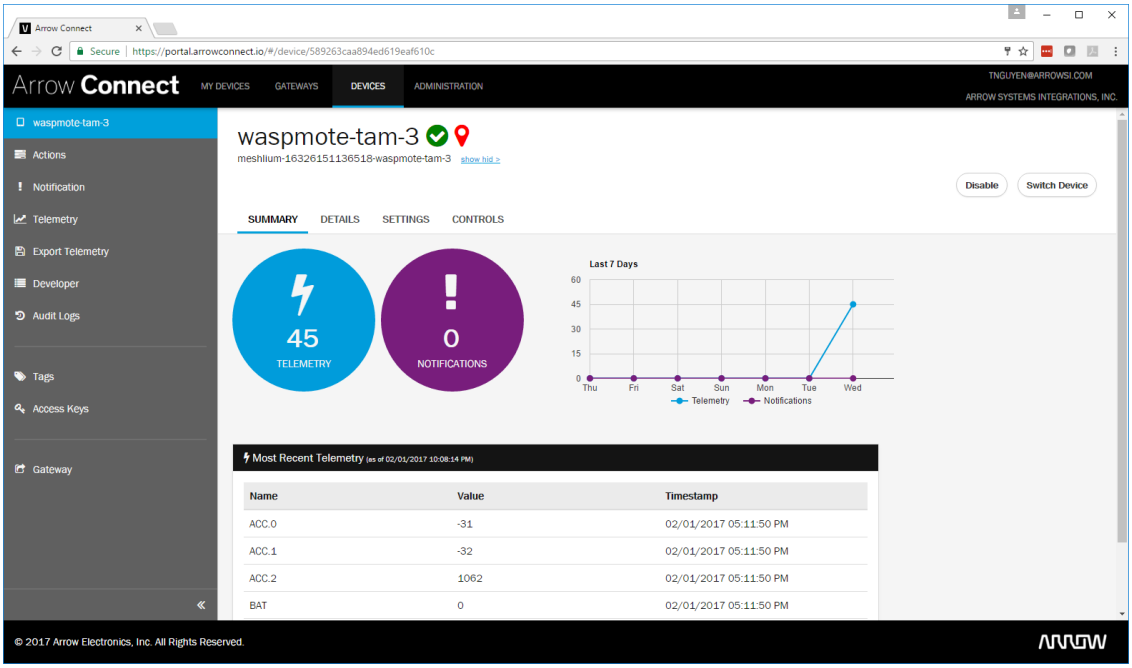


Figure : Arrow Waspmote display

12.1.2. ElementBlue - RightSensor

RightSensor is a solution company designed to provide sensors, services and support for Industrial Internet of Things projects.

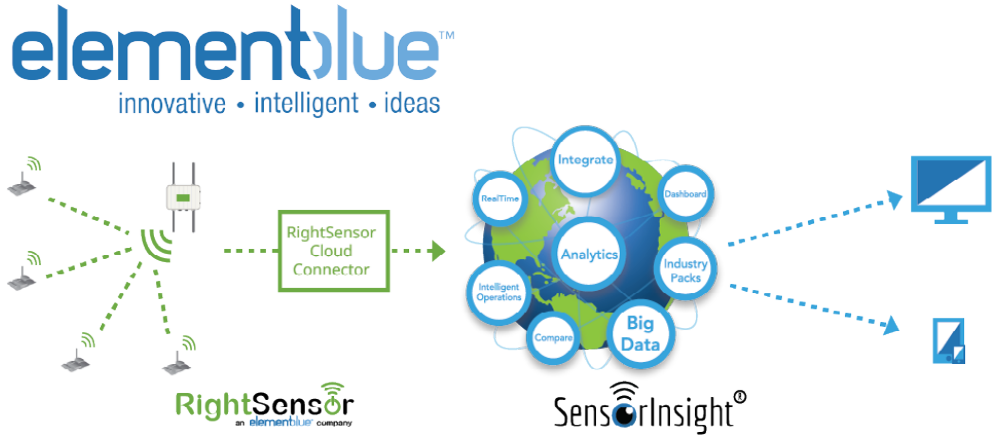


Figure : ElementBlue RightSensor Cloud

The RightSensor cloud connects interface works directly with Element Blue's SensorInsight® Industrial Internet of Things Platform.

SensorInsight® is used by companies for the integration, display and detailed analysis of data from multiple sources providing an environment where users can view and compare real-time and historic data patterns, receive alerts and notifications, and trigger workflows and actions based on the data for use in industrial environments.

To use the service you must have an account with Element Blue's SensorInsight Industrial Internet of Things Platform. For more information visit: www.sensorinsight.io and www.rightsensor.com.

Configuration

By expanding the RightSensor menu item on the list you can see the form in which to set your connection parameters. The form accepts the following 4 parameters:

- **Client ID**: This is a unique ID provided to you from the SensorInsight service.
- **Gateway ID**: This is an ID you provide to uniquely identify this Meshlium device.
- **User Name**: This is the user name required to send your data to the SensorInsight cloud.
- **Password**: This is the password required to send your data to the SensorInsight cloud.

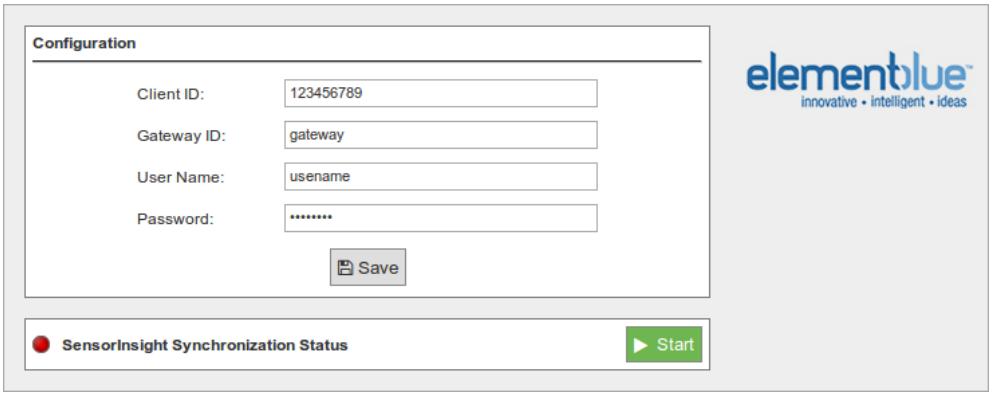


Figure : Configuring ElementBlue RightSensor in Meshlium

These parameters can be obtained from your SensorInsight account page. Learn more at www.sensorinsight.io.

Controlling synchronization

To launch the cloud connector service and start sending your data to RightSensor press the “Start” button.

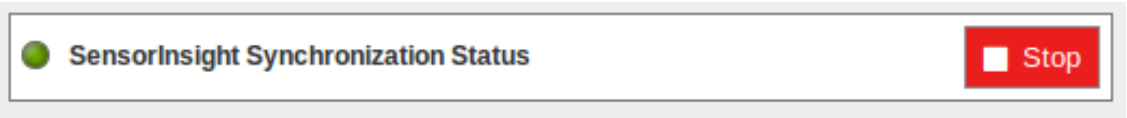


Figure : ElementBlue RightSensor synchronization service is running

You can stop at any moment clicking on the “Stop” button.

Figure : ElementBlue RightSensor synchronization service is stopped

Problems

The RightSensor Cloud Connector service has built-in logging and debugging capability. Please contact RightSensor at www.rightsensor.com for troubleshooting information.

12.1.3. Ericsson DDM

The DDM (Device and Data Management) cloud connector will integrate Meshlium as a DDM gateway with minimal effort and configuration.

More information on DDM and IoT Accelerator:

https://www.ericsson.com/ourportfolio/products/iot-accelerator?nav=productcategory002|fgb_101_973

Register Meshlium in DDM

Register a gateway with the pre-configured Meshlium gateway type in your device network in DDM. If the Meshlium gateway type is missing in your instance of DDM, please contact DDM support. If you are unfamiliar with DDM, you can read more about how the platform works on http://docs.appiot.io/.

Configuration

The figure below shows the main configuration page for the DDM cloud connector. The following is a description of the components in the connector's User Interface.

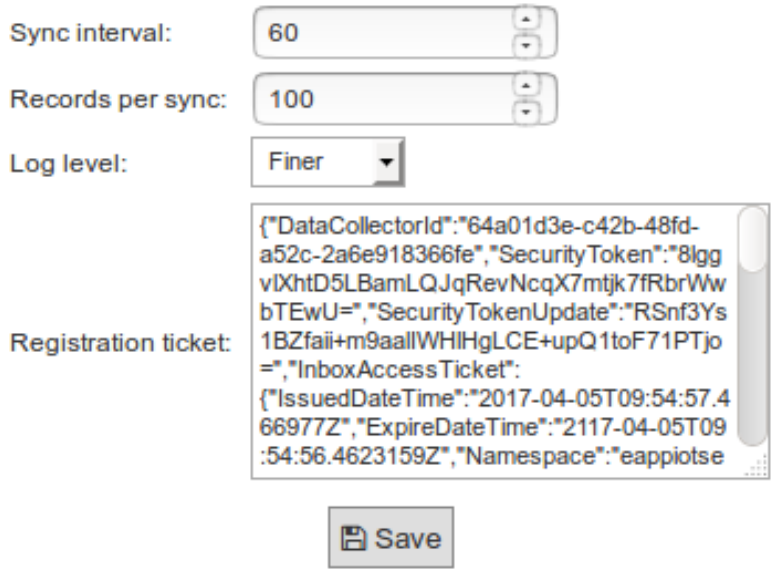


Figure : Configuration overview

Sync Interval: Sets the number of seconds to synchronize with DDM. The default value is 60 seconds. Please take into consideration when setting this value that Meshlium is a resource-constrained device. The shortest synchronization interval is 30 seconds.

Records per sync: Sets the number of sensor values read from the Meshlium's database per synchronization. The default value is 100 records per synchronization. It is not recommended setting this value any higher than 200 due to the memory limitations of Meshlium.

Log level: This controls the log output from the DDM connector. The log levels Severe and Warning will only output messages that in some way may affect operational functionality of the cloud connector. All other levels of logging, except Off, will output general events from the cloud connector. The log level Info will focus on events regarding actions taken by the connector. Other levels, like Fine, will output third party library logs as well.

Registration ticket: The registration ticket that will be used by the gateway integrated in the cloud connector.

Sensor mapping: Toggles the sensor mapping table used to translate sensor types between Meshlium and DDM. An example of the table can be seen in the next image, where the left hand side lists the sensors from Meshlium and the right hand side lists the sensors URI from DDM. Note that the empty value represents an unmapped sensor. The DDM cloud connector will only try to synchronize sensor values that are not empty.

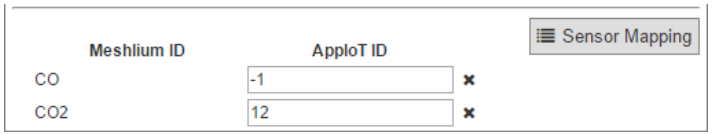


Figure : Sensor mapping

As shown in the figure below, add custom mappings in the case that your sensor type is not present in the standard set of sensor types on the Meshlium. Click the 'X' next to the mapping to remove a mapping.

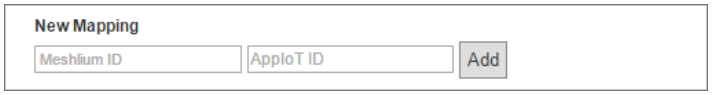


Figure : Adding a sensor mapping

Serial numbers: The drop-down lists all the Waspmotes available and/or registered in DDM. The configured state between DDM and the Waspmotes are displayed next to the Waspmote ID. There are 3 states: Not registered, Registered and Registered (Not seen). The Not registered state represents a Waspmote transmitting to the Meshlium that is not registered in DDM. The Registered state represents a Waspmote transmitting to the Meshlium that is registered in DDM. This is the state where the connector synchronizes data with DDM. The last state, Registered (Not seen), is when a Waspmote is registered in DDM but has no record of transmitting to the Meshlium. The button next to the drop-down will copy the selected Waspmote ID to your clipboard.

Figure : Sensor mapping

DDM Gateway Status and Start/Stop: The round icon indicates the current state of the connector. The icon will be either green or red depending on if the connector is running or not. The green start button will start the connector. When clicked, it changes into a stop button. To stop the connector press the red stop button.

Figure : Gateway status

Log tab: As seen the following screenshot, you can review the latest logs from the connector. The refresh button will reload the log window and display the latest logs. The delete button will empty the log file on the Meshlium unit.

Figure : Log overview

Setup

1. Stop the connector, if it was running.
2. Supply the registration ticket field with a valid ticket.
3. Start the connector. It will begin listening for notifications from DDM.
4. Create device types in DDM to represent your Waspmotes. The cloud connector will receive notifications about devices and sensors modifications.
5. Set the desired synchronization interval and limit, please select these parameters with care.
6. Set the desired logging level.
7. Restart the connector. It will initialize internal structures according to newly created and modified sensors.
8. Map the Meshlium sensor types to the corresponding sensor hardware types in DDM.
9. If you are using user-defined sensor types on the Meshlium, make sure to add these to the connector's sensor mapping table.
10. A restart of the connector is always required for saved changes to take effect: stop the connector before saving, the press the "Save" button and the the "Start" button.

Notes

- When registering a Waspmote device in DDM, the device name used in DDM must correspond to the Waspmote name that the developer used in the Frame instance.
- The cloud connector can handle sensor types with multiple fields if they are defined in the sensors table on Meshlium. Example: ACC has three fields and translates to APPIOT0_ACC, APPIOT1_ACC and APPIOT2_ACC. The APPIOT prefix will appear on all multi-field types and the numbering is the index of each field value.
- The cloud connector does not support multi-field, user-defined sensor types.

12.1.4. Libelium Cloud Hive service

The Libelium Cloud Hive service enables to any sensor node to push data directly and in a secured way to the main cloud services.

This service also enables the user to manage their Meshlium units remotely.

For more information about the advantages of using the Hive, check its guide:

http://www.libelium.com/downloads/documentation/hive_technical_guide.pdf

Register Meshlium and Plug & Sense! units on Libelium's Services Cloud Manager

To start using the Hive you must have first registered your Plug & Sense! devices and your Meshlium units on the Services Cloud Manager, as explained in the "Service Cloud Manager" guide that you will find on the following link:

http://www.libelium.com/downloads/documentation/service_cloud_manager_guide.pdf

Configuration

You will use the configuration previously obtained on the Hive to certify your Meshlium as a valid sender of messages.

The Hive plugin is located in:

Cloud Connector → Premium Cloud Partner → Libelium Cloud Hive

In the configuration panel, the user can set:

- **User**: email used to access to the Services Cloud Manager.
- **Password**: Password used to access to the Services Cloud Manager.
- **Enable Remote Management**: When the parameter is on, you enable this Meshlium unit to be remotely managed. If the parameter is off, you will not be able to manage the Meshlium remotely.

Figure : Configuration panel

- **Sensor nodes**: In this section you can select the devices which will send data to the Hive. To be able to select any of the devices shown on the list, you must have registered them on the Services Cloud manager, as indicated before.

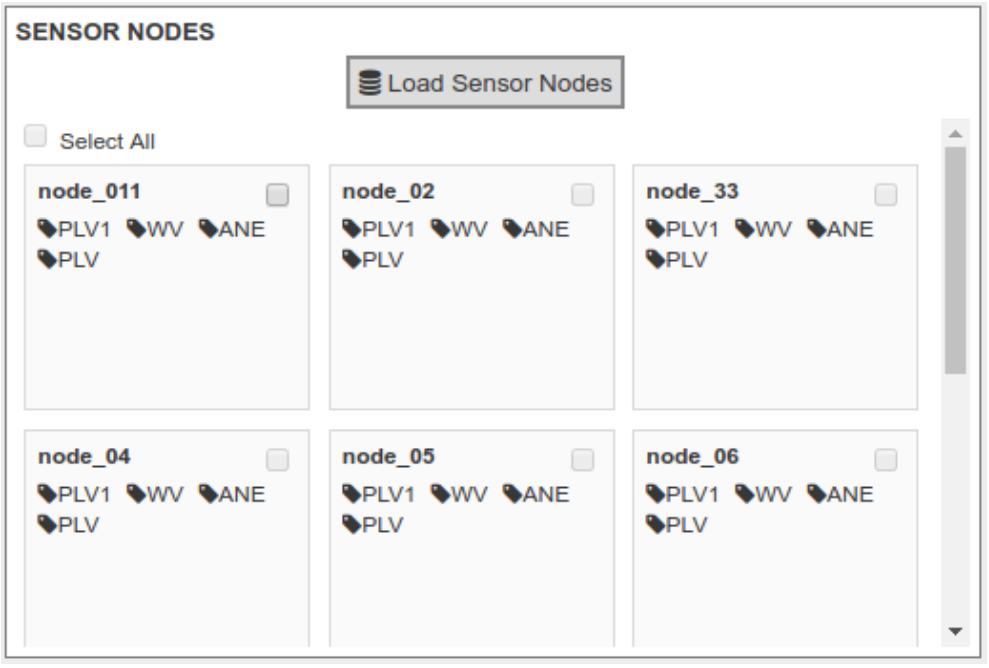


Figure : Plug & Sense! units panel

When you change any of the configuration parameters, the "Save" button turns red and a message will appear indicating that some changes are not saved. You cannot use the cloud connector with the new configuration until the changes are saved.

Figure : Configuration pending

Click the "Save" button for storing the configuration fields and start using the service. If you entered wrong parameters, a message will appear indicating that one or more of your parameters are incorrect.

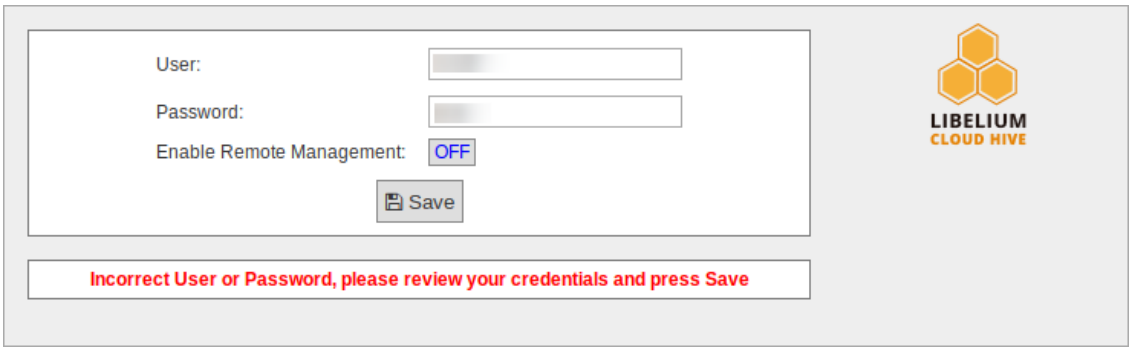


Figure : Incorrect User or Password error message

If you have not registered your Meshlium on the Hive, even though you have correctly configured your access data from the cloud, a message will appear indicating that you need to register your Meshlium, as shown in the following image.

Figure : "Meshlium not registered" error message

Controlling synchronization

Once the cloud connector is configured, the user can start the synchronization clicking the "Start" button. The system will start the synchronization of data from the devices that you previously selected. All data will be synchronized, not only the newly received frames, but also the past data stored in the database.

The status indicator displays the current state, saying "Running" or "Stopped".

Figure : Hive cloud connector is running

You can stop the synchronization anytime clicking on the "Stop" button.

Figure : Hive cloud connector is stopped

12.1.5. Telit

The Meshlium Cloud Connector for the Telit deviceWISE IoT Cloud platform is provided to connect Libelium Waspmote sensor devices to the Telit Cloud Platform.

Register in Telit

The basic steps required to connect Waspmote sensors to the cloud platform are simple and straight forward:

a) Using the Telit Management Portal at portal.telit.com:

- Create/Obtain a Telit IoT Portal Account on a Cloud Organization.
- Define 2 Thing Definitions:
  - Meshlium Cloud Connector.
  - Waspmote Sensor.
- Create an Application Token for the Meshlium Cloud Connector Definition.

b) Using the Meshlium Manager System web browser interface:

- Access the Telit deviceWISE TR-50 Cloud Platform Configuration Panel.
- Enter and Save the configuration details for your Meshlium including the Application Token created above.
- Start the deviceWISE TR-50 Sensor Processing Service.

Once your Telit IoT Cloud account has been configured with the new 'Thing Definitions' and 'Application' in support of the Meshlium Cloud Connector, you are ready to proceed with configuring your Meshlium gateway.

Configuration

Select the 'Telit deviceWISE Cloud Connector' service icon located on the Cloud Connector → Basic Cloud Partner panel. Configure the fields to provide the required configuration and control information on the 'Telit deviceWISE Cloud Connector' management panel as shown below.

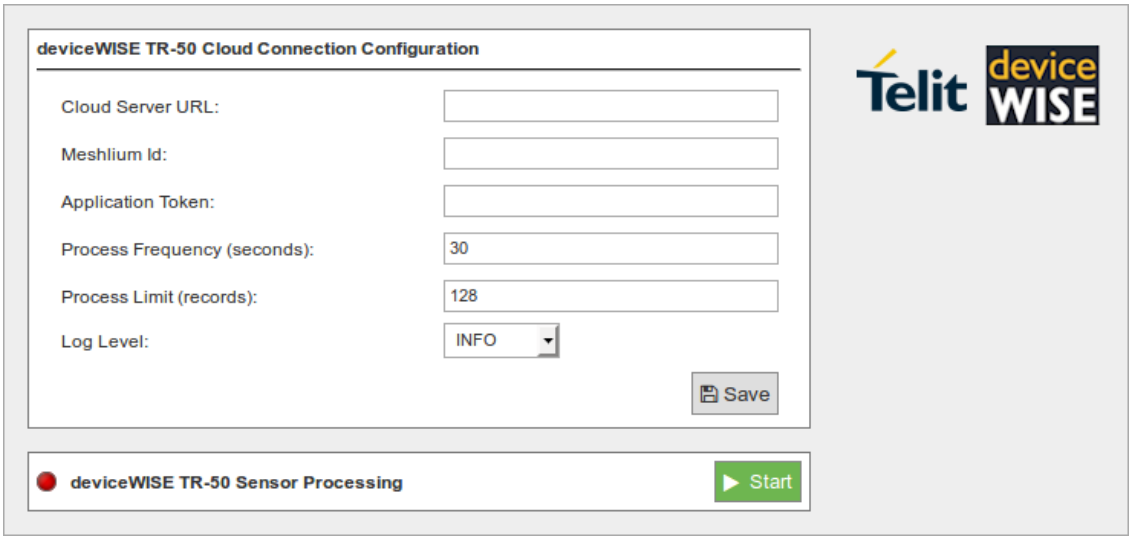


Figure : Telit configuration panel

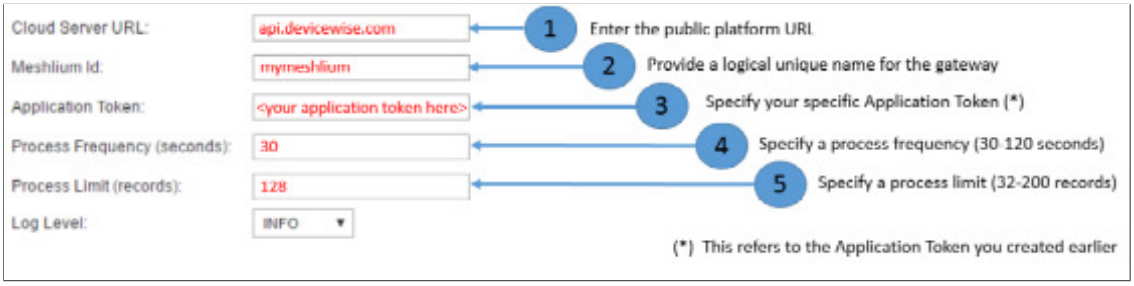


Figure : Telit configuration options

Where:

- 'Cloud Server URL' specifies the target Telit IoT Cloud Platform.
- 'Meshlium Id' indicates the unique name that you would like your Meshlium Gateway to be known as in the Telit IoT Cloud Platform.
- 'Application Token' indicates the unique secure token generated by the Telit IoT Cloud Platform for devices to be able to access your private cloud organization.
- 'Process Frequency' indicates how often the cloud connector should check for and process newly received data frames from the associated Waspmote/Edge devices. Valid values range from 30 to 120 seconds.
- 'Process Limit' specifies the maximum number of waiting records to process during the data frame processing cycle. Valid values range from 8 to 200.

Once all settings are provided, save the configuration settings by pressing the **Save** button.

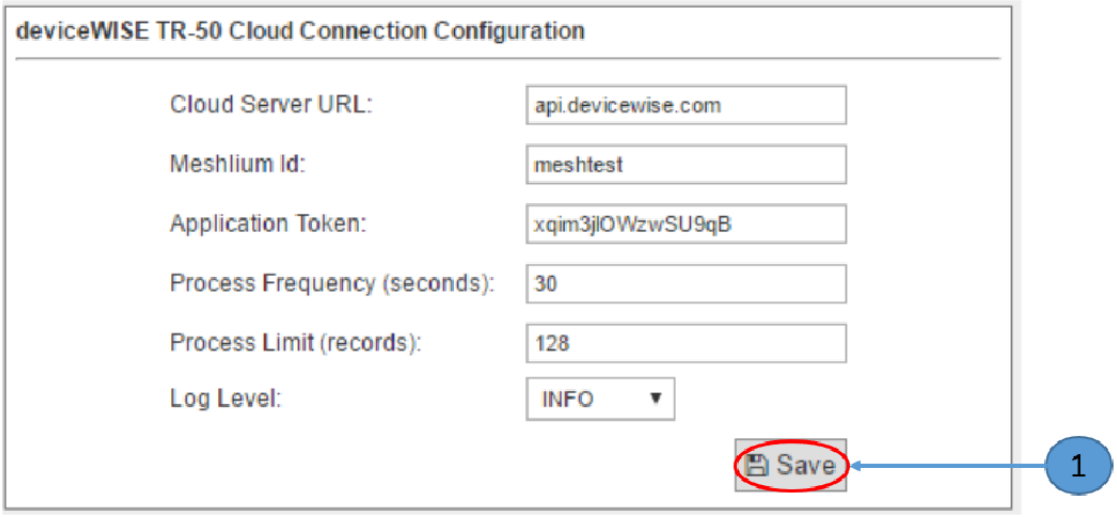


Figure : Save configuration button

In the event that a field has been left blank or a string has been entered into a numeric data field, the field entry frame will be highlighted and an error message will be displayed as shown in the screen images below.

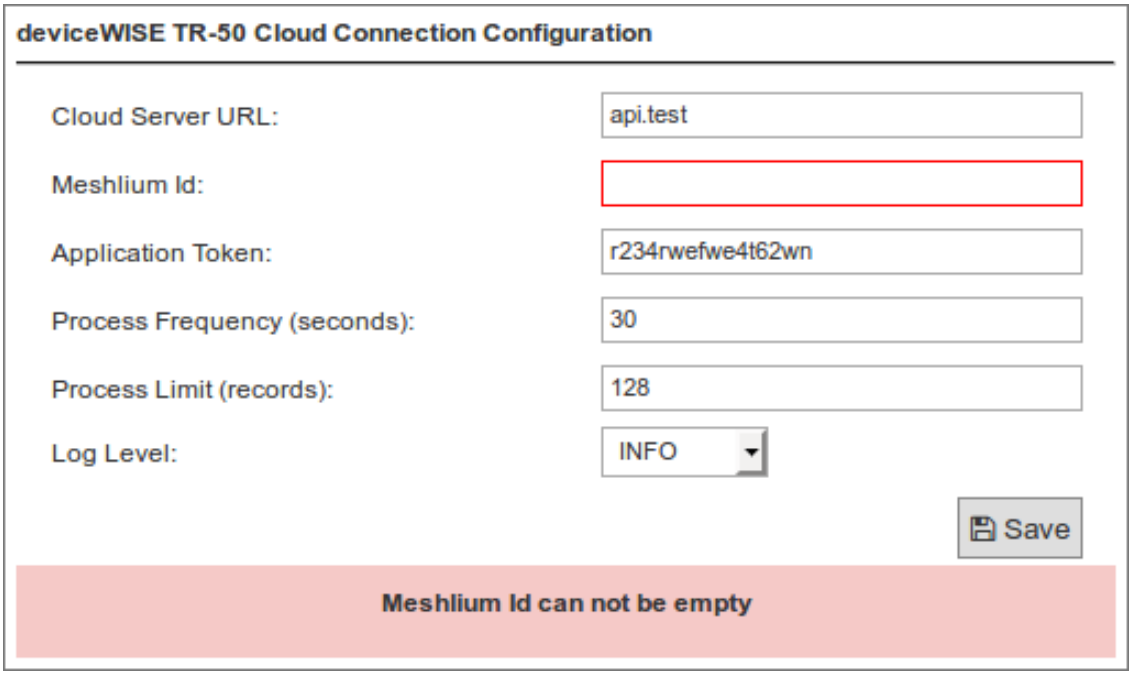


Figure : Save configuration error

At this point the Meshlium Cloud Connector is configured and ready to start.

Controlling synchronization

Once the Telit deviceWISE IoT Cloud Connector has been configured with the proper runtime parameters, it is ready for operation. To initiate the connection from the Meshlium gateway to the Telit IoT Cloud Platform and start the background service awaiting data frames from Waspmote sensor device nodes, press the Start button.

To initiate the connection from the Meshlium gateway to the Telit IoT Cloud Platform and start the background service awaiting data frames from Waspmote sensor device nodes, press the Start button.

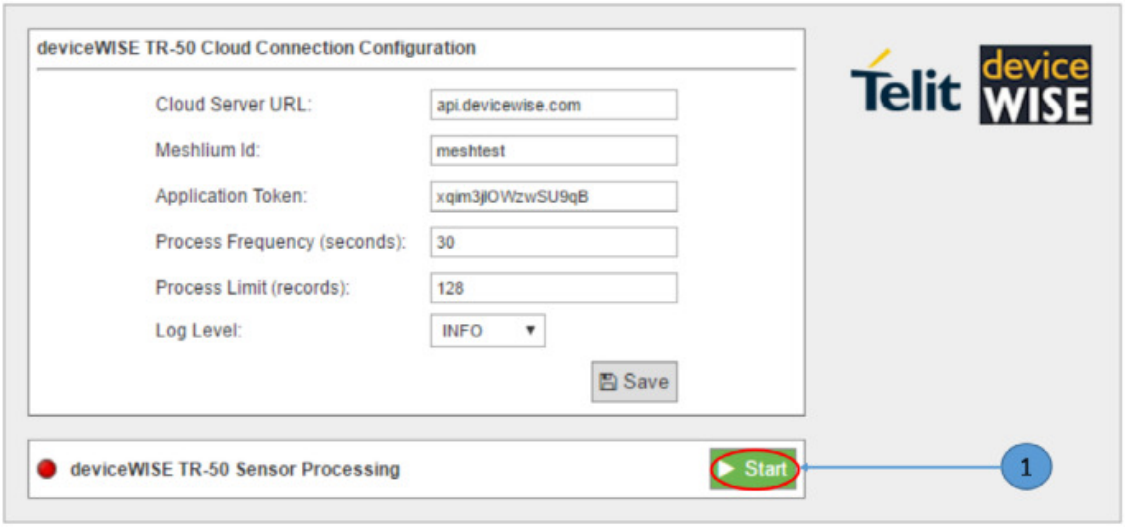


Figure : Telit Start button

To stop the background service from awaiting data frames from Waspmote sensor device nodes and terminate the connection from the Meshlium gateway to the Telit IoT Cloud Platform, press the Stop button.

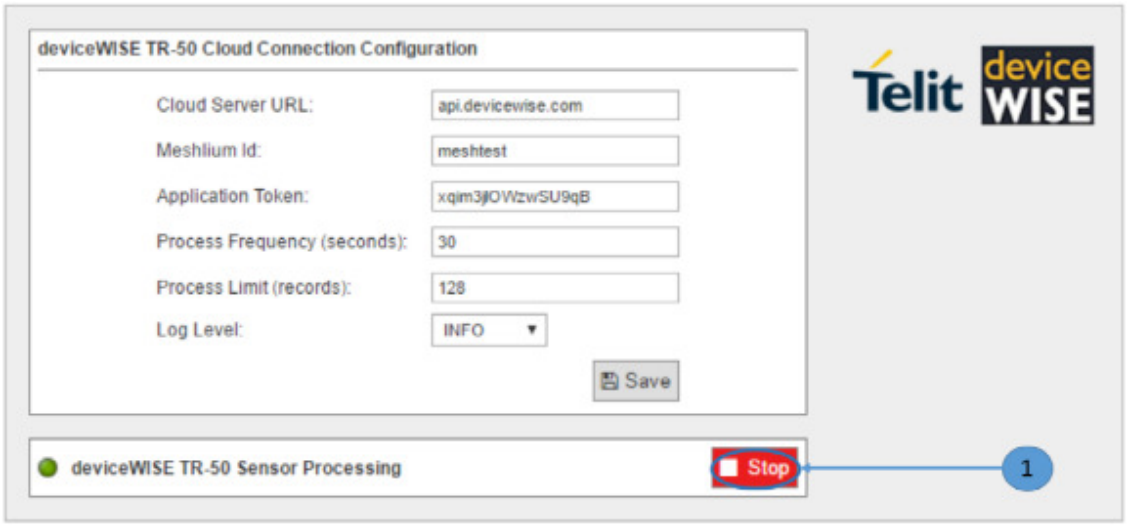


Figure : Telit Stop button

12.1.6. ThingWorx

ThingWorx is the first software platform designed to build and run the applications of the connected world. ThingWorx reduces the time, cost, and risk required to build innovative Machine-to-Machine (M2M) and Internet of Things applications by providing a complete application design, runtime, and intelligence environment. The ThingWorx platform includes flexible device connectivity options, rapid application development tools, scalable storage, and supports various deployment models.

More information: http://www.thingworx.com.

ThingWorx includes the following features:

- **ThingWorx Composer™**: an end-to-end application-modeling environment designed to help you easily build the unique applications of today's connected world. Composer makes it easy to model the Things, Business Logic, Visualization, Data Storage, Collaboration, and Security required for a connected application.
- **Codeless Mashup Builder**: a "drag and drop" Mashup Builder empowers developers and business users to rapidly create rich, interactive applications, real-time dashboards, collaborative workspaces, and mobile interfaces without the need for coding.
- **Execution and Storage Engine**: ThingWorx's event-driven execution engine and 3-Dimensional storage allows companies to make business sense of the massive amounts of data from their people, systems, and connected "Things" - making the data useful and actionable. It also features a data collection engine that provides unified, semantic storage for time-series, structured, and social data at rates 10X faster than traditional RDBs.
- **Search-based Intelligence**: ThingWorx SQUEAL™ (Search, Query, and Analysis) brings Search to the world of connected devices and distributed data. With SQUEAL's interactive search capabilities, users can correlate data that delivers answers to key business questions.

***Note**: if you need more information about these components, go to http://www.thingworx.com/platform/*

Configuration

Inside the "ThingWorx" plugin you can setup which Waspmotes in the system will be published in ThingWorx server.

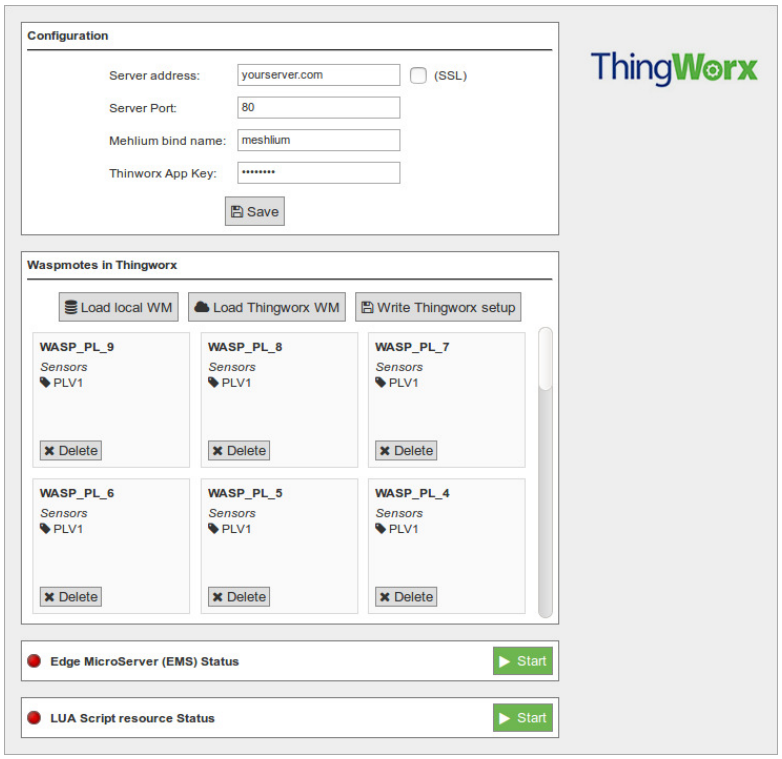


Figure : ThingWorx configuration

The parameters to setup are:

- **Server address**: The address of your ThingWorx server.
- **Server Port**: The port where your ThingWorx server is accessible.
- **Meshlium bind name**: The name of the Meshlium "thing" in ThingWorx. Meshlium thing is detected in ThingWorx but will not send any data.
- **ThingWorx App Key**: Security key to send data to your ThingWorx server.
- **SSL**: Enable this option if your ThingWorx server uses encrypted connection.

Click on the "Save" button to write this setup to the ThingWorx service.

The steps to setup Waspmotes to send to ThingWorx are:

- Click on the button "Load local WM". This will read Waspmotes that have data in the sensor database.

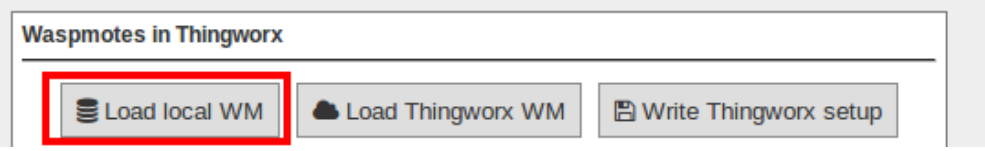


Figure : Getting Waspmotes from the system

- A panel with the devices is displayed, with a list of the sensors received from each Waspmote.

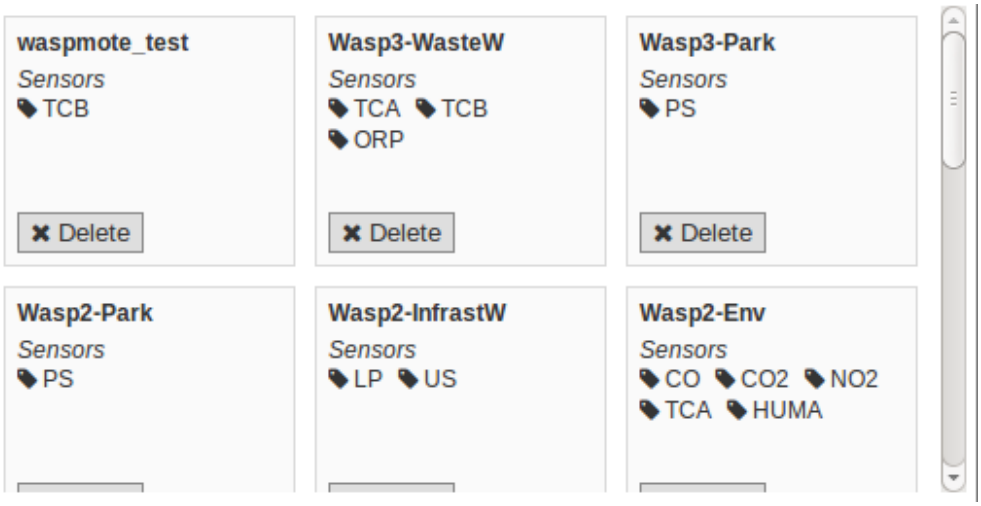


Figure : Waspmotes to be send to ThingWorx

- It is possible to delete a Waspmote from the list clicking on its "Delete" button. This device will not be published to the ThingWorx platform.

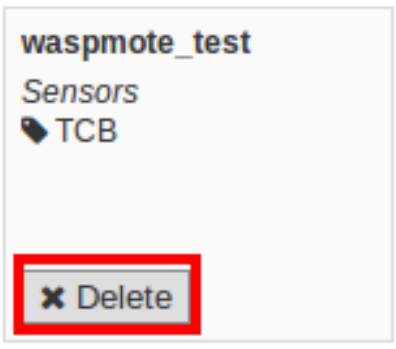


Figure : Delete a Waspmote from the list

- Once the list is correct, clicking on the button “Write ThingWorx setup” will push this setup to the ThingWorx EMS service.

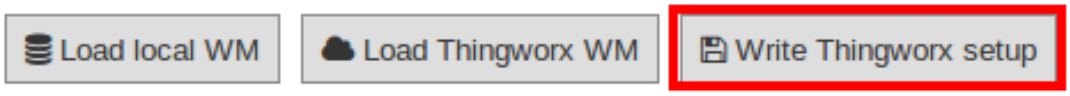


Figure : Write the setup to ThingWorx service

After restarting the EMS and LUA Script services, the setup will be applied and you will see your devices in your ThingWorx server.

If you click again on the “Get Waspmotes from DB” button, the plugin will read again the DB and display all the Waspmotes. If you do not write this changes to ThingWorx setup, this will not propagate to the EMS service.

You can recover the current ThingWorx EMS service setup by clicking on the “Load WM from ThingWorx”.

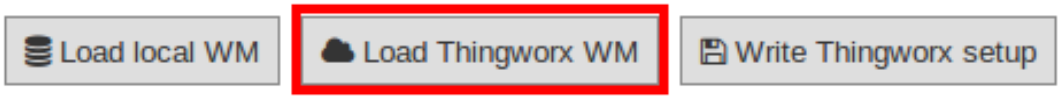


Figure : Write the setup to ThingWorx service

Using the Edge MicroServer (EMS)

ThingWorx has the ability to connect very quickly and easily to the physical world and structured data systems. ThingWorx supports a variety of communication protocols and system interfaces. Many are implemented directly on the ThingWorx Platform. However, for edge devices or data stores that need to connect to the platform using the Internet or through firewalls on an Intranet, ThingWorx provides an Edge MicroServer (EMS) solution that can be deployed where the data is, and allows secure, efficient communication back to the ThingWorx Platform. This section will concentrate on the EMS and the corresponding Edge Thing software components.

In order to send data to the ThingWorx environment, another component is needed: the LUA Script Resource service. This service needs to be running at the same time with Edge Micro Server to allow data acquisition.

To launch the Edge MicroServer (EMS) press Start button, and to stop it, press Stop button.

Figure : ThingWorx Edge MicroServer running

Figure : ThingWorx Edge MicroServer stopped

To launch the LUA Script resource press the Start button, and to stop it, press the Stop button.

Figure : ThingWorx Edge MicroServer running

Figure : ThingWorx Edge MicroServer stopped

12.2. Advanced Cloud Partners

12.2.1. Microsoft Azure Event Hubs

Azure is a cloud platform provided by Microsoft. This platform has a lot of services to reach communication between machines and devices.

This section focuses on Event Hubs, we can refer this technology as a way to send short messages via HTTP REST request. Event Hubs is part of Service Bus. Event Hubs implements a simple message consumer M2M technology.

For more information about Event Hubs, see the following link:
https://azure.microsoft.com/en-us/services/event-hubs/

Setup in Azure - Creating NameSpace

Before getting the parameters to connect to Event Hub, it is necessary to create a Service Bus Namespace (skip this section if you already have one).

Go to the Azure Event Hub Portal:
https://manage.windowsazure.com

Select the *Service Bus* menu. At the bottom of the Bus service manager screen you will see a *NEW* button with a plus image, click on it. A pop-up window will appear where you must select message type "EVENT HUB".

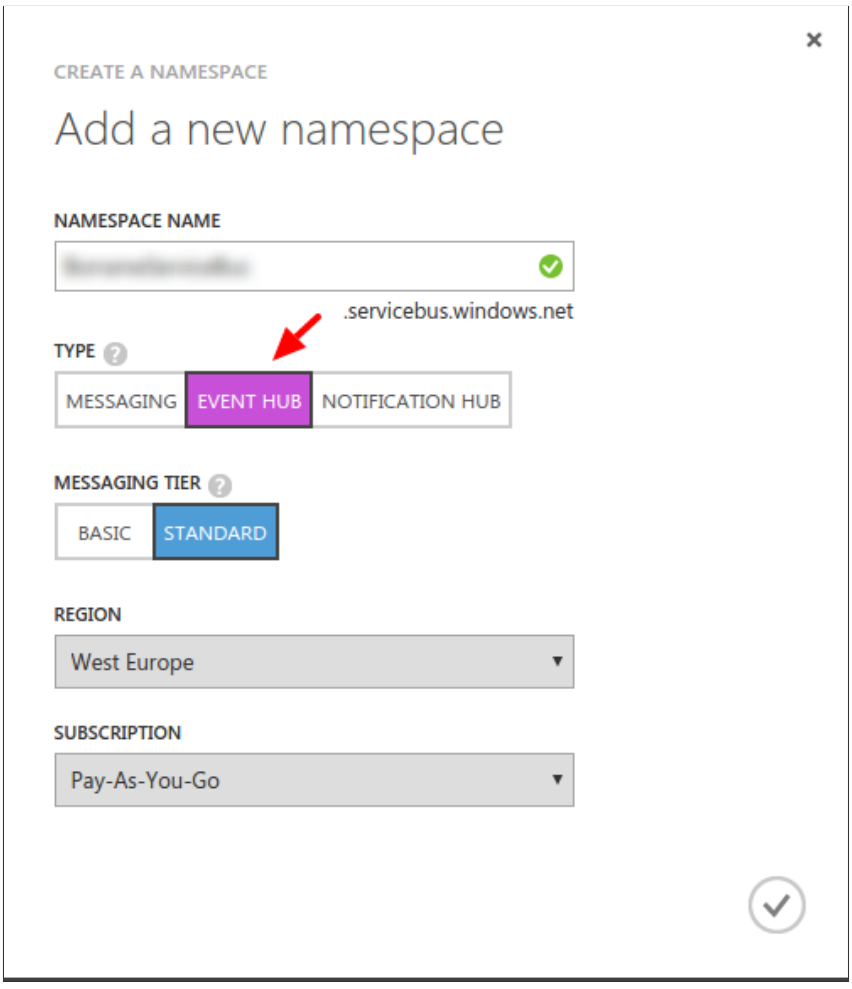


Figure : IBM Bluemix Create new Namespace

Setup in Azure - Creating an Event Hub

In this section we will create an Event Hub that will receive our data from Meshlium. After we dive into the Service Bus we have previously created, we can see a menu on the top of the screen, then we should choose “Event Hubs” and “Create a New Event Hub”:

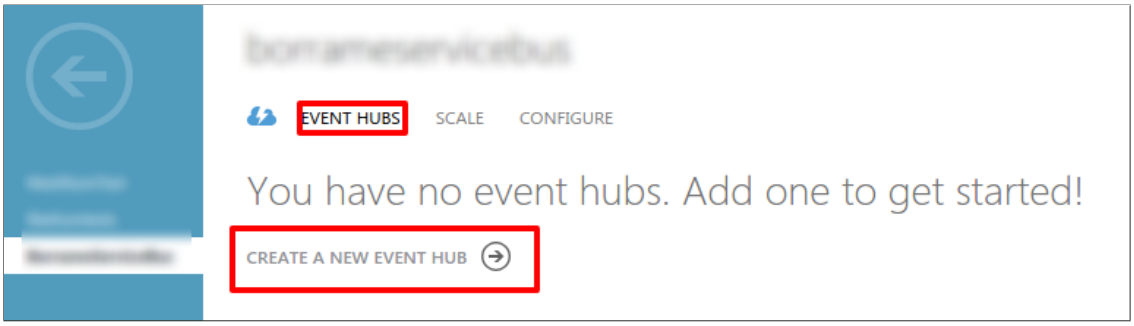


Figure : Create New Event Hub

After clicking on this menu, a new screen will appear. At this point we can create a new event hub clicking on the bottom left icon labeled as New:

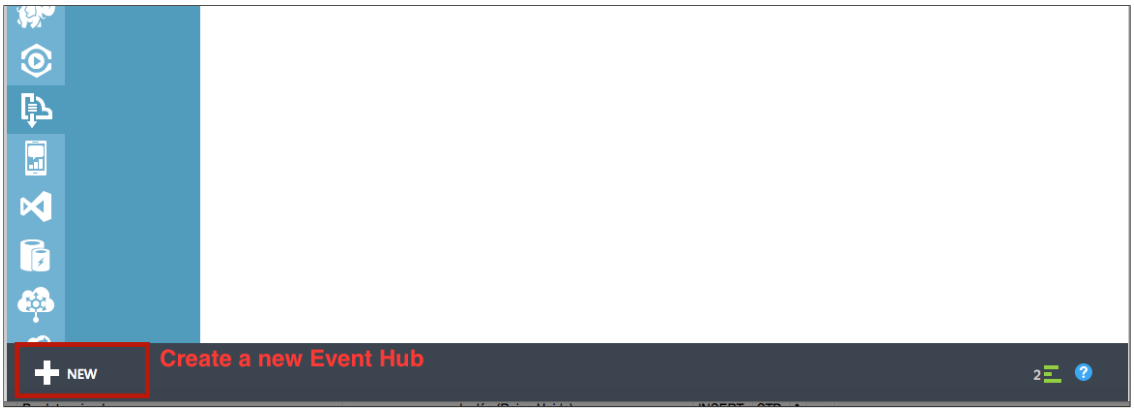


Figure : Create a New Event Hub

After click this button, a pop-up window raises above and you are now able to create an event hub, we are going to choose “quick create” option to make this step easier:

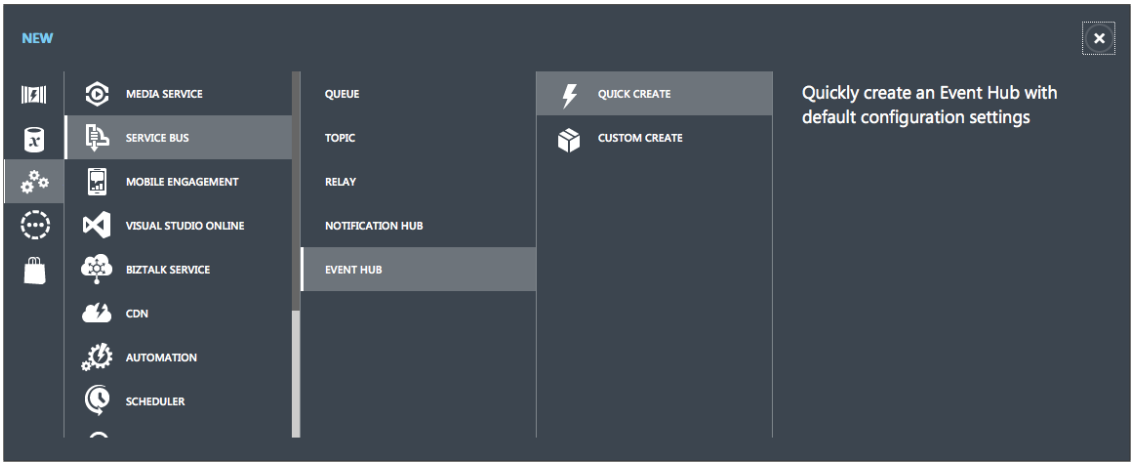


Figure : Quick Create

Type your event hub name and click “Create a new event hub” button to finish the configuration process.

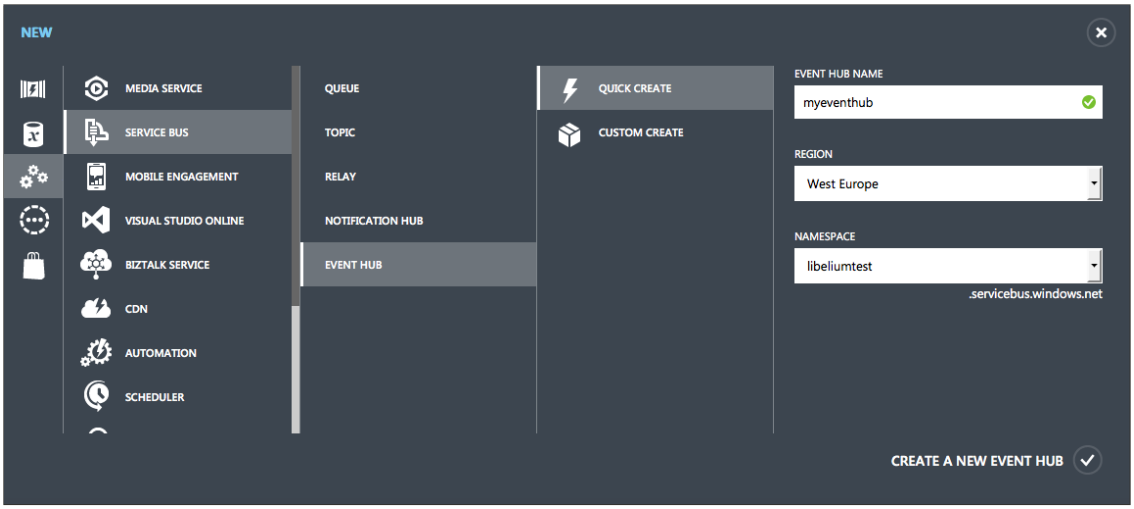


Figure : Type the new Event Hub name

You have created an Event hub for one day data retention, which means that your data will be kept one day.This method sets a partition section with value ‘4’, which means the number of partitions the Event Hub may have.

Setup in Azure - Setting up shared access in Event Hub

We set up a shared directive to send data with custom credentials.Once we entered on event hub information (by clicking on event hub), these credentials can be set up in the configuration section, this menu is on top of the screen:

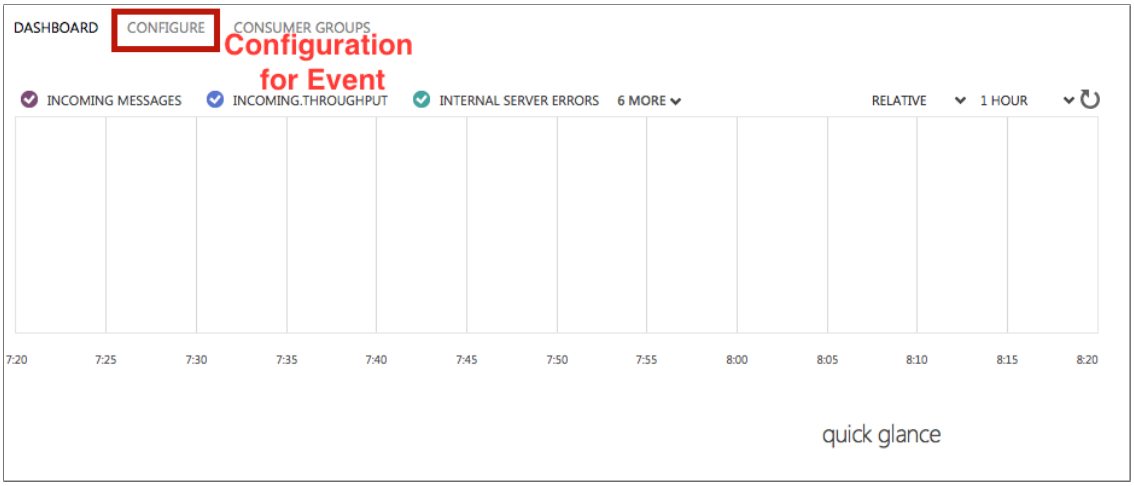


Figure : Configure the Event

Click on the “configure” option and a new screen will be displayed. Here you can configure message retention, event hub state, partition count and shared access policies. This last point (shared access policies) manages credentials to send and listen messages (or both action), we will create a new credential to send messages. On “shared access policies”, type a name for your key, and in the permissions drop-down menu, select “Manage” permission. Then press “save” on the bottom of the screen.

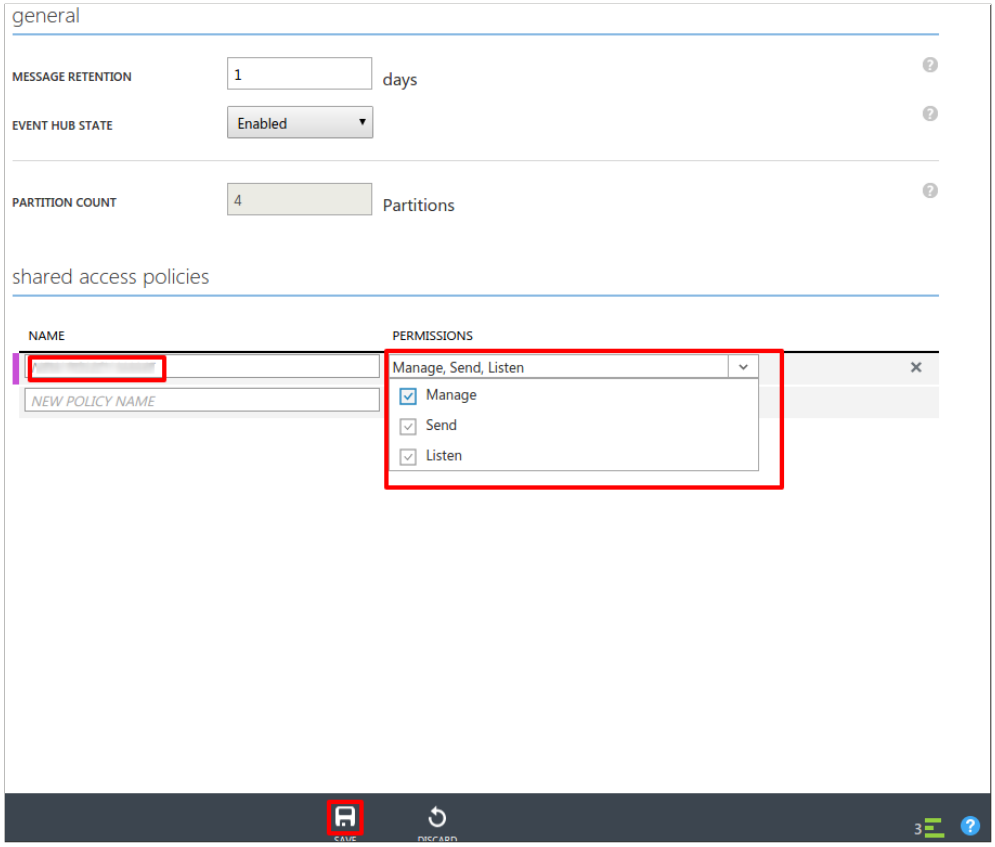


Figure : Configure the Event Permissions

Finally, we will copy the information needed to connect the Event Hub connector. In order to do that, go to the "Dashboard" of the Event Hub and select "View Connection String".

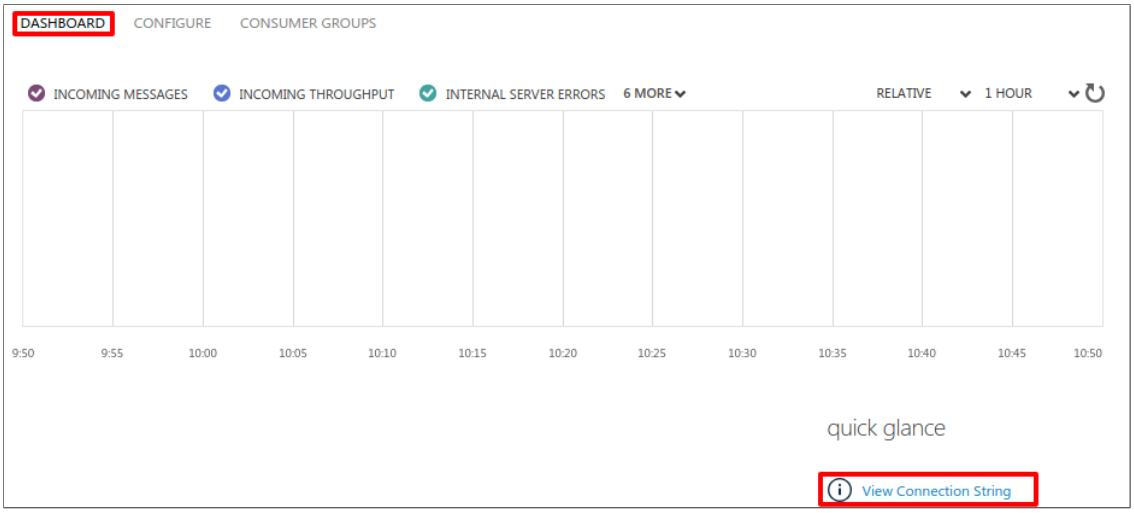


Figure : Event Hub Dashboard

Copy the “Connection String” that appears in the screen.

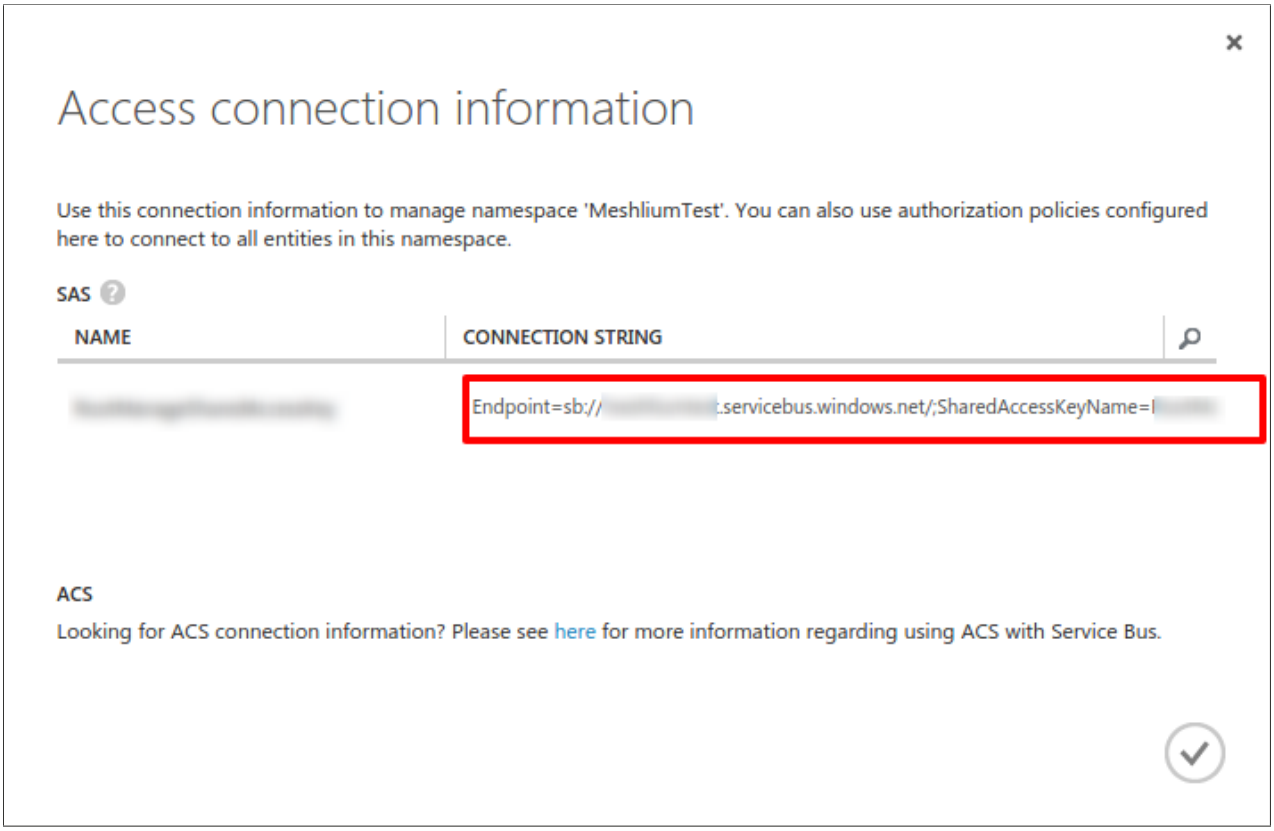


Figure : Connection String

We will extract the information needed to connect the Meshlium from the “Connection String”. You have to copy the **NAMESPACE**, the **DIRECTIVE_NAME**, the **DIRECTIVE_KEY** and the **EVENTHUB_NAME** from the string as the following example:

Namespace=Endpoint=sb://**NAMESPACE**.servicebus.windows.net/;SharedAccessKeyName=**DIRECTIVE_NAME**;SharedAccessKey=**DIRECTIVE_KEY**;EntityPath=**EVENTHUB_NAME**

Note that the namespace is only a part of the string Endpoint, it does NOT include the “sb://” neither the “.servicebus.windows.net/” part.

Configuration

As result of previous steps, we should have a namespace, a directive name, a directive key and an event hub name. These are the main properties we should set in the Manager System to configure an Event Hub connection.

Now we can access the Meshlium Manager and fulfill the Azure Event Hub fields with the previously obtained configuration.

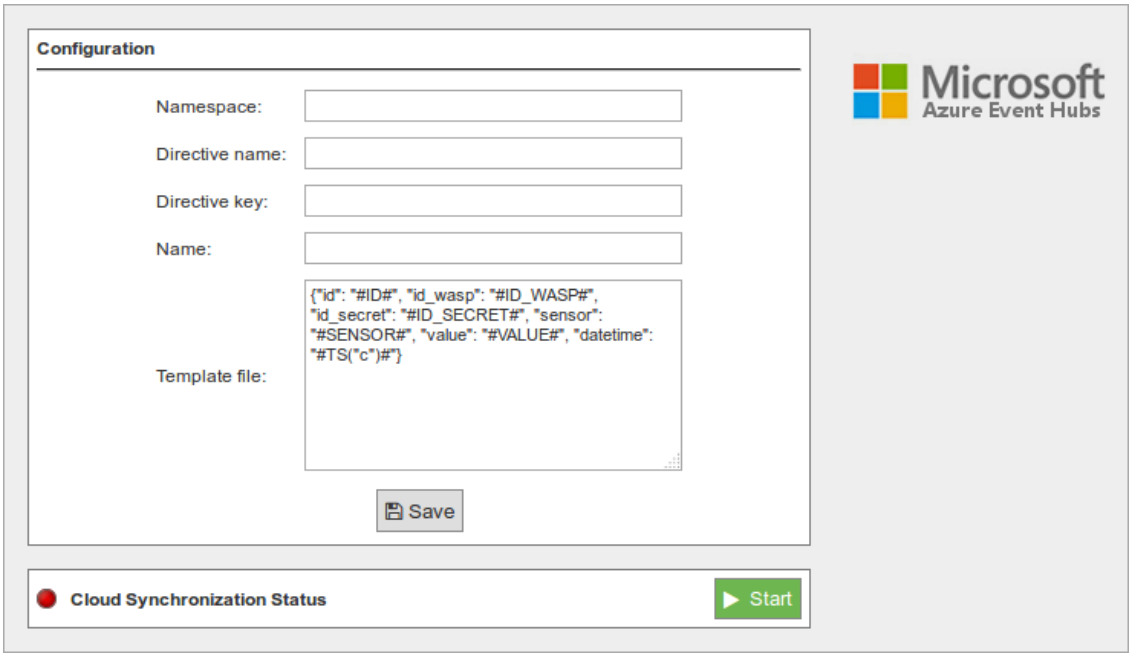


Figure : Configuring Azure Event Hubs in Meshlium

- **Namespace**: Name of the space created in the Azure service cloud.
- **Directive name**: Name of the directive created in Azure.
- **Directive key**: Key of the directive associated to the previous name.
- **Name**: Name of the Event Hub established in Azure.
- **Template file**: Users can define their own data structure using these wild-cards:
  - #ID# : Unique identifier for data.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.

#TS("c")#: Date with custom format. The parameter passed in this wild-card corresponds to the same ones you can use in PHP date function (see format parameters in http://php.net/manual/es/function.date.php#refsect1-function.date-parameters).

Controlling synchronization

Once you have saved the configuration, you can start sending your data via Event Hub to your Azure Cloud by pressing the “Start” button. You will notice about it because the screen shows a spinning wheel when the process starts and displays a “running” status.

Figure : Azure Event Hubs synchronization service is running

If you want to stop this process just press the “Stop” button. You can start/stop this process whenever you want.

Figure : Azure Event Hubs synchronization service is stopped

12.2.2. Microsoft Azure IoT Hub

Azure IoT Hub is a fully managed service that enables reliable and secure bi-directional communications between millions of Internet of Things (IoT) devices and a solution back end. One of the biggest challenges that IoT projects face is how to reliably and securely connect devices to the solution back end. To address this challenge, IoT Hub:

- Offers reliable device-to-cloud and cloud-to-device hyper-scale messaging.
- Enables secure communications using per-device security credentials and access control.
- Includes the most popular communication protocols.

More information: https://www.microsoft.com/en-us/cloud-platform/internet-of-things-azure-iot-suite.

With this plugin, Meshlium can send messages to your cloud back-end.

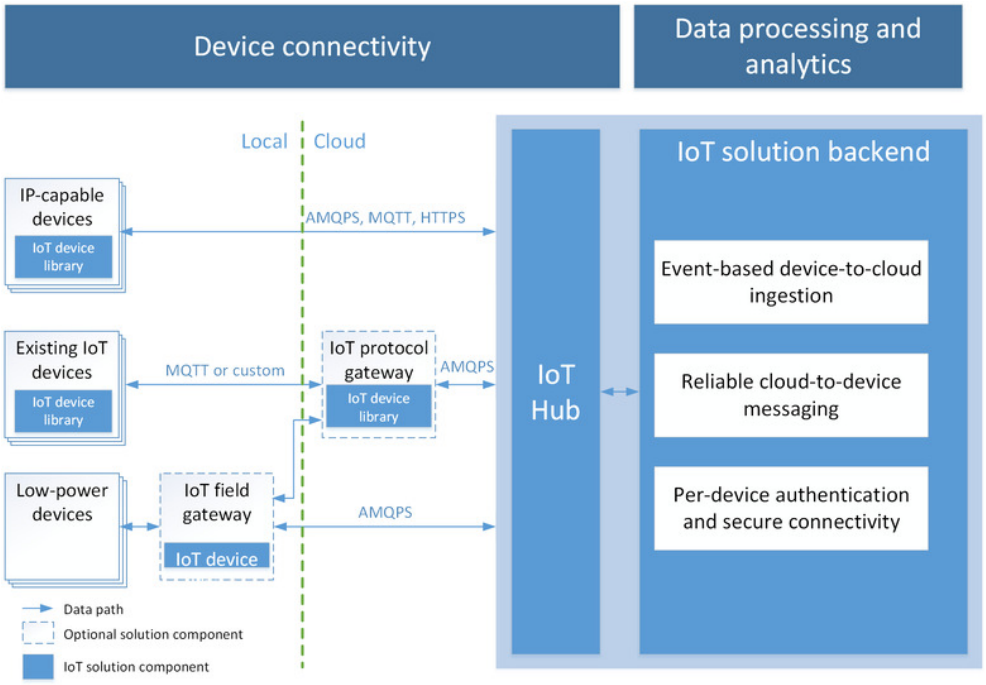


Figure : Azure IoT Hub plugin

Register Meshlium in Azure Portal

To register Meshlium in Azure Portal, you have to follow the guide “Get started with Azure IoT Hub for Java”: https://azure.microsoft.com/en-us/documentation/articles/iot-hub-java-java-getstarted/.

The guide explains how to create an IoT Hub and a device entity. It is important to annotate the connection string generated after creating the device entity. You will need this parameter later for the Meshlium configuration.

In the Microsoft Azure Portal, go to IoT Hub menu and select:

Devices > myCreatedDevice > Shared access policies > iothubowner > Connection string - primary key

You have to annotate the value of this field.

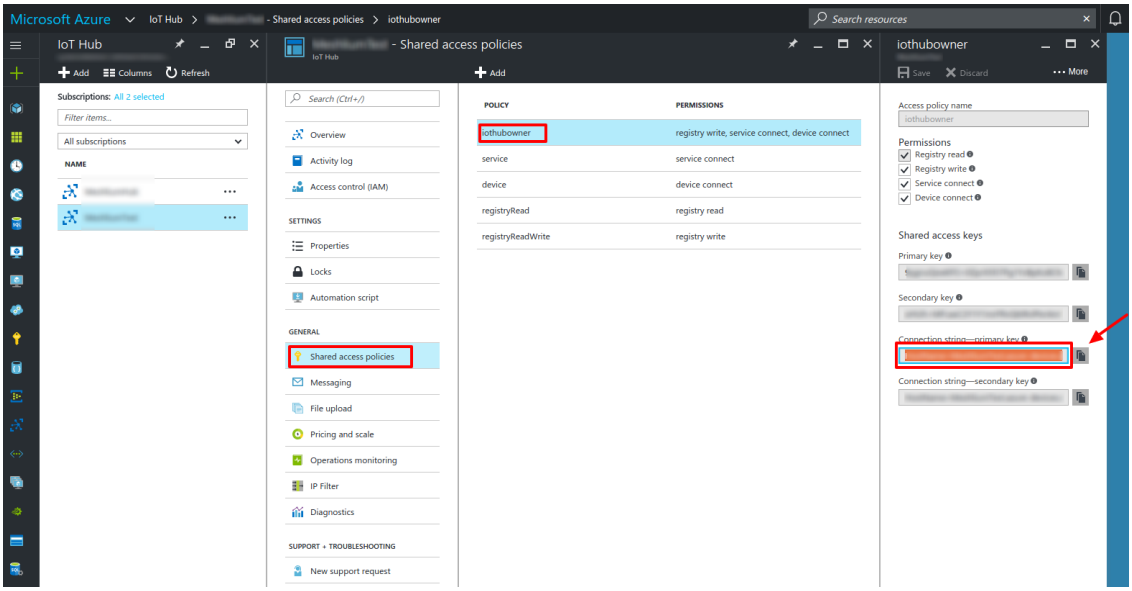


Figure : Annotate the value of the field

Configuration

You will use the previously obtained "connection string" from the Azure portal to certificate your Meshlium as a valid sender of messages.

Microsoft Azure IoT Hub plugin is located in:

Cloud Connector → Premium Cloud Partner → Azure IoT Cloud

In the Configuration panel, the user can set:

- **Connection String**: Connection string previously copied.
- **Number Requests**: Number of requests to send per iteration.
- **Sync Interval**: Time duration in seconds between synchronizing data batches.
- **Protocol**: Choose the protocol to communicate with Azure IoT Hub. Valid protocols are: MQTT (by default), AMQPS and HTTPS.
- **Log Level**: Generate log messages. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG, REPORT. Default is OFF.

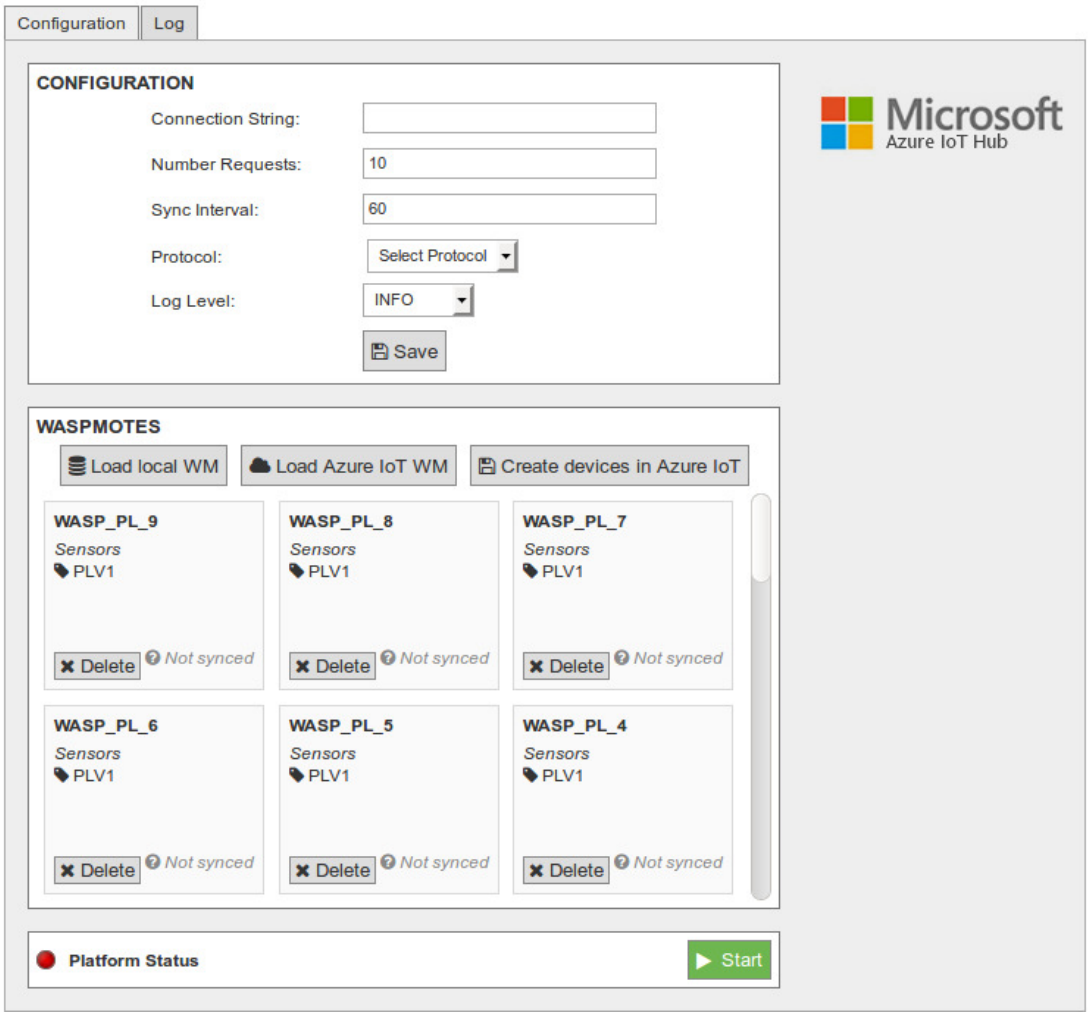


Figure : Azure IoT Hub configuration panel

Controlling synchronization

Once configured the server/broker, the user can launch the Meshlium Microsoft Azure IoT Hub script (Start button). The program will send test messages to the Azure IoT Hub platform via the selected protocol. The status indicator displays the current state, saying “Running” or “Stopped”.

Figure : Azure IoT Hub sender is running

You can stop the Azure IoT Hub program anytime clicking on the “Stop” button.

Figure : Azure IoT sender is stopped

12.2.3. Ensura

The Ensura Command & Control platform (http://www.ensuracc.com/) manages live and recorded video, audio, and data flows, and shows simultaneous live and recorded views in a single window. The system provides server-side and sensor-side Video Content Analytics for all video channels, displaying triggered alerts from any 3rd party end-element.

The system supports biometric and face recognition modules, ANPR systems, Access Control solutions, Security and Failsafe platforms. Open-ended and integration-agnostic, additional systems can be rapidly and effectively integrated into an Ensura deployment.

Ensura supports event-driven triggering of activities throughout the system, and supports definition of predefined scenarios for management of routine and emergency events.

Ensura enables users to manage multiple remote sites using a single client interface, displaying real-time and archived data of all monitored sites concurrently.

Ensura uses its Sensor Server to receive data from the Meshlium devices over HTTP requests.

Configuration

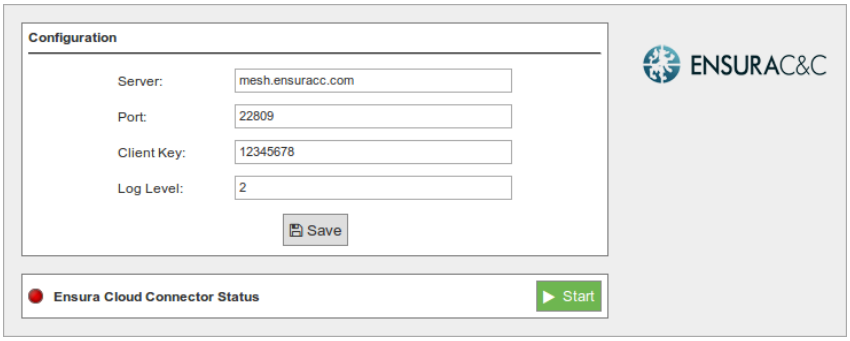


Figure : Configuring Ensura in Meshlium

- **Server**: The IP address or the URL of your Ensura Sensor Server.
- **Port**: The port number on which the Ensura Sensor Server is listening for connections.
- **Client Key**: The identifier that the Ensura Sensor Server is configured to accept information from.
- **Log Level**: This is the log level used by the cloud connector within the Meshlium device. Should be 0, except for debugging purposes.

This data will be provided by the administrators of the Ensura platform.

Controlling synchronization

The synchronization will be done for all data that has not been synchronized in the Sensor Parser table each time. You can start and stop the data synchronization to the Ensura Sensor Server. In the interface you can see an indicator of whether the status service is running or not. If you click on "Start", the synchronization will begin.

Figure : Ensura cloud connector status "Stopped"

You can stop the synchronization at any moment clicking on the "Stop" button.

Figure : Ensura cloud connector status "Running"

12.2.4. Infiswift

For more details on the platform, please visit https://www.infiswift.com and review swiftLab documentation.

Configuration

In order to publish data from the Meshlium Gateway to the infiswift broker, you will need to login to the Meshlium Manager for configuration. Navigate to Cloud Connector → Basic Cloud Partner → infiswift . You will need to complete the following fields in the infiswift Configuration:

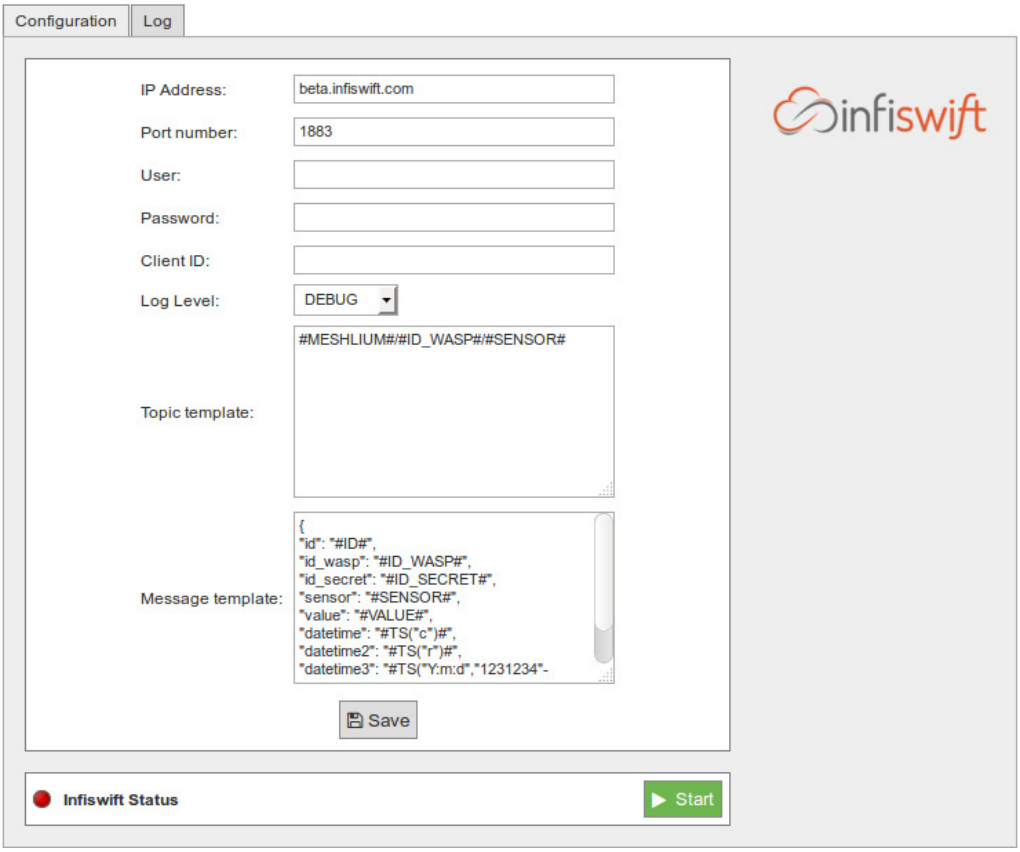


Figure : Configuration panel

With this plugin, the Waspmote sensor data can be directly integrated with an infiswift MQTT broker.

Pull required configuration information from infiswift portal using email id created in previous steps.

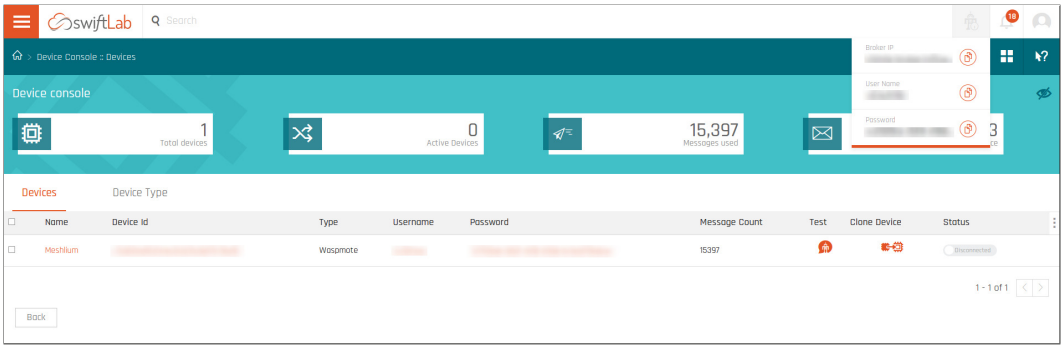


Figure : infiswift portal

- **IP Address**: This is the broker IP address (beta.infiswift.com) of the Gateway you want to connect to.
- **Port Number**: This is the port you opened upon configuration of Meshlium plugin.
- **User**: This is the username of the device created and to be supplied upon configuration of Meshlium plugin.
- **Password**: This is the password of the device created and to be supplied upon configuration of your Meshlium plugin.
- **Client ID**: This is the device ID from DEVICE CONSOLE page and to be supplied upon configuration of Meshlium plugin.
- **Topic Template**: This is the topic you supplied upon configuration of Meshlium plugin.
- **Message Template**: This is the message template of the data you want to send. Meshlium generates a default template, but you can provide your own if it is aligned with the Meshlium specifications.

Meshlium will start to listen for and ingest data coming from your device, and then forward it to infiswift's cloud.

Controlling synchronization

Once infiswift's swiftPV server/broker is configured, the user can launch the Meshlium infiswift script ("Start" button). The program will search for the received frames on the local database, and will send them to the swiftLab platform via the MQTT protocol. The status indicator displays the current state, saying "Running" or "Stopped".

Figure : swiftLab IoT sender is running

You can stop the infiswift IoT program anytime by clicking on the "Stop" button on the bottom of the page.

Figure : swiftLab IoT sender is stopped

12.2.5. Ubicamovil

Ubicamovil IoT is an IoT Web Interface to connect and manage your devices and allows you to create your KPIs based on the data transmitted by your IoT devices or things, even with Meshlium and Waspmote.

More information can be found at http://cellforce.mx/Publicacion?noticia=1436

Configuration

The Ubicamovil IoT plugin is configured with the following parameters:

- **Host**: iot.ubicamovil.com
- **Port**: 3095
- **ClientID**: Assigned by Ubicamovil through the web service using the username.

After uploading these parameters, save the configuration by clicking the “Save” button.

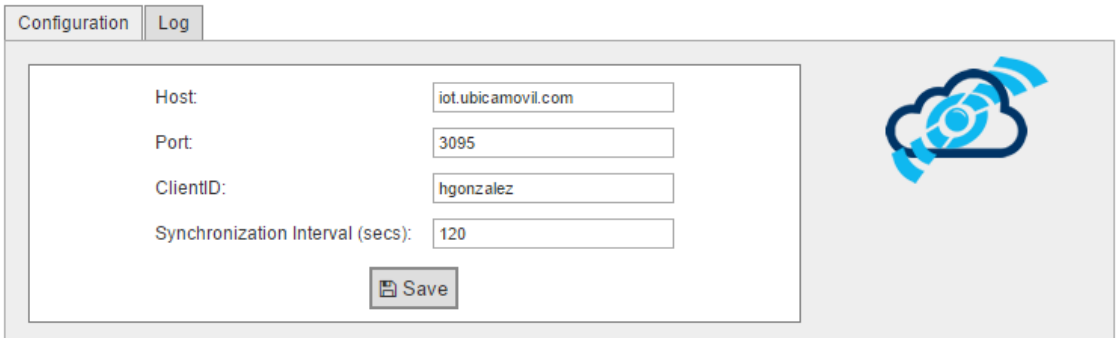


Figure : Ubicamovil panel

Controlling synchronization

By pressing the “Start” button, the Cloud Connector would start to send the data to the database of the webservice in Ubicamovil.

You can see the information also on the “Log” tab.

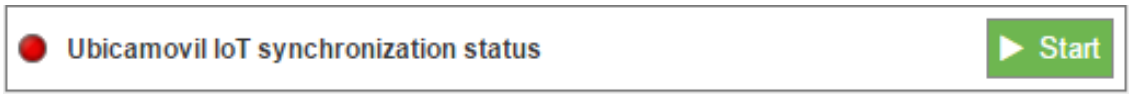


Figure : Ubicamovil sender is stopped

12.3. Basic Cloud Partners

12.3.1. Alibaba Cloud

Alibaba Cloud is a cloud platform provided by Alibaba. This platform provides a wide variety of cloud services such as web hosting, elastic computing, big data analytics or database services.

In this case, the data is stored in a MongoDB database. More information about this service can be found through the following link: www.alibabacloud.com/product/apsaradb-for-mongodb.

Register Meshlium in Alibaba Cloud

Neither MongoDB nor Alibaba Cloud requires to register your Meshlium devices; the data will be dumped into the database regardless of the device who makes this operation.

Configuration

In order to synchronize the local data of your Meshlium device with Alibaba Cloud, you should create a MongoDB database on Alibaba Cloud. Once created, you will be provided with a **host**, **port**, **user** and **password**, which allows you to access to your MongoDB shell.

Then, you will need to create a MongoDB database and a collection inside this database. This can be done in different ways, the easiest one is to connect to your MongoDB shell using the credentials obtained in the previous step (https://docs.mongodb.com/tutorials/connect-to-mongodb-shell/) and then create the **database** and the **collection** (https://docs.mongodb.com/manual/core/databases-and-collections/). Notice that it does not matter how you create or configure your collection or database, the connector will only attempt to dump the data to the given database and collection without any further check-ins.

In the Configuration panel, the user can set:

- **Host**: MongoDB Alibaba Cloud host obtained in the previous step.
- **Port**: MongoDB Alibaba Cloud port obtained in the previous step.
- **User**: MongoDB Alibaba Cloud user obtained in the previous step.
- **Password**: MongoDB Alibaba Cloud password associated to the user obtained in the previous step.
- **Database**: MongoDB Alibaba Cloud database name obtained in the previous step.
- **Collection**: MongoDB Alibaba Cloud collection name obtained in the previous step.
- **Requests**: Number of requests to be sent from Meshlium to the cloud in each iteration.
- **Log level**: Detail level in log messages. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG and REPORT. Default is INFO.

Finally, click on the “Save” button for storing the configuration fields.

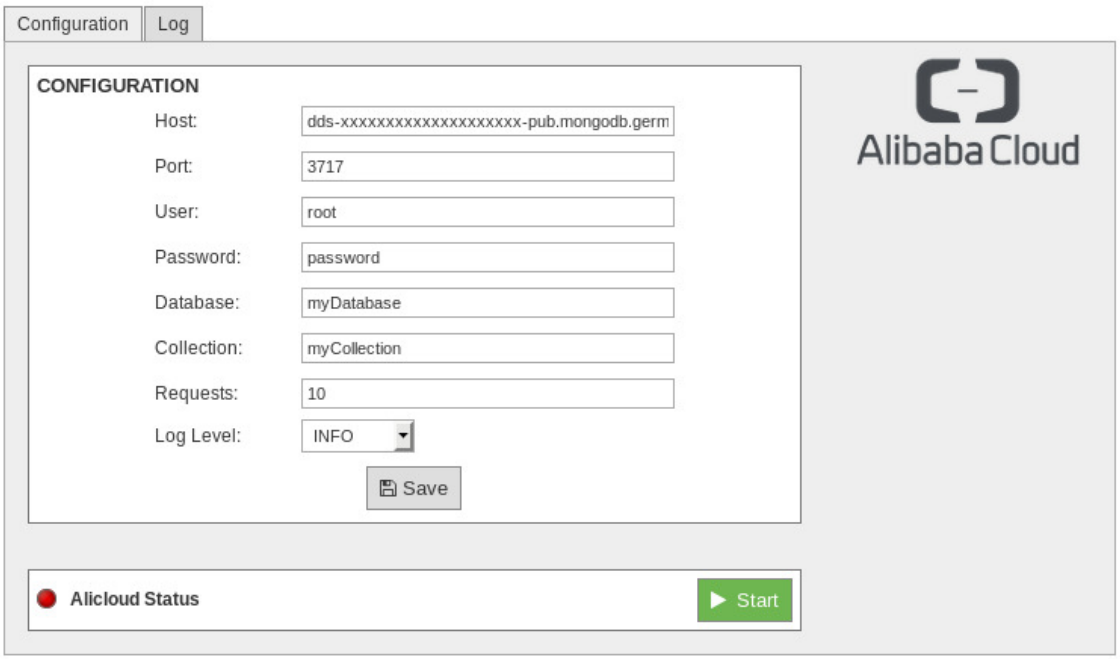


Figure : Alibaba Cloud configuration panel

Controlling synchronization

Once configured the connector, the user can launch the Meshlium Dummy script (Start button). The program will search for the received frames on the local database, and will send them to Alibaba Cloud. The status indicator displays the current state, saying “Running” or “Stopped”.

Figure : Alibaba Cloud sender is running

You can stop the Alibaba program anytime clicking on the “Stop” button.

Figure : Alibaba Cloud sender is stopped

12.3.2. Amazon IoT

Amazon Web Services IoT enables secure, bi-directional communication between Internet-connected things (such as sensors, actuators, embedded devices, or smart appliances) and the AWS cloud over MQTT and HTTP.

More information: http://aws.amazon.com/iot/.

With this plugin, Waspmote sensor data can be directly integrated with Amazon AWS IoT broker.

Figure : Amazon IoT plugin

Register Meshlium in Amazon IoT

To register Meshlium in Amazon IoT, you have to create a "thing" in your Amazon AWS IoT dashboard, attach a security certificate and policy statement and copy the parameters to the plugin. Follow these steps to register your Meshlium:

1. Select AWS IoT in the Amazon Dashboard.

Figure : Select AWS IoT

2. Create a "Thing".

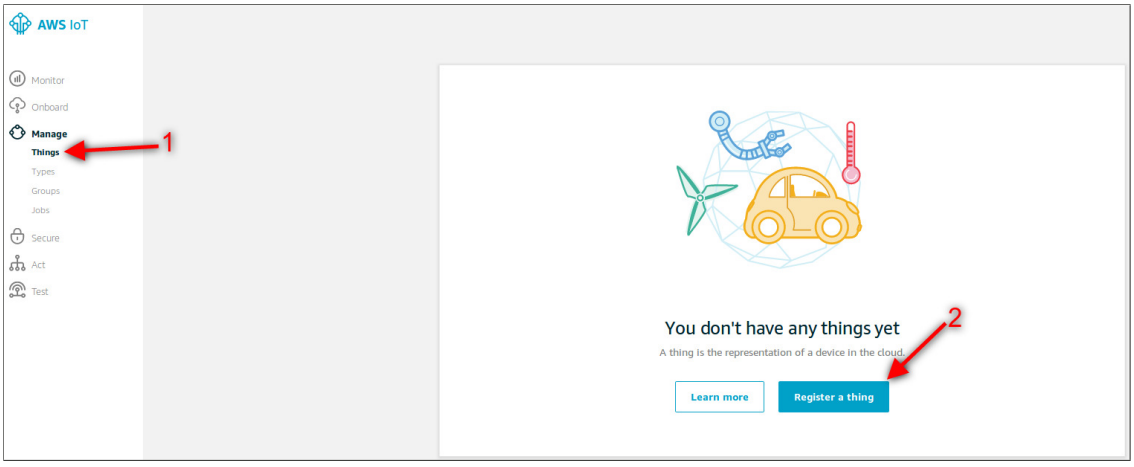


Figure : Create a thing

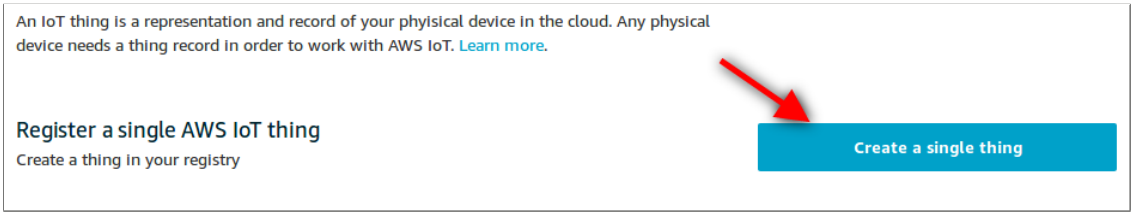


Figure : Push the button to create a thing

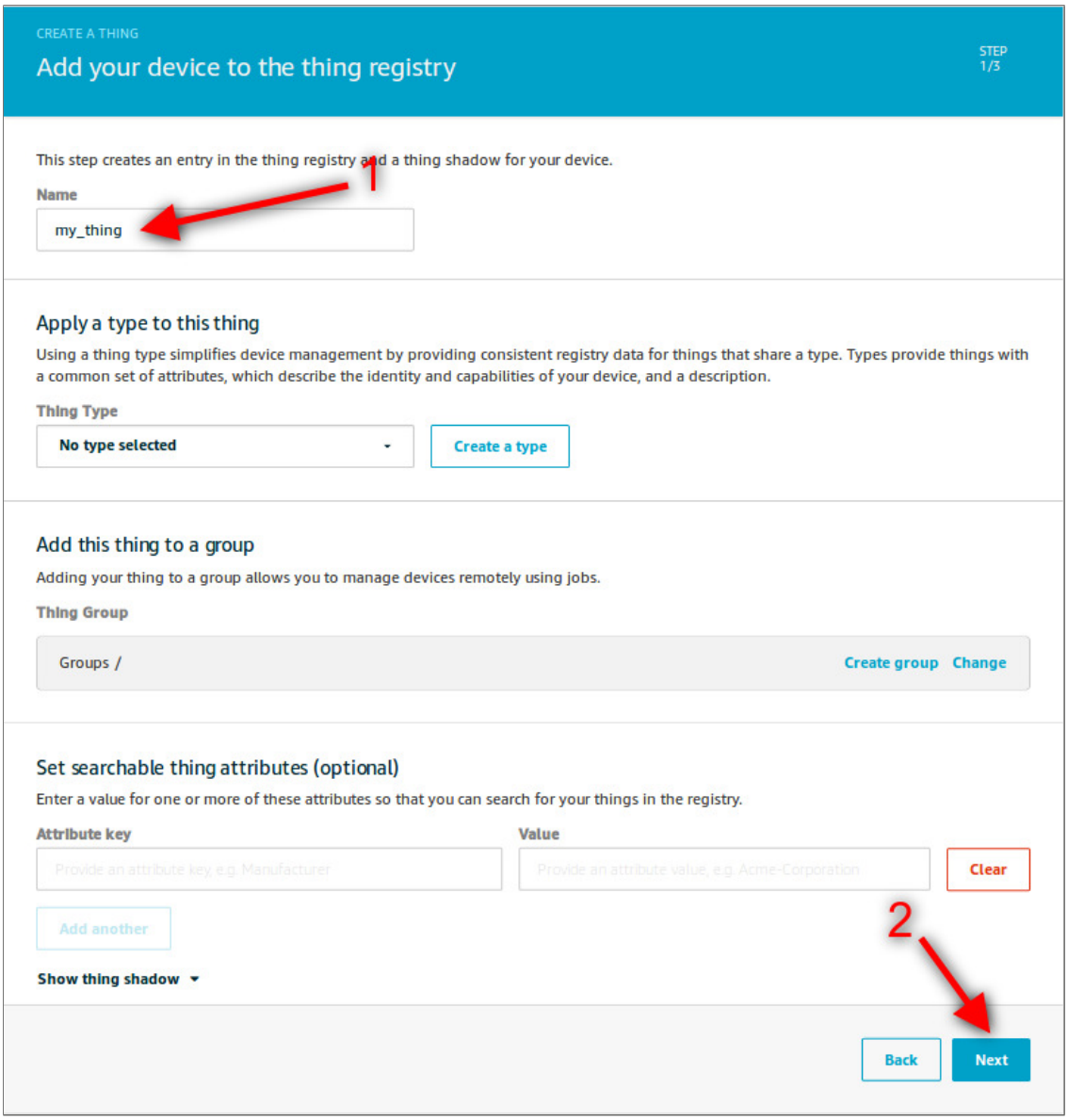


Figure : Add your device to the thing registry

3. Create a security certificate and download the files for later use.

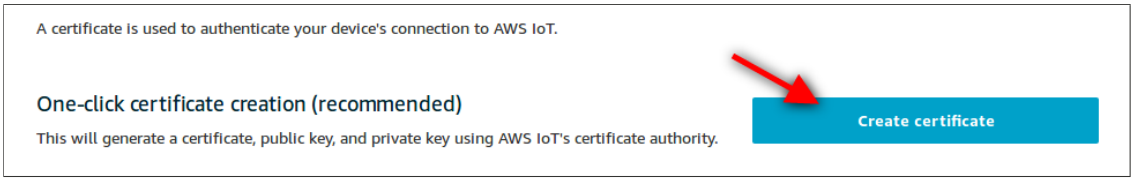


Figure : Select Create certificate

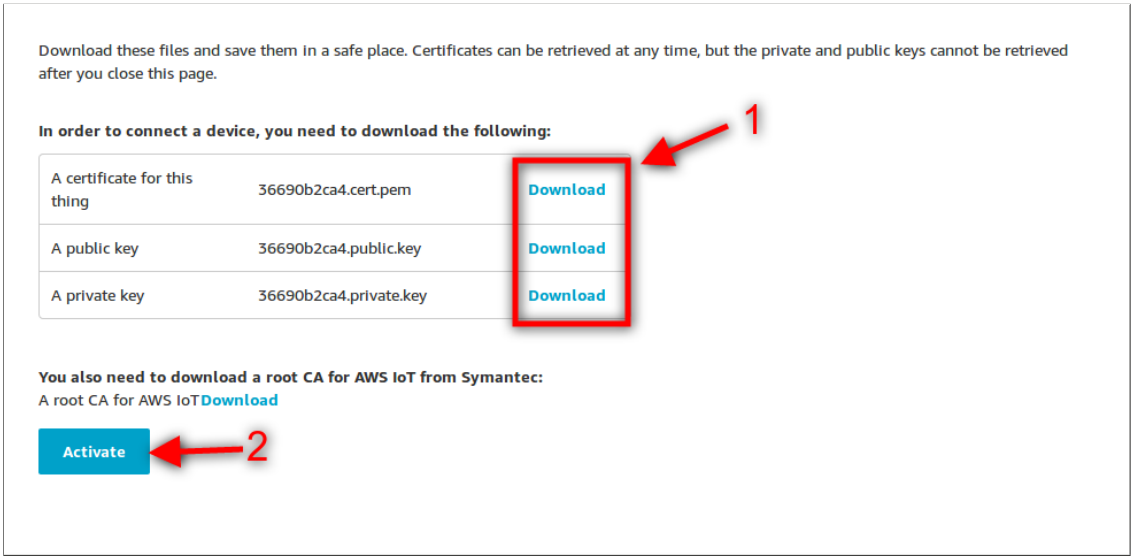


Figure : Save the credential files when connecting device

4. Create a policy with the parameters **iot.*** and ***.

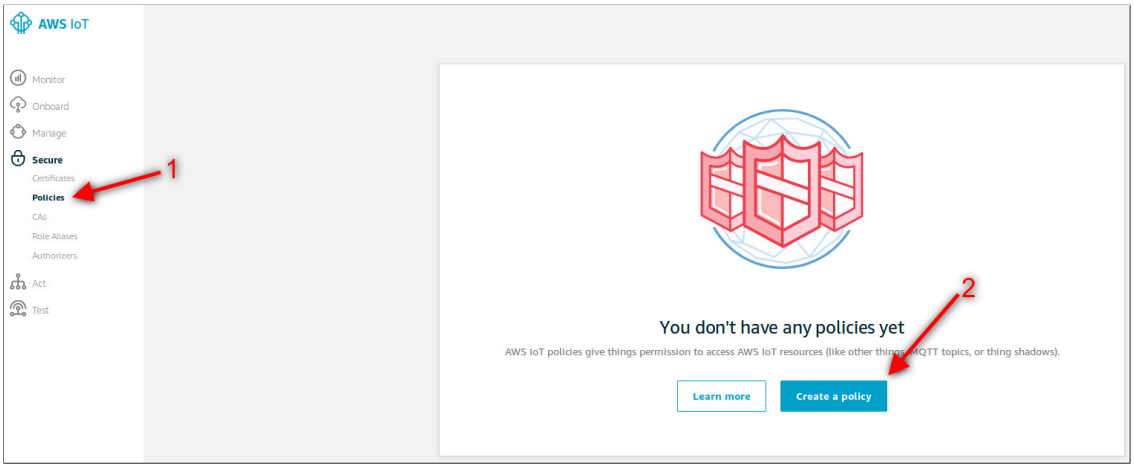


Figure : Create a policy

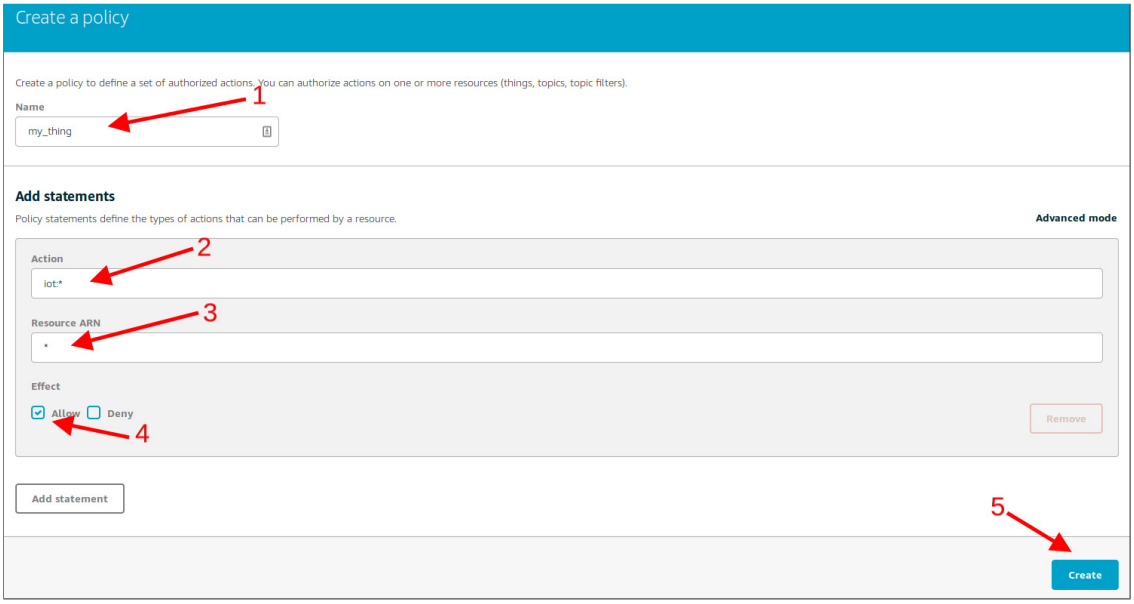


Figure : Fill the policy form

5. Attach a policy.

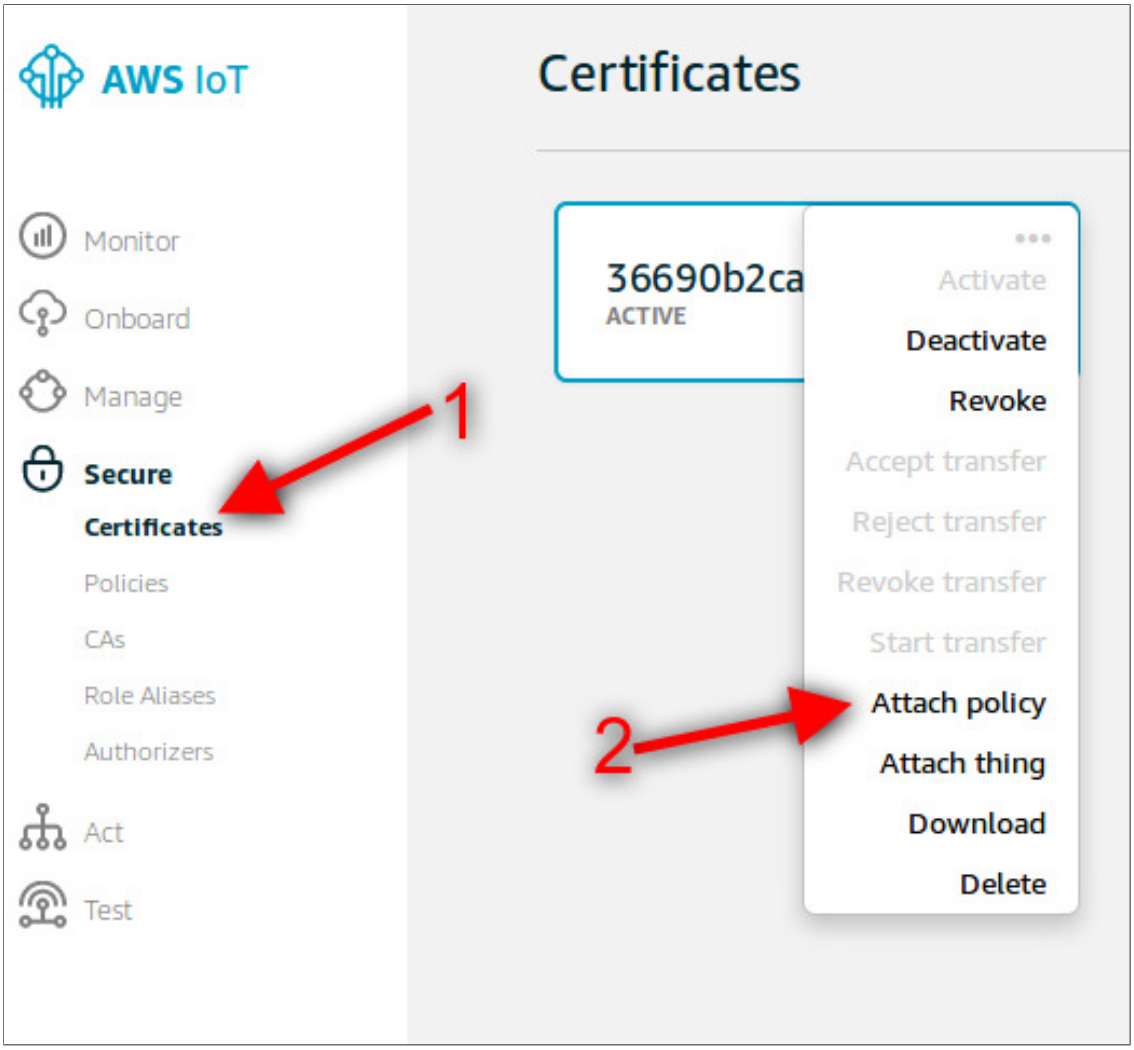


Figure : Select "Attach policy"

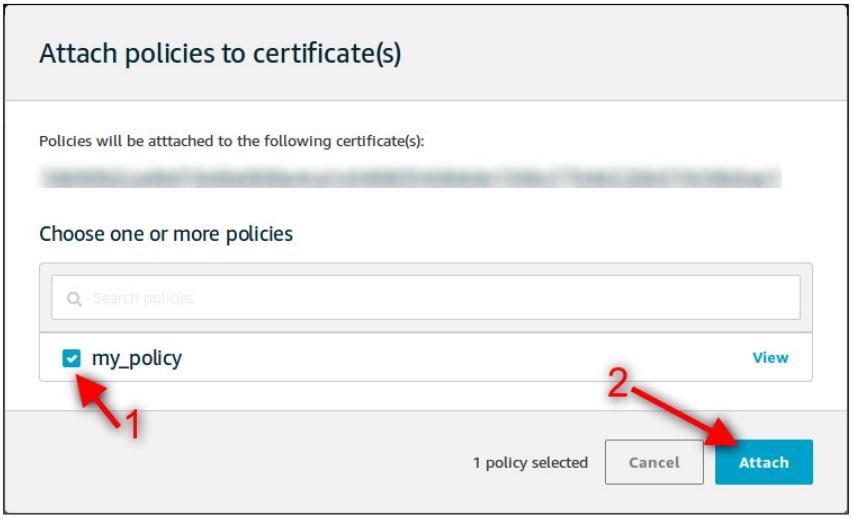


Figure : Attach the policy to the certificate

6. Copy the HTTPS connection string for later use.

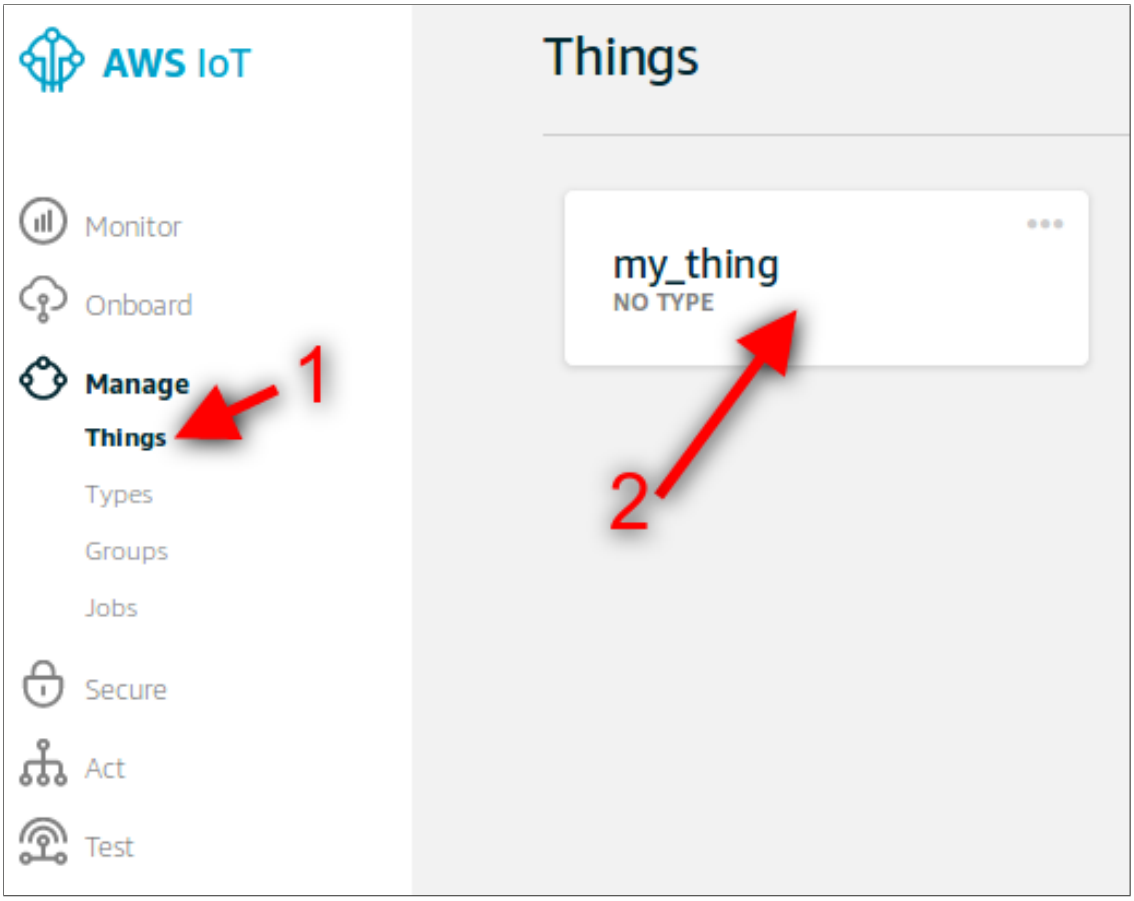


Figure : Select the thing created

Figure : Annotate the value of the field

It is important to annotate the configuration displayed and save the credential files when connecting the device. You will need these files and parameters later for the Meshlium configuration.

Configuration

You will use the previously obtained configuration from the AWS IoT platform to certificate your Meshlium as a valid sender of MQTT messages.

In the Configuration panel, the user can set:

- **Public key**: User public key file previously downloaded.
- **Private key**: User private key file previously downloaded.
- **Certificate**: Certificate file previously downloaded.
- **Host**: HTTPS connection string previously annotated.
- **Port**: AWS IoT MQTT port (by default 8883 for MQTT).
- **ClientID**: AWS IoT Client identification.
- **QoS**: Quality of Service levels for publishing and subscribing to a topic.
- **Log Level**: Generate log messages. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG, REPORT. Default is OFF.
- **Topic template**: Topic of your message. The user can use these wild-cards creating a personalized structure:
  - #ID#: Unique identifier for data.
  - #MESHLIUM#: Host name of the Meshlium unit.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.
  - #TIMESTAMP#: MySQL TIMESTAMP type ('YYYY-MM-DD HH:MM:SS' UTC).
- **Message template**: Data structure of your message. The user can use these wild-cards creating a customized content:
  - #ID#: Unique identifier for data.
  - #MESHLIUM#: host name of the Meshlium.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.
  - #TIMESTAMP#: MySQL TIMESTAMP type ('YYYY-MM-DD HH:MM:SS' UTC).

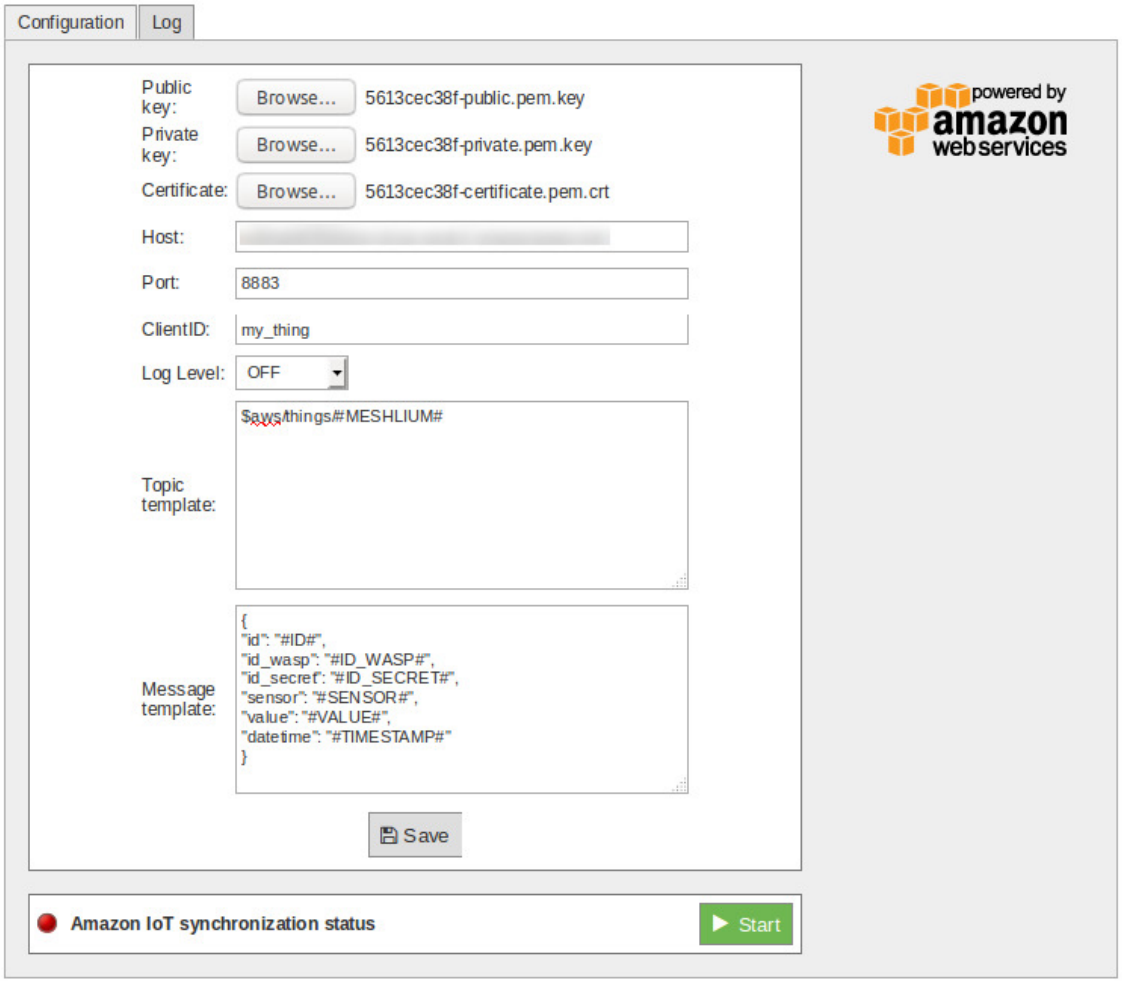


Figure : Amazon IoT configuration panel

Controlling synchronization

Once configured the server/broker, the user can launch the Meshlium Amazon IoT script (Start button). The program will search for the received frames on the local database, and will send them to the Amazon IoT platform via MQTT protocol. The status indicator displays the current state, saying “Running” or “Stopped”.

Figure : Amazon IoT sender is running

You can stop the Amazon IoT program anytime clicking on the “Stop” button.

Figure : Amazon IoT sender is stopped

12.3.3. Amplia's OpenGate

Configuration

Inside the Amplia's plugin you can find the different fields that you must configure for using your Meshlium against OpenGate.

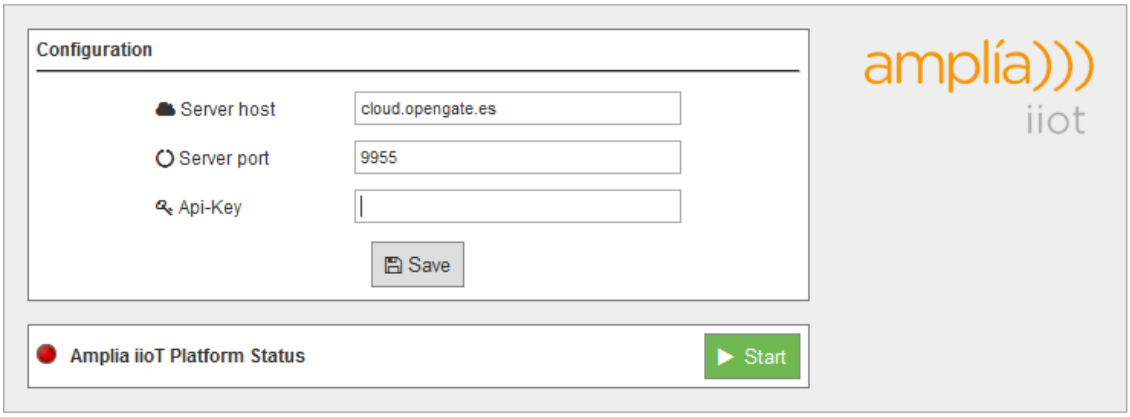


Figure : Amplia Cloud Connector configuration panel

- **Server Host**: You must enter the host name that you are going to use for collecting the Meshlium events.
- **Server Port**: The port where the host is accessible.
- **API-Key**: Security key used for validating the access to the Host.

Click on the "**Save**" button for storing the configuration fields.

After that, press the "**Start**" button, and you will start to receive data from the configured Meshlium.

If you want to stop the event sending, just press the "**Stop**" button.

In the OpenGate OSS web portal you could check the different values collected by Meshlium and by the Waspmote units which have sent messages using the configured Meshlium as gateway.

How to get your own API-key

For getting your own API-key you have to send an e-mail to info@amplia.es and Amplia Solutions will provide one for you. In the same mail send the serial number of your device for creating it in the OpenGate platform.

12.3.4. Aveva (Wonderware)

Wonderware Online Insight Cloud Platform

If you don't already have a valid Wonderware Online InSight account, you can get one for free by registering at either Wonderware's North American or European instance:

- North America website: https://online.wonderware.com.
- European website: https://online.wonderware.eu.

Next, click on the "Sign up" button in the top right corner to get started. Then complete the registration form.

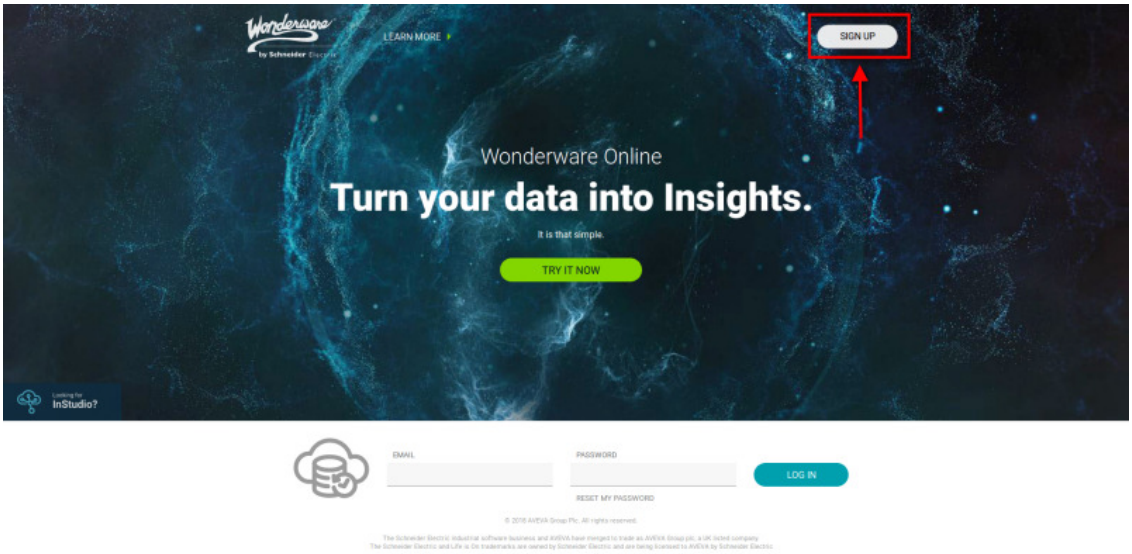


Figure : Wonderware Online InSight landing page with "Sign up" button to get started

You will then be prompted to create your Wonderware Online InSight solution. Provide any name that you like for your new solution that will contain your soon to be published data.

Once your solution has been defined, create a new CSV/JSON data source. To create a CSV/JSON data source, access the Administration page as shown below.

Figure : Wonderware Online InSight Administration page menu access

Once in the administration page, click on "Data Sources".

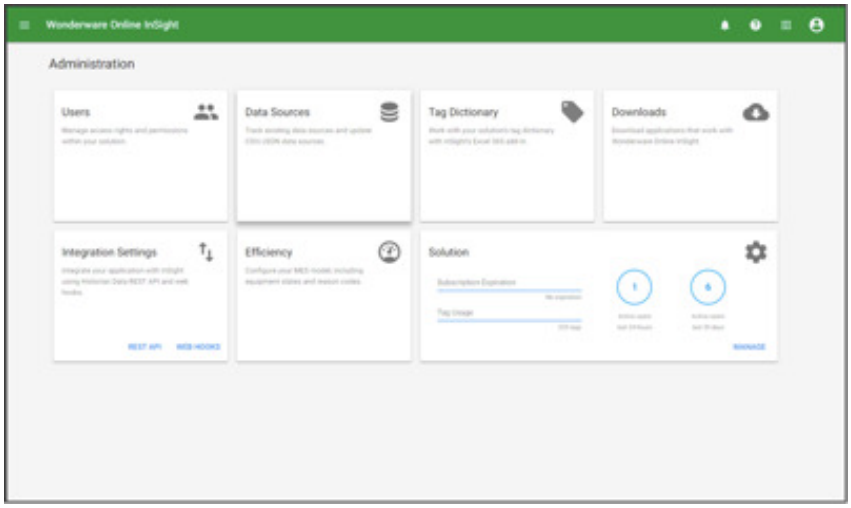


Figure : Wonderware Online InSight Administration page

Click the "+" icon to create a new data source.

Figure : Data Sources menu option highlighting how to create a new CSV/JSON data source

Provide any name that you like for your new CSV/JSON data source and click "OK".

Once your new data source has been created, you will be shown its API authentication token. Click on the "Copy" icon to copy this string to the clipboard.

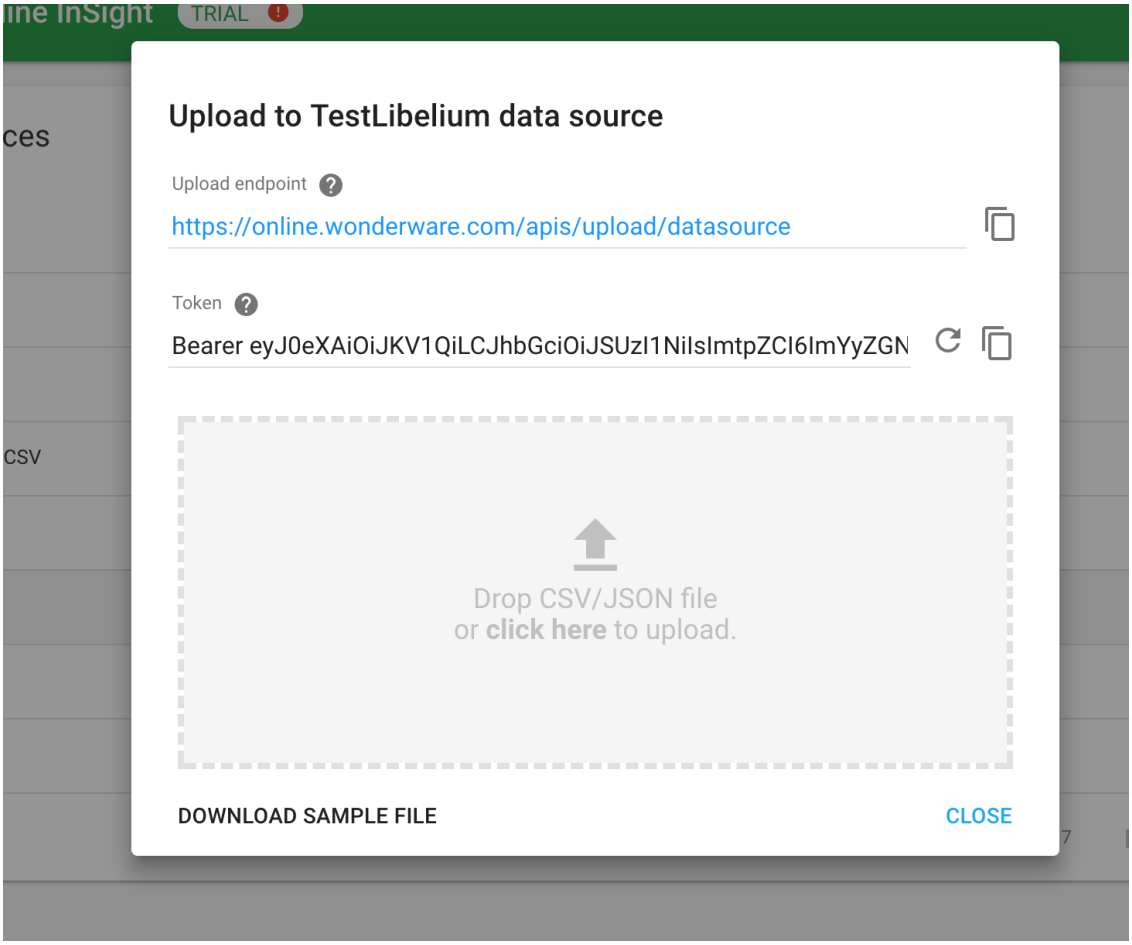


Figure : New CSV/JSON data source

Configuration

To get started, you will simply need:

- A valid Wonderware Online account with administrative access to a solution.
- A defined CSV/JSON data source in your Wonderware Online solution.
- The API Token key for authentication from your CSV/JSON data source to publish your data.

If you already have a valid Wonderware Online InSight account, please copy your API authentication token from the defined CSV/JSON data source and paste in the Token field as shown below.

Go to Cloud Connector → Basic Cloud Partner → Wonderware Online Insights .

In the Configuration panel, the user can set:

- **Endpoint**: Wonderware Online Insight upload endpoint for CSV/JSON payloads.
  - For the upload endpoint, you can specify either the North American or European instance.
  - For more information on the difference between these two regions, please review the trust website at this link: https://www.wonderware.com/trust/secure/ under the heading of "Data Residency and Digital Sovereignty".
- **Token**: Authentication token obtained from your Wonderware Online solution data source configuration menu (see below for an example). Copy and paste it from your Wonderware Online portal to here.
- **Log Level**: Select between VERBOSE, ERROR, INFO and DETAILED to log the appropriate level of diagnostic information as needed.

- **Sync Interval**: Frequency at which the data is synchronization to the cloud.
- **Synchronized tag**: Clear the synchronized tags if you wish to create them again.

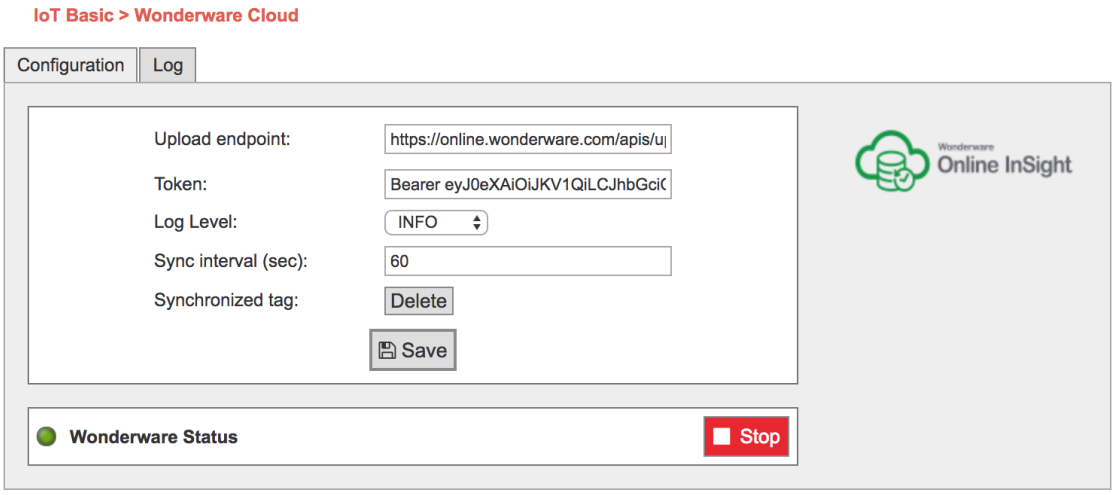


Figure : Wonderware Online Insight Configuration panel

Click on the "Save" button for storing the configuration fields.

Controlling synchronization

With all entries of the configuration filled, you can start the connector by clicking on the "Start" button.

Figure : Wonderware Online Insight sender is running

You can stop the program anytime clicking on the "Stop" button.

Figure : Wonderware Online Insight sender is stopped

Support

Need help? Write to feedback@wonderware.com. Your questions will be answered, helping you to get started.

12.3.5. BaseN

Follow this short instruction to get Meshlium synchronized to BaseN Platform running. This guide assumes that you are already a customer of BaseN and have access to your BaseN admin wiki.

Configuring Microagent receiver in the BaseN Platform

In your chosen wiki page, configure the following minimum setup for synchronizing with the BaseN Platform.

Figure : Configuration of Microagent Receiver

- **Id**: Unique ID of this Microagent receiver.
- **Path**: Measurement data path for data storage.
- **Username**: Username for authentication of Microagent receiver.
- **Password**: Password for authentication of Microagent receiver.

Configuring BaseN Platform in Meshlium

Use the same Microagent receiver parameters in the Meshlium BaseN Cloud Connector plugin.

BaseN plugin is located in:

Manager System → Cloud Connector → Basic Cloud Partner → BaseN

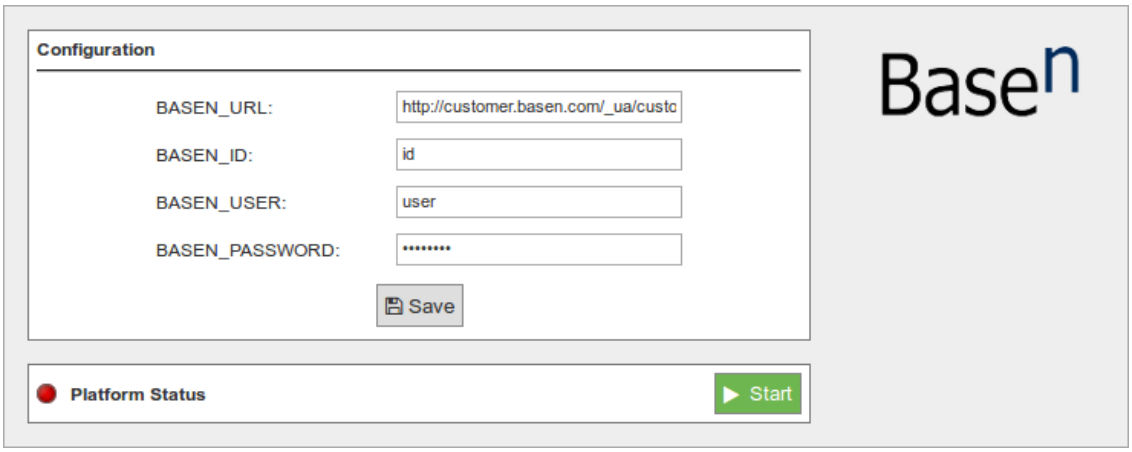


Figure : Configuring BaseN Platform in Meshlium

- BASEN_URL: Usually http://customername.basen.com/_ua/customername/.
- BASEN_ID: Unique ID of this Microagent receiver.
- BASEN_USER: Username to authenticate the Microagent receiver.
- BASEN_PASSWORD: Password to authenticate the Microagent receiver.

After setting the parameters, save the configuration clicking on the “Save” icon.

Start BaseN Cloud Connector

Click on “Start” for Cloud Synchronization. A green icon should start rolling to indicate synchronization is running.

Verify BaseN Cloud Connector synchronization

Go to the BaseN Platform wiki where the Microagent Receiver and MicroAgentConfig were configured, and check under “Debug Information for <id>” and “Stats” that the Observer Request Counts does show requests arriving.

12.3.6. Biz4Intellia

Biz4Intellia is a well-integrated combination of IoT devices, an IoT platform suite, and configurable business services.

More information: http://www.biz4intellia.com/.

With this plugin, Waspmote sensor data can be directly integrated with Biz4Intellia.

Figure : Biz4Intellia plugin

Configuration

You will use the previously obtained configuration from the Biz4Intellia platform to certificate your Meshlium as a valid sender.

In the Configuration panel, the user can set:

- **Public key**: User public key file provided by biz4intellia.
- **Private key**: User private key file provided by biz4intellia.
- **Certificate**: Certificate file provided by biz4intellia.
- **Host**: HTTPS connection string provided by biz4intellia.
- **Port**: MQTT port (by default 8883 for MQTT).
- **ClientID**: Client identification.
- **Log Level**: Generate log messages. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG, REPORT. Default is OFF.
- **Topic template**: Topic of your message. The user can use these wildcards creating a personalized structure:
  - #ID#: Unique identifier for data.
  - #MESHLIUM#: Host name of the Meshlium unit.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.
  - #TIMESTAMP#: MySQL TIMESTAMP type ('YYYY-MM-DD HH:MM:SS' UTC).
- **Message template**: Data structure of your message. The user can use these wildcards creating a customized content:
  - #ID#: Unique identifier for data.
  - #MESHLIUM#: host name of the Meshlium unit.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.
  - #TIMESTAMP#: MySQL TIMESTAMP type ('YYYY-MM-DD HH:MM:SS' UTC).

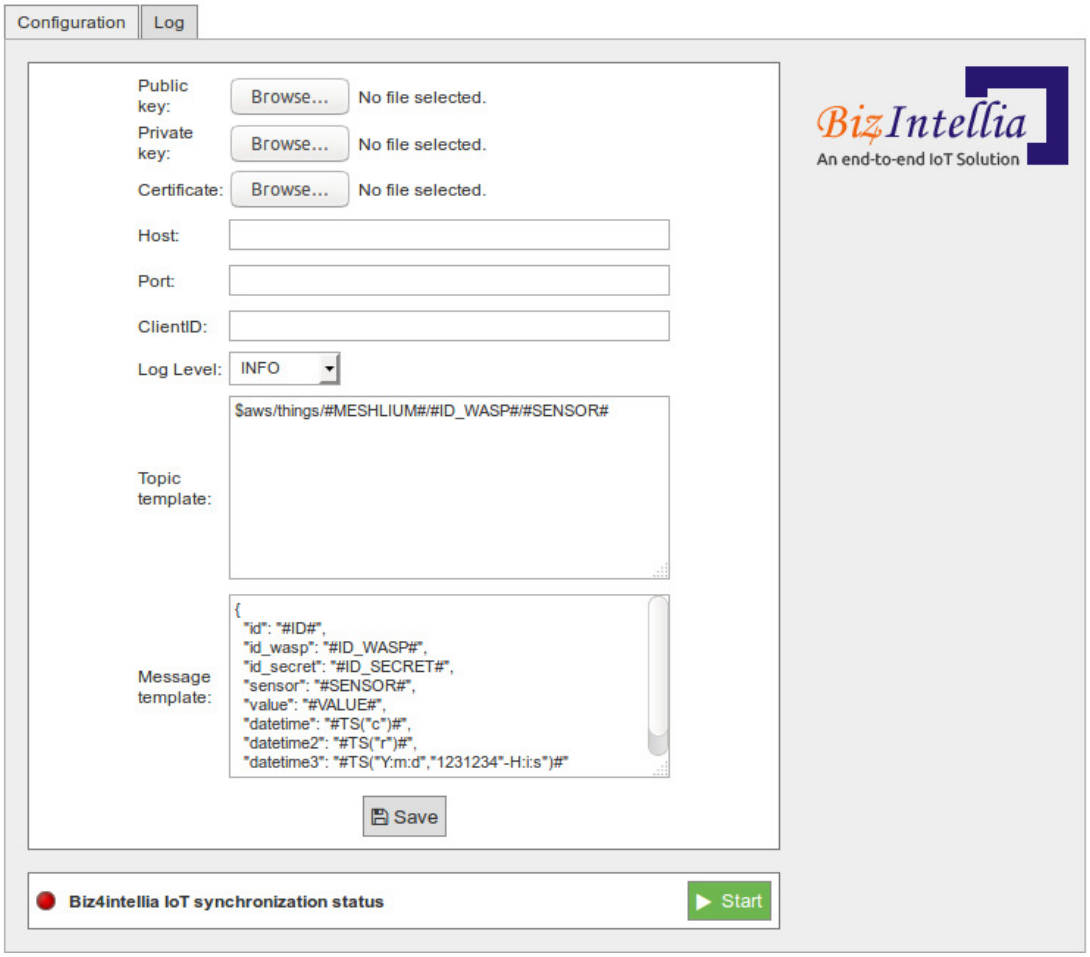


Figure : Biz4Intellia configuration panel

Controlling synchronization

Once configured the server/broker, the user can launch the Meshlium Biz4Intellia IoT script (“Start” button). The program will search for the received frames on the local database, and send them to the Biz4Intellia platform via MQTT protocol. The status indicator displays the current state, saying “Running” or “Stopped”.

Figure : Biz4Intellia sender is running

Figure : Biz4Intellia sender is stopped

12.3.7. IBM Bluemix

IBM Bluemix is a cloud platform as a service (PaaS) developed by IBM that gives a wide scope of services to use the cloud, one of them is based on MQTT communications. This is a great alternative if the user do not want to build his own MQTT server.

Configuration

Configuration options are shown in the M2M Platform menu, enlarging the IBM Bluemix MQTT section. You will notice that the configuration for this plugin is very straight-forward, you have most of the needed parameters on the IBM Bluemix web panel:

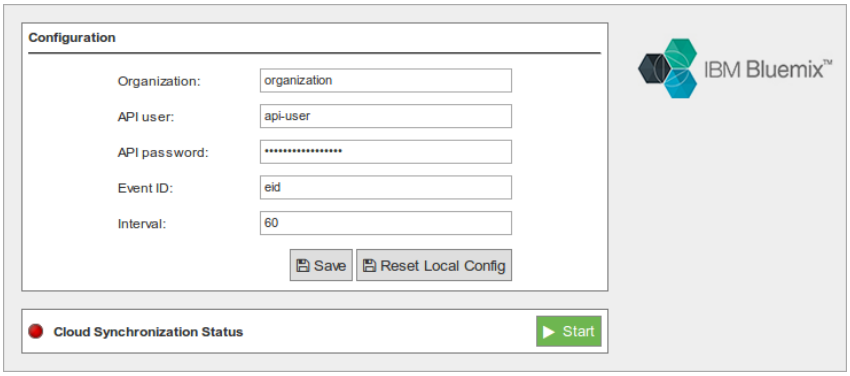


Figure : Configuring IBM Bluemix in Meshlium

- **Organization ID**: Identifier of your organization; you can get it from the platform credentials.
- **API user**: User generated in the API platform section.
- **API password**: Password generated in the API platform section.
- **Event ID**: Used to configure the event where you want to send the information. If you do not know what to type in this field, you can use 'eid'.
- **Interval**: Used to delay the communication after a bunch of messages were sent.

Press the "Save" button for storing the configuration. In the case that you have run a previous configuration, we advise that you also reset the local configuration pressing "Reset Local Config".

Controlling synchronization

You can stop or start the IBM Bluemix synchronization process anytime, hitting on the buttons "Star" and "Stop". Then, the status indicator displays the current state, saying "Running" or "Stopped".

Figure : IBM Bluemix synchronization service is running

You can stop the synchronization anytime clicking on the "Stop" button.

Figure : IBM Bluemix synchronization service is stopped

More information can be found on this Recipe we created for IBM:
https://developer.ibm.com/recipes/tutorials/bluemix-configuration-guide-for-meshlium/

12.3.8. B-Scada

B-Scada® VoT platform allows you to create rich, sophisticated IoT and M2M applications that consolidate and organize data from anywhere, and visualize it in real-time on any device. Connect to thousands of potential data sources. Visualize your data using modern, high-performance customized graphics. Leverage powerful analytic tools and automation. Connect your devices, processes and people in a continuous real-time information system.

More information about VoT Platform: http://www.votplatform.com/.

Configuration

A new option is shown in the M2M Platforms menu: the **B-Scada Cloud Connector**. If you expand it, you can see this form with 6 fields in it:

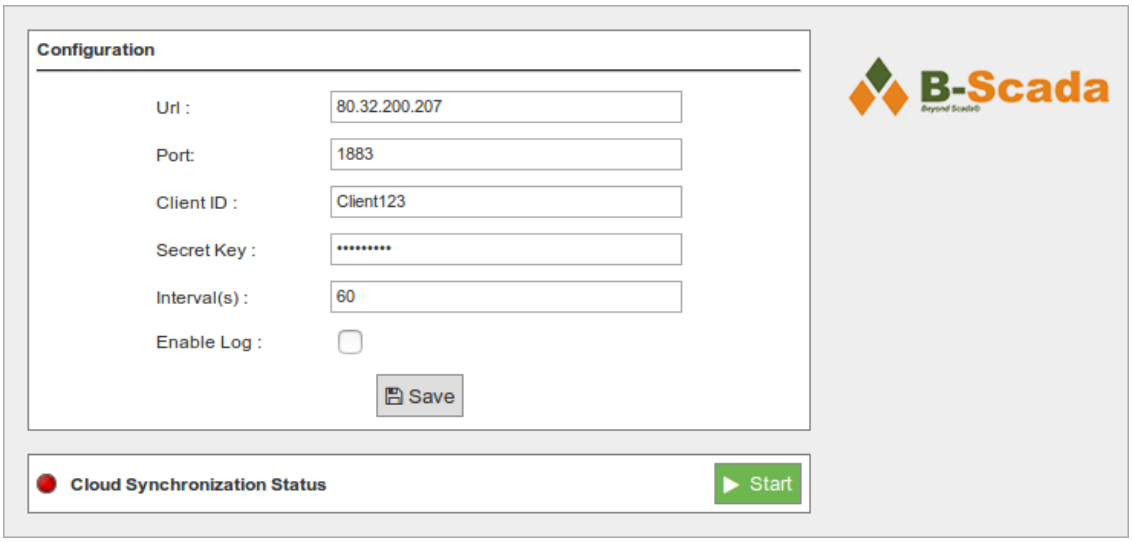


Figure : Configuring B-Scada in Meshlium

- **URL**: IP address of the VoT platform service by B-Scada. This address should be provided by B-Scada.
- **Port**: The port in which the VoT Server is listening to connections.
- **Client ID**: Customer's identifier or company name.
- **Secret Key**: The security key to send encrypted data to VoT.
- **Interval(s)**: Time duration in seconds between operations of updating data.
- **Enable Log**: This option enables the creation of log files to save all communications processed to the VoT server.

Controlling synchronization

The synchronization will be done for all data that has not been synchronized in the Sensor Parser table each time. You can start and stop the data synchronization to the VoT service. In the interface you can see an indicator of whether the status service is running or not. If you click on “Start”, the synchronization will begin.

Figure : B-Scada synchronization service is running

You can stop the synchronization at any moment clicking on the “Stop” button.

Figure : B-Scada synchronization service is stopped

12.3.9. C2M

C2M® is an end-to-end IoT and Digital Enterprise platform that allows easy, secure and rapid prototyping and deployment of IoT/M2M solutions.

Configuring C2M Platform in Meshlium

1. Select the C2M® plugin.
2. Login with your C2M® credentials.

   (If you do not have C2M® credentials, please click the “Sign Up” button at the bottom of the screen to register).

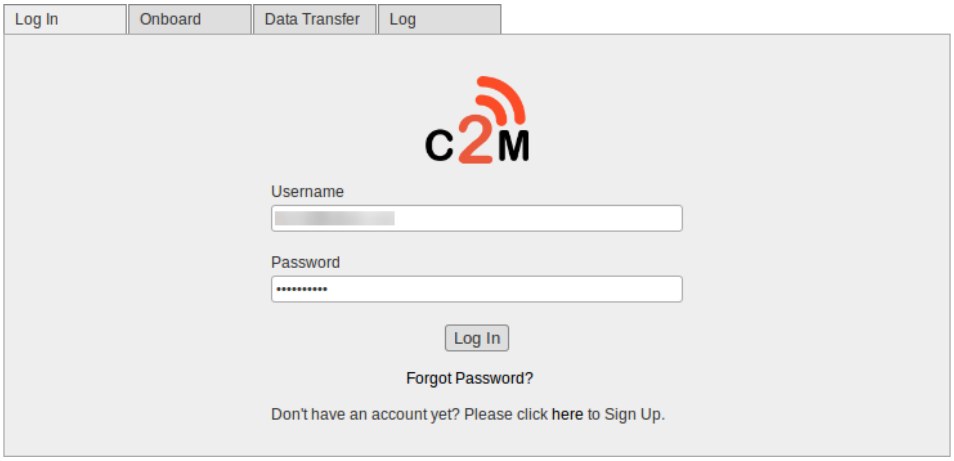


Figure : C2M logging

Controlling synchronization

1. Select the Onboard tab and press the “+” button on the Waspmote/Sensors that you wish to onboard.
   - Turn the switch On to enable the Waspmote/Sensor.
   - To Disable the Channel: Toggle the On/Off switch to Off.
   - To Delete the Channel: Press the Trashcan icon.

   Note: This will delete the channel and all data will be permanently removed.

   For a temporary pause in sending data, see next step.

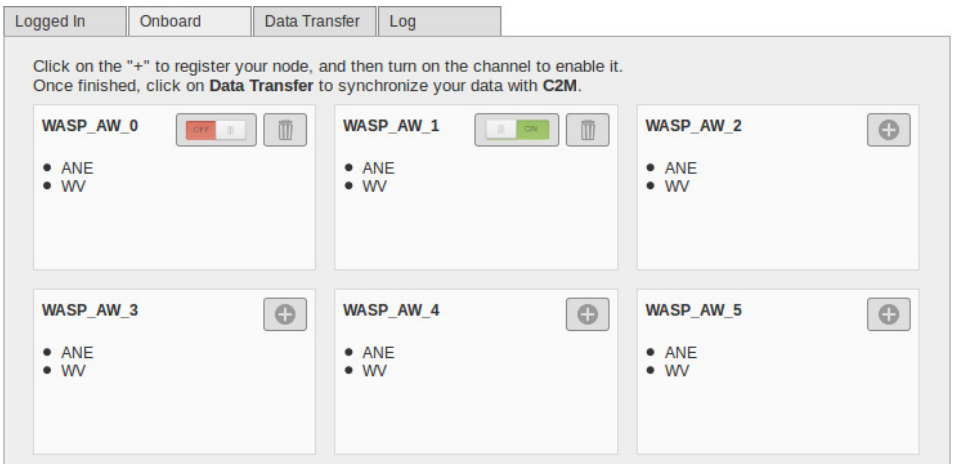


Figure : C2M Onboard tab

2. Select the Data Transfer tab. Here you will see your enabled Waspmote/Sensors. Select your Transmission method, toggle on the devices in which you want to send data and press “Save”.
3. Click on the “Start” button to begin the C2M Sync service. Click the “Stop” button to terminate the C2M Sync service.

   To temporarily pause a node, toggle the On/Off switch to Off and press “Save”.

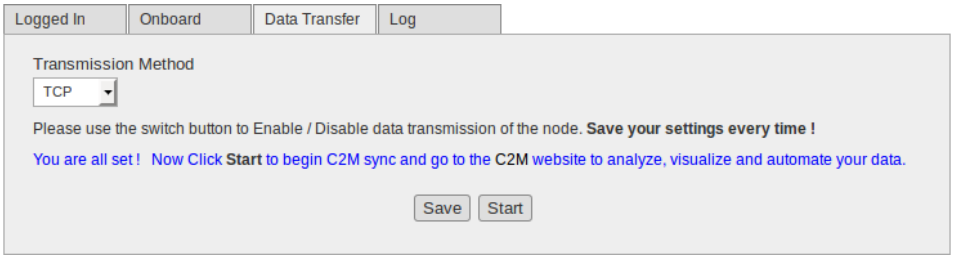


Figure : C2M Data Transfer tab

4. Login to your C2M account at https://cloud.c2m.net/login.aspx to analyze, visualize and automate your data.

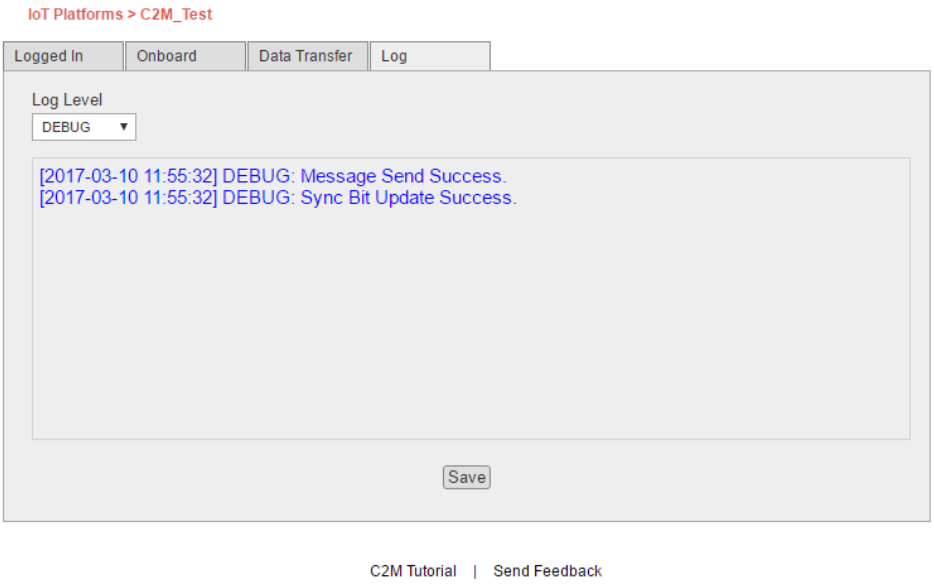


Figure : C2M log

There is also a Log tab. Here you will find access to various log levels. Default log view is set to WARNING. Once you change the Log Level you must press “Save” to store this operation and to view the updated information.

12.3.10. Cumulocity

Cumulocity IoT solution allows for a seamless integration of your Meshlium and Waspmote devices to the cloud. Once Meshlium is connected to this cloud, the Cumulocity IoT solution will automatically manage all your devices. Registering and gathering device data to Cumulocity Platform provides a fully customizable interface, allowing for deep data analysis.

Cumulocity works with a multitude of devices, centralizing all your IoT technologies in a single place. For more information, visit www.cumulocity.com.

Configuration

First of all, make sure your Meshlium is receiving data from your Waspmote or Plug & Sense! units. Please access to the Cumulocity cloud configurator in the Manager System. You need to fill the following fields with your correct Cumulocity account settings.

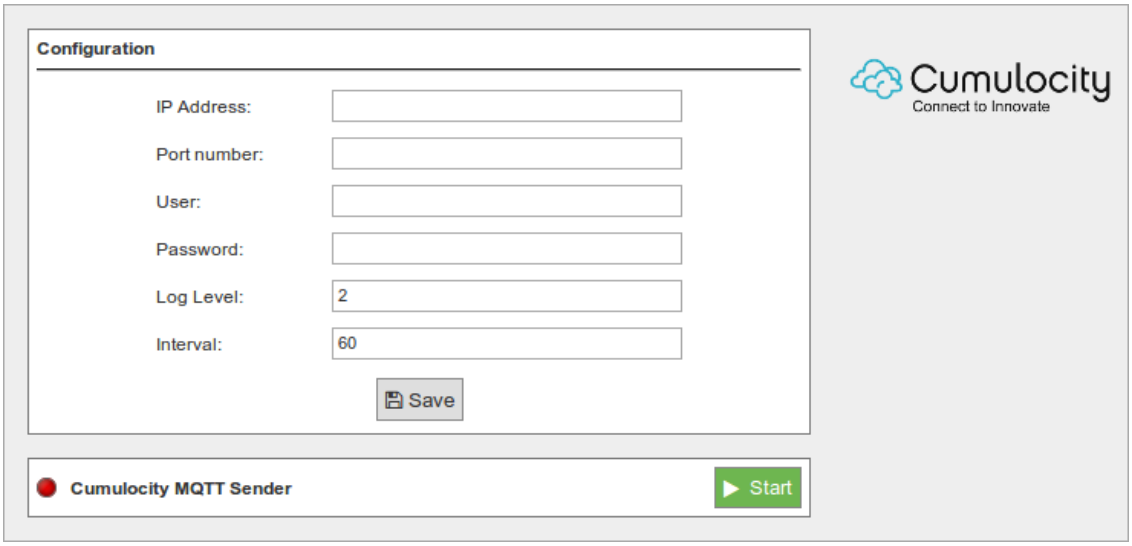


Figure : Configuring Cumulocity in Meshlium

- **IP address**: Address provided by Cumulocity.
- **Port number**: Port Number provided by Cumulocity.
- **User**: Username provided by Cumulocity.
- **Password**: Password provided by Cumulocity.
- **Log level**: Generate log messages. Valid values go from 0 (no log) to 4 (debug).
- **Interval**: Time duration in seconds between synchronizing data batches.

Then click “Save” to store your new settings.

After that, the only thing left to do is to activate the Cumulocity MQTT Sender by clicking on “Start”.

The Cumulocity platform will handle all the devices’ creation and data gathering for you. You can access now to your Cumulocity account with your browser and you should be able to see your devices and incoming data.

12.3.11. DeviceLynk

DeviceLynk is a high-level cloud service based on the ThingWorx cloud. Interface walk-through:

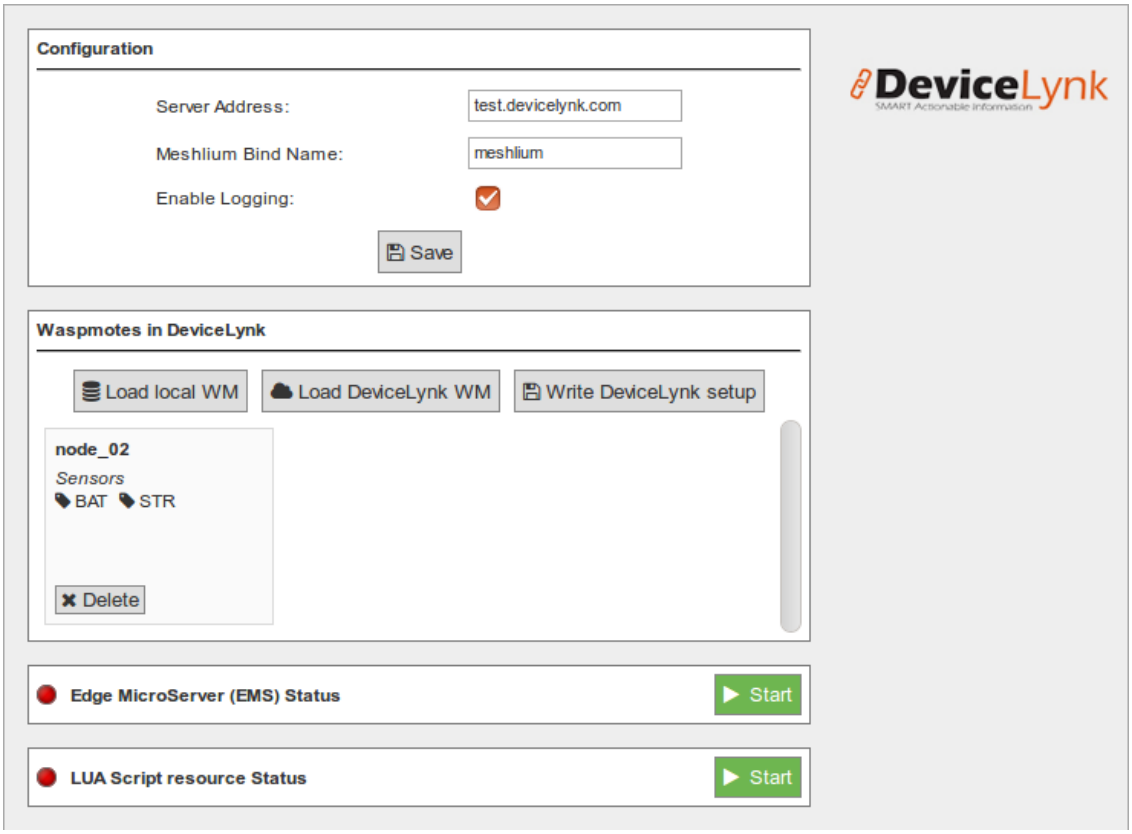


Figure : DeviceLynk cloud plugin interface

Interface elements:

- **DeviceLynk logo**: Click to jump to the DeviceLynk website.
- **Server Address**: The DeviceLynk Server address you wish to connect to.
- **Meshlium Bind Name**: The name that the DeviceLynk Server uses to identify the Meshlium unit.
- **Enable Logging**: Check-box to enable/disable internal logging.
- **Save**: Click to save the Server Address, Meshlium Bind Name, and Logging configuration.
- **Load Local WM**: Click this button to load the list of Waspmotes that the Meshlium is connected to.
- **Load DeviceLynk WM**: from DeviceLynk button Click to load the list of Waspmotes which are connected to the DeviceLynk Server.
- **Write DeviceLynk setup**: Click to write the current list of Waspmotes to the DeviceLynk Server (Waspmotes details are sent to the Server).
- **Waspmote**: Click the “Delete” button to remove that Waspmote unit from the current Waspmote clicking the “Load Local Waspmotes” button, or by clicking the “Load WM Config” from DeviceLynk button.
- **Edge Microserver (EMS) Status**: The status of the DeviceLynk Agent will be indicated, displaying “Running” or “Stopped”. Click button to to start/stop the service.
- **LUA Script resource Status**: The status of the LUA Script Resource will be indicated, displaying “Running” or “Stopped”. Click button to to start/stop the service.

Steps to start the DeviceLynk plugin:

1. Type the DeviceLynk Server address that you wish to connect to in the Server Address field.
2. Type the Meshlium Bind Name that the DeviceLynk Server will use to identify the Meshlium device.
3. Click the “Save” button.
4. Click the “Load Local Waspmotes” button. All Waspmotes connected to this Meshlium unit will show up.
5. Delete the Waspmotes that you do not want to be connected to the DeviceLynk Server by clicking on their respective “Delete” button.
6. Click the “Write DeviceLynk Setup” button to make the DeviceLynk Server listen to those Waspmotes.
7. To show the Waspmotes units that the DeviceLynk Server is currently listening to, click the “Load WM Config. from DeviceLynk” button.
8. Click the “Start” button under “DeviceLynk Agent Status”. “Running” will be displayed.
9. Click the “Start” button under “LUA Script Resource Status”. “Running” will be displayed.

12.3.12. eagle.io

eagle.io is a hub connecting monitoring assets, engineers and decision makers. Acquire data in real-time from Meshlium and Waspmote devices, receive alerts for critical events, and share access with stakeholders. Transform your time-series data into beautifully presented, actionable information.

More information can be found at www.eagle.io.

Configuration

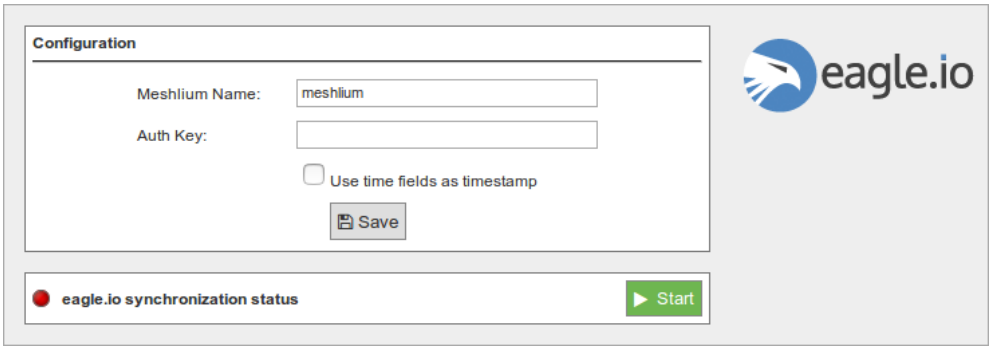


Figure : eagle.io cloud connector configuration panel

The eagle.io plugin is configured with the following three parameters:

- **Meshlium Name**: a name to help identify this device (required parameter).
- **Auth Key**: optional secret key; if this is defined then the same key will be required when configuring the device within eagle.io as a data source. If this key is not defined, then the Meshlium device ID is sufficient to identify the device within eagle.io configuration.
- **Use time fields as timestamp**: if this box is checked, then any time field contained in a Meshlium database record will be used as the eagle.io timestamp for the record. If this box is not checked, the data reception time of the record will be used as the eagle.io timestamp.

After changing any of these parameters, save the configuration by clicking the "Save" button, then restart the eagle.io synchronization (if it is currently running) by clicking the "Stop" button followed by the "Start" button.

Controlling synchronization

The synchronization will be done in batches of 200 records at a time, so the system is not overloaded. The time between batches is 60 seconds. This means that when synchronization is first started on a device with many existing records in the database, it may take some time for all the records on the device to be synchronized with eagle.io.

When the synchronization is not running (red status indicator), it can be started by clicking the green "Start" button.

Figure : Eagle.io start button

When the synchronization is running (green status indicator), it can be stopped by clicking the red "Stop" button.

Figure : Eagle.io stop button

12.3.13. Esri

ArcGIS is a complete spatial information platform provided by **Esri**, that allows to create, analyze, store and spread data, models, maps and 3D globes. It can be accessed via desktop application, browser or handsets. ArcGIS is targeted at GIS professionals, location analysts and developers that want to create their own applications based on geographical data.

More information: http://www.esri.com/products

Waspmote sensor data could be integrated into your existing maps and ArcGIS applications following the configuration steps described for ArcGIS Online service.

ArcGIS Online

We can configure in this form all the parameters needed to connect and send data to the ArcGIS Online platform.

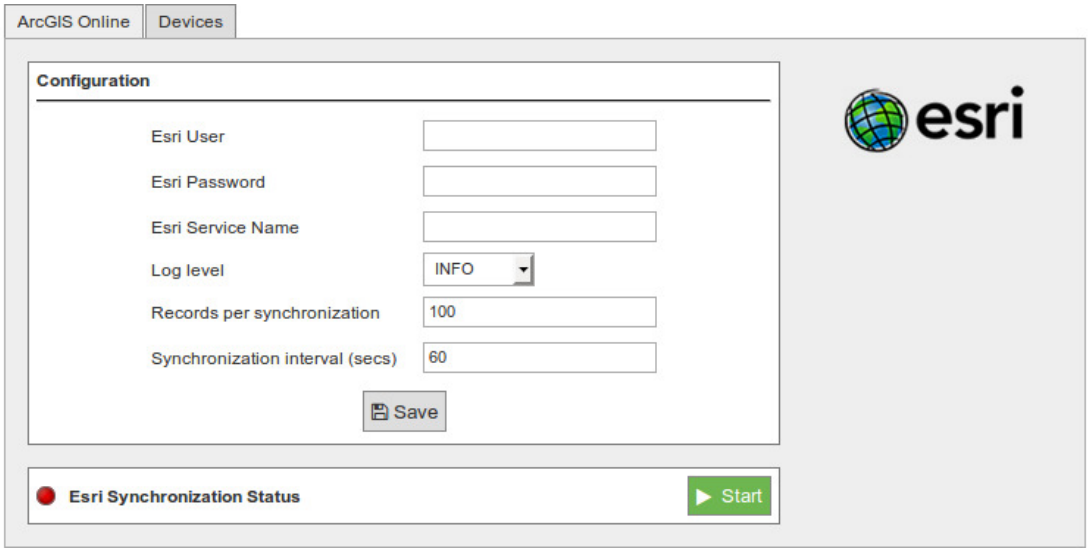


Figure : ArcGIS Online configuration

The parameters to setup are:

- **esri_user**: User for the Esri ArcGIS online platform.
- **esri_password**: Password for this Esri user.
- **esri_service_name**: Name of the service which will receive the data.
- **Log level**: Generate log messages.
- **Records per synchronization**: Records sent for each synchronizing data batch.
- **Synchronization interval**: Time duration in seconds between synchronizing data batches.

Clicking on the "Save" button, this setup is sent to the ArcGIS online service.

Clicking on the "Start" button enables the Esri Cloud Connector to send data periodically to the ArcGIS Online service previously configured. A "running" status is displayed on screen showing that the Cloud Connector is sending data.

Figure : ArcGIS Online "Start" button

Clicking on the "Stop" button will disable the Esri Cloud connector so Meshlium device stops feeding the ArcGIS Online service with data.

Figure : ArcGIS Online "Stop" button

Check the Feature Server in ArcGIS Online

In order to check that data is arriving to ArcGIS Online, you should login in the platform:

https://www.arcgis.com/home/signin.html

Click on the option named "Gallery" and you should see a new Feature Server with the name that you provided in the Meshlium configuration plugin:

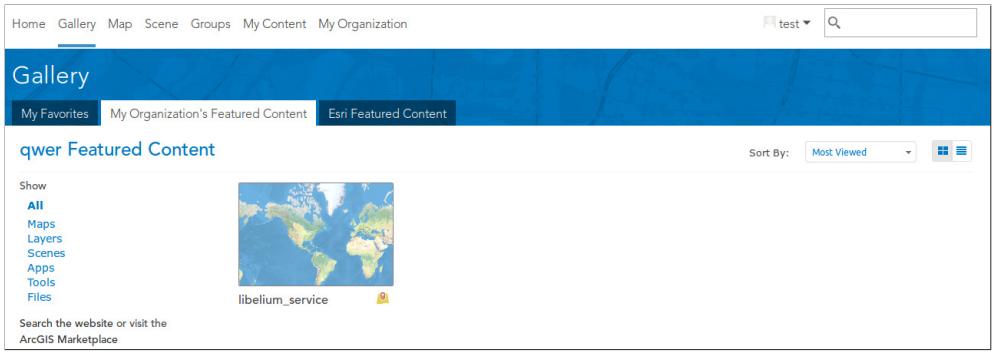


Figure : ArcGIS Gallery

Opening the new content, you should see a map where each layer is one sensor type available in your project. Clicking on the table icon, all the data collected for this type of sensor will be displayed.

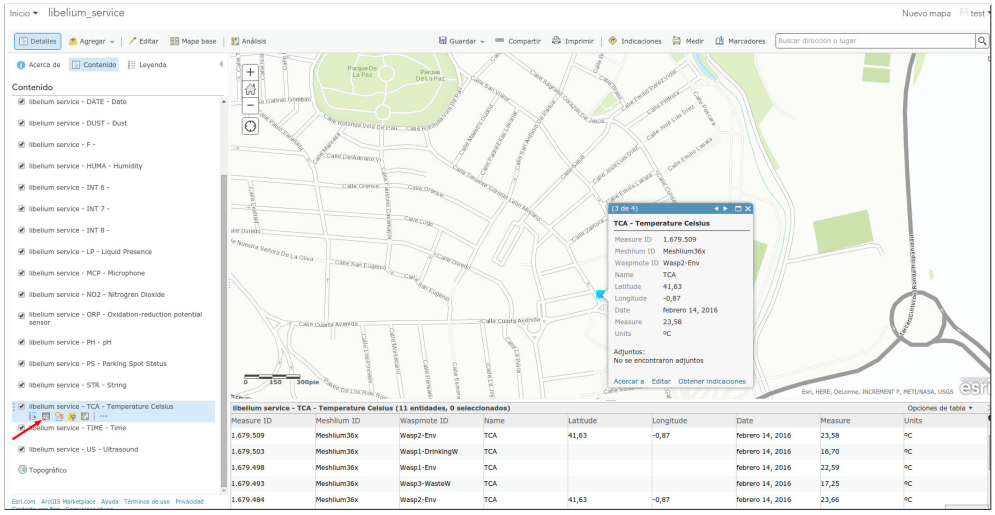


Figure : ArcGIS sensor map view

At this point, it is possible to use this data to create new maps, collaborative apps or analytics making use of the complete array of services provided by ArcGIS Online: https://developers.arcgis.com/en/.

Devices

Meshlium

In the Meshlium section, the user can set and modify the name and description of the Meshlium. You can select the option “Use GPS values”. This option overrides the Waspmote’s positions with the Meshlium’s position obtained by the GPS (except if the position is obtained with Waspmote’s GPS sensor).

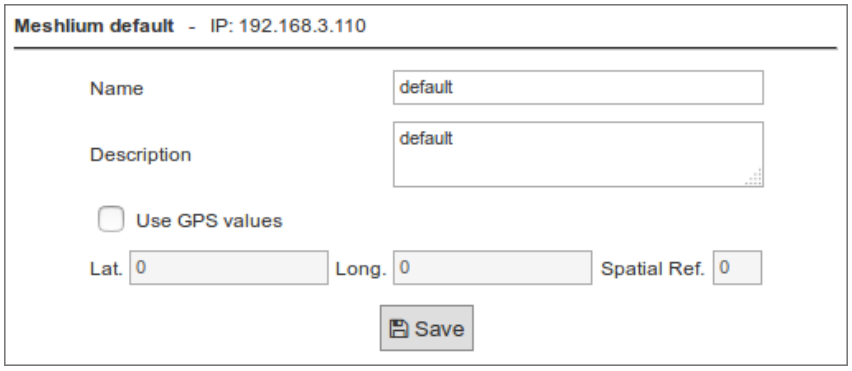


Figure : Meshlium info in ArcGIS

Waspmotes

In the Waspmotes section, the user can manage the Waspmote units which are sending information to Meshlium.

Figure : Waspmotes list in ArcGIS

To add a new Waspmote, click on “Add new”. Then fill up this information:

- Name: The Waspmote name. **Must match with the Waspmote identifier used with the frame**. See chapter “Capturing and storing sensor data” for more information.
- Description: A description of that Waspmote unit.
- Sensor count: Number of sensors on that Waspmote. **Must match with the number of fields of the frame**. See chapter “Capturing and storing sensor data” for more information.

And click on the “Add” button.

To modify a Waspmote, click on the Waspmote name for showing the attributes view.

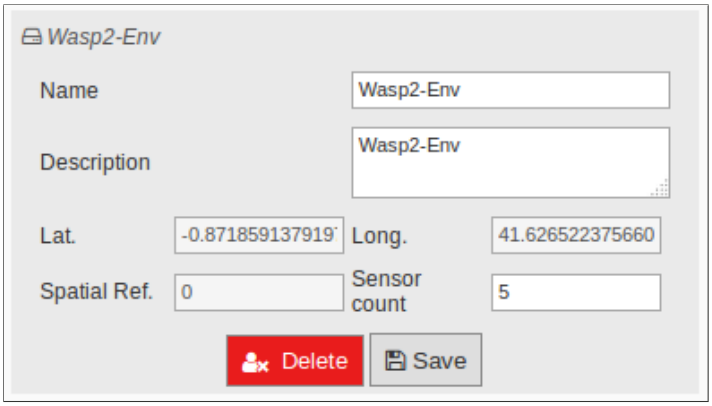


Figure : Modify Waspmote in ArcGIS

Then the user can modify the name, description, and sensor count information. To save the properties, click on "Save".

To delete this Waspmote unit, click on "Delete".

Devices location

In the section Devices location, there is a viewer where the user can see Meshlium and Waspmote located on a map.

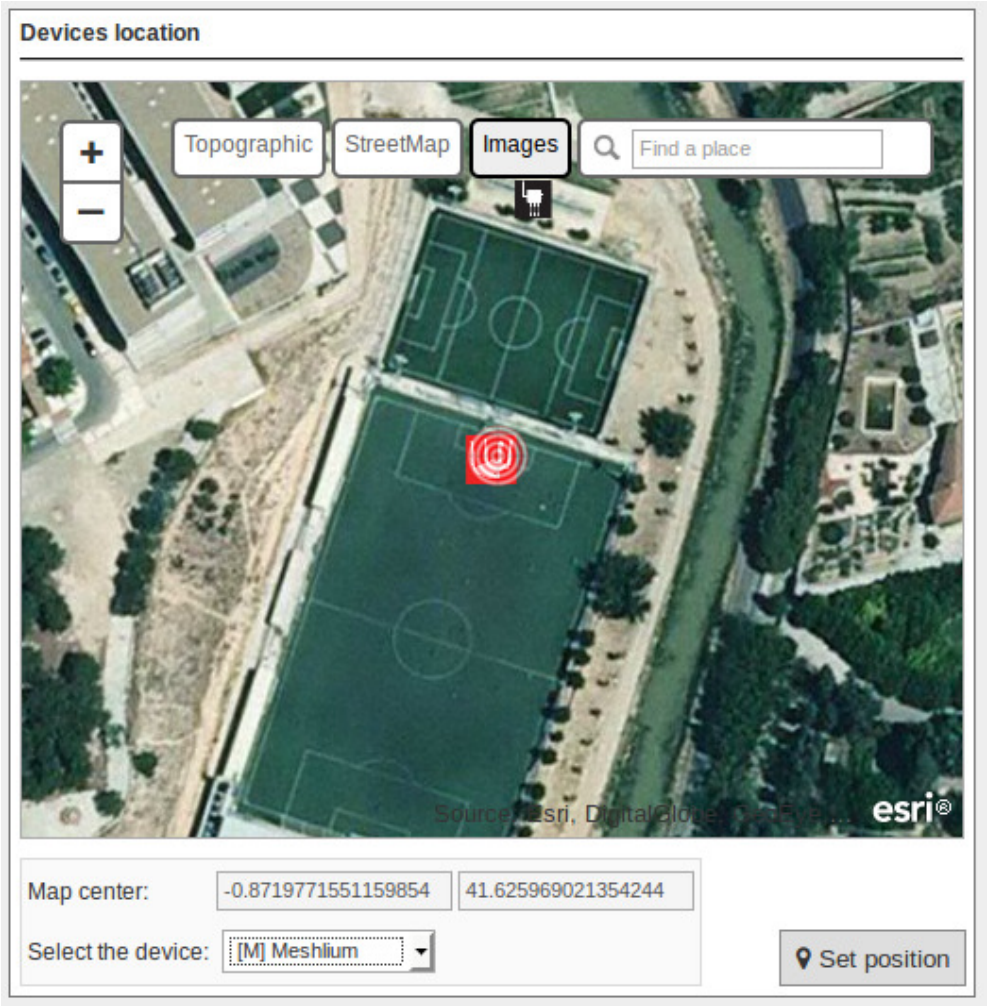


Figure : Devices location in ArcGIS

To change the location of the devices, center the map on the desired location, select the device, and click on "Set Position".

12.3.14. Extunda

Extunda IoT platform (http://www.extunda.com/) is a horizontal platform which also enables vertical applications. Libelium Smart Cities, agriculture and various devices are ready to be launched for service over Extunda IoT platform. The sensor data can be gathered, analyzed, stored and reported so the users can interpret and develop actions based on online data.

Extunda uses MQTT structure for the integration of your Meshlium devices to its platform easily. Therefore, the connector will easily send the sensor data to Extunda platform.

Configuration

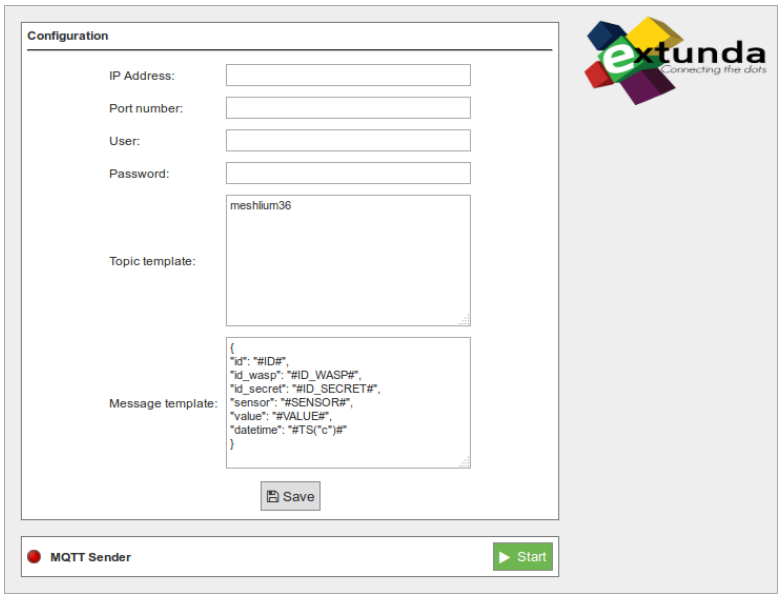


Figure : Extunda cloud connector configuration panel

- **IP Address / URL**: The IP address or the URL will be provided to you by Extunda.
- **Port Number**: This is the port which Extunda server is listening for connections.
- **User Name & Password**: This is the Extunda Server username & password to be used for connecting to Extunda servers. This information will be provided by Extunda.
- **Waspmote ID**: When you login to your Extunda IoT platform account with your username and password, you are authorized to define and match your Waspmote with a specific definition (i.e. Istanbul_gases_1). Your Waspmote data will be transfered to the server as in the above message template.

Controlling synchronization

The synchronization will be done for all data that has not been synchronized in the Sensor Parser table each time. You can start and stop the data synchronization to the Extunda service. In the interface you can see an indicator of whether the status service is running or not. If you click on “Start”, the synchronization will begin.

Figure : Extunda start button

You can stop the synchronization at any moment clicking on the “Stop” button.

Figure : Extunda stop button

12.3.15. Fujitsu IoT Connector

The Fujitsu IoT Connector provides an interface between Meshlium and the Fujitsu IoT Cloud Service (K5). The Fujitsu IoT Connector forwards the sensor data captured by Meshlium to the Fujitsu IoT Cloud Service (K5).

The Fujitsu IoT Connector supports different protocols in order to support different bandwidth and security considerations. The following sections describe the configuration options of the Fujitsu IoT Connector. For more information on the use and configuration of the Fujitsu IoT Cloud Service (K5) see the documentation:

https://k5-doc.jp-east-1.paas.cloud.global.fujitsu.com/doc/en/service_doc.html

General configuration

The following configuration settings apply to both MQTT and REST configurations:

- **Transmission Type**: Select MQTT or REST transmission type.
- **Host**: Server host address, either of the webserver if using REST, or the message broker in the case of MQTT.
- **Port**: Server's port number to use for connection, the default ports are as follows:
  - **REST without SSL**: 80
  - **MQTT without SSL**: 1883
  - **REST with SSL**: 443
  - **MQTT with SSL**: 8883
- **Use SSL Security**: Selects if the SSL security is to be used by the selected protocol. Both REST and MQTT messages can be secured via SSL encryption.
- **Server Certificate**: Identifies if a specific certificate file is to be uploaded and used by the Meshlium connector to validate the server (only valid when SSL security is enabled). Tick the checkbox to use a specific server certificate uploaded onto the Meshlium device. Such server certificates are required to be in the DER encoded format for uploading to the Meshlium device. Clear the checkbox to use the default set of certificate authority server certificates embedded on the device.
- **Client Keys**: Identifies if a specific public private keystore file is to be used by the Meshlium connector in encoding the connection (only valid when SSL security is enabled). Tick the checkbox to use a set of public-private keys uploaded to the Meshlium device. The client keys are required to be in the PKCS#12 keystore file format, the keystore file is itself encrypted by a password which is required to be configured if using a specific 'Client Keys' file. Clear the checkbox next to 'Client Keys' whereby no specific set of public-private keys are specified by the Connector.
- **Keys Password**: Keystore password associated with an uploaded 'Client Keys' file (only valid when SSL security is enabled and a Client Key file is in use).
- **Log Level**: Identifies the level of logging produced by the Connector.

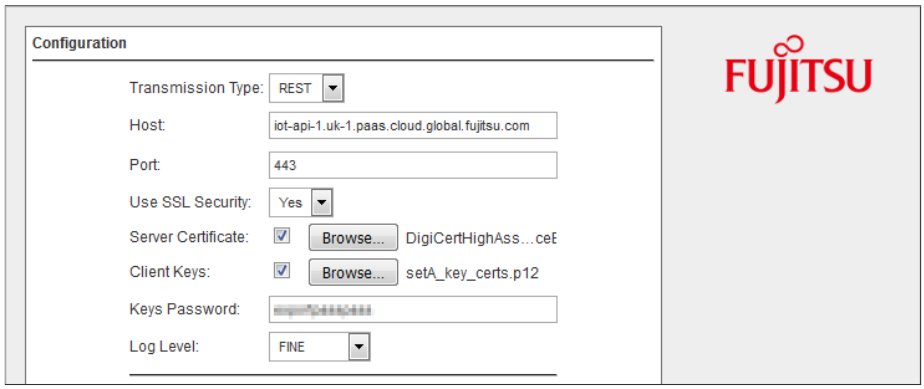


Figure : Meshlium Fujitsu IoT Connector - global configuration parameters

Some parameters must be configured in the Fujitsu IoT Cloud Service (K5) platform portal, navigate to the following URL and login with a valid user:

https://iot-portal-1.uk-1.paas.cloud.global.fujitsu.com/LoginShow

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

Transmission Type

Navigate to the 'Access Code' tab and select the Access Code that will be associated with the Meshlium device. The Access Protocols are then shown or configured on the 'Access Code Information Update' dialogue:

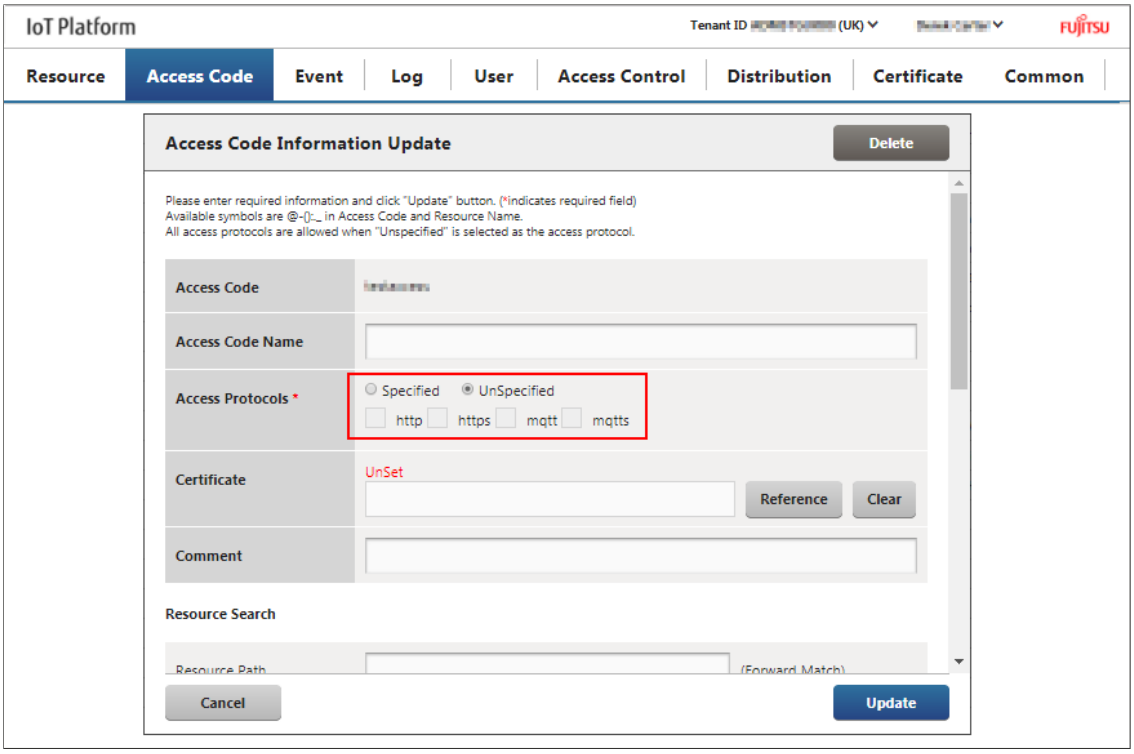


Figure : Fujitsu IoT Platform Portal - Access Code information update dialogue, Access Protocols

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

Host

Navigate to the “Resource” tab, and select the Resource that will be associated with the Meshlium device. The service hostname is then contained within the REST URL displayed in the ’Resource Information Update’ dialogue:

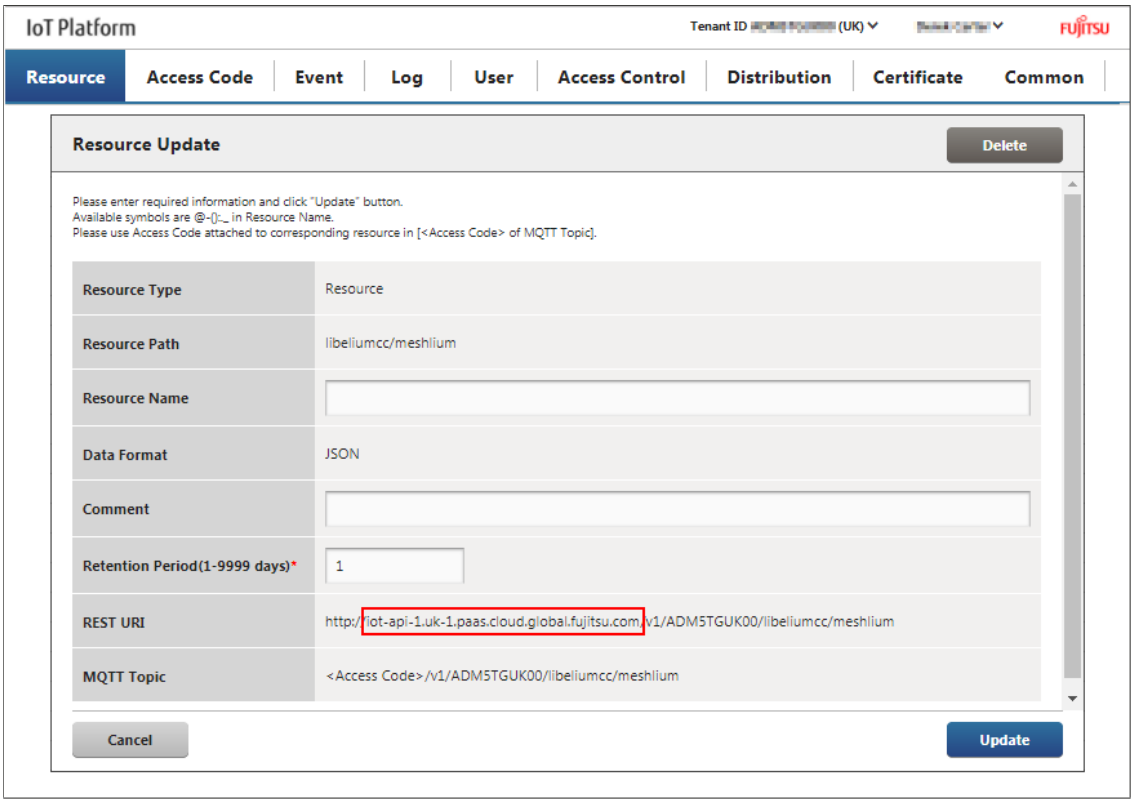


Figure : Fujitsu IoT Platform - resource information update dialogue, Hostname

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

SSL Security

Navigate to the “Certificate” tab to get a valid PKCS#1 file, navigate to “User” tab and select “Generate” button to upload a valid keystore file:

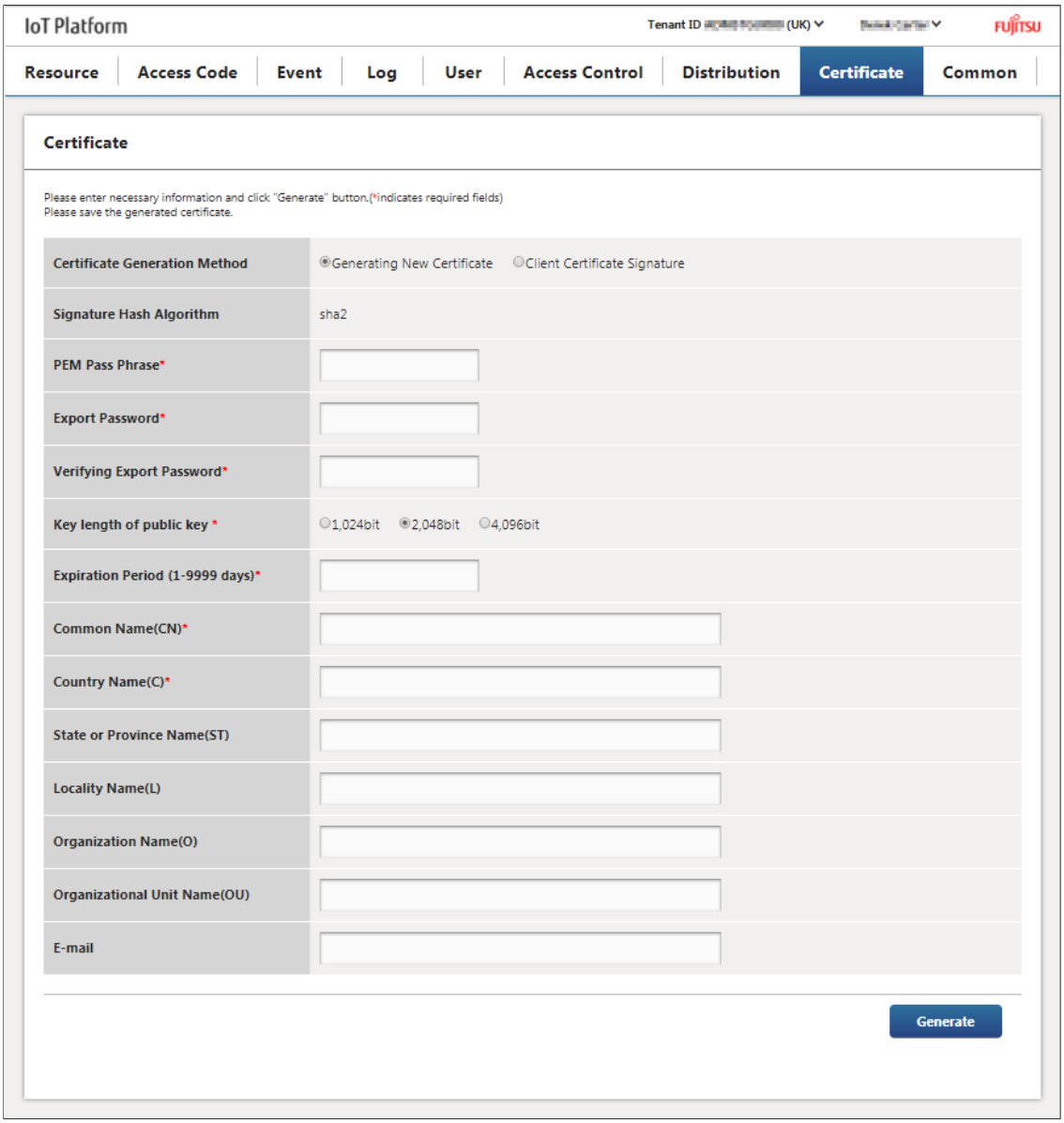


Figure : Fujitsu IoT Platform Portal - certificate and keys generation

Note that in order for the Meshlium to be correctly configured to use a Keystore file, the ’Export Password’ that was used in its creation is required to be configured on the Meshlium. The Fujitsu IoT Cloud Service (K5) currently uses “DigiCert SHA2 High Assurance Server CA” provided by DigiCert, Inc. for SSL/TLS server certificates. This certificate is currently embedded on the Meshlium device with an expiry date of 10th November 2031 see:

https://www.digicert.com/digicert-root-certificates.htm#roots

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

MQTT Configuration

The following configuration settings apply to the IoT Connector when the Transmission Type is set to the MQTT protocol:

- **MQTT User**: Server user name to log into the MQTT broker.
- **MQTT Password**: Server password to log into the MQTT broker.
- **MQTT QoS Level**: Level regarding the delivery of messages to MQTT broker.
- **Topic template**: Topic of MQTT message. The user can use the following substitution-expressions to create a personalized structure:
  - #MESHLIUM# : Identifier for Meshlium device.
  - #ID# : Unique identifier for data.
  - #ID_WASP# : Identifier for Waspmote.
  - #SENSOR# : Sensor identification.
- **Message template**: Data structure of MQTT message. The user can use the following substitution-expressions to create a customized message content:
  - #ID# : Unique identifier for data.
  - #ID_WASP# : Identifier for Waspmote.
  - #ID_WASP# : Secret identifier.
  - #SENSOR# : Sensor identification.
  - #VALUE# : Value obtained from the sensor.
  - #TS("c")# : Date with custom format. The parameter passed in this expression corresponds to those used by Java's SimpleDateFormat (see Date and Time Patterns in https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html.

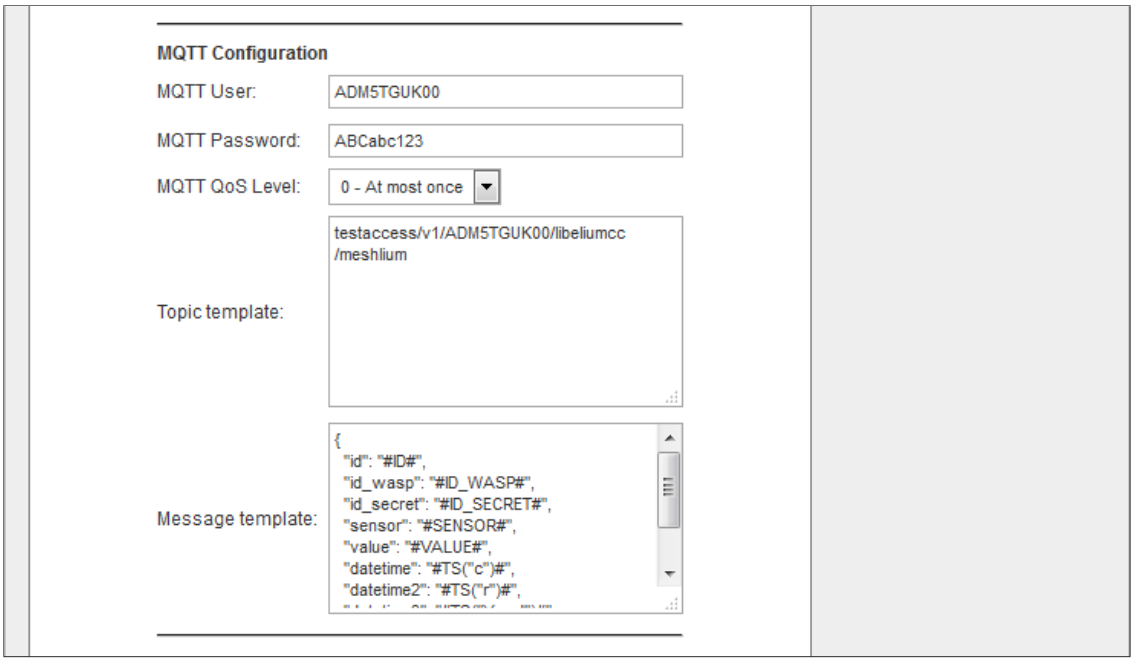


Figure : Meshlium Fujitsu IoT Connector - MQTT configuration parameters

Some parameters must be configured in the Fujitsu IoT Cloud Service (K5) platform portal, navigate to the following URL and login with a valid user:

https://iot-portal-1.uk-1.paas.cloud.global.fujitsu.com/LoginShow

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

MQTT User and MQTT Password

Navigate to the 'Common' tab, and select the MQTT. The user name and password for communication via the MQTT protocol will then be displayed in the following dialogue:

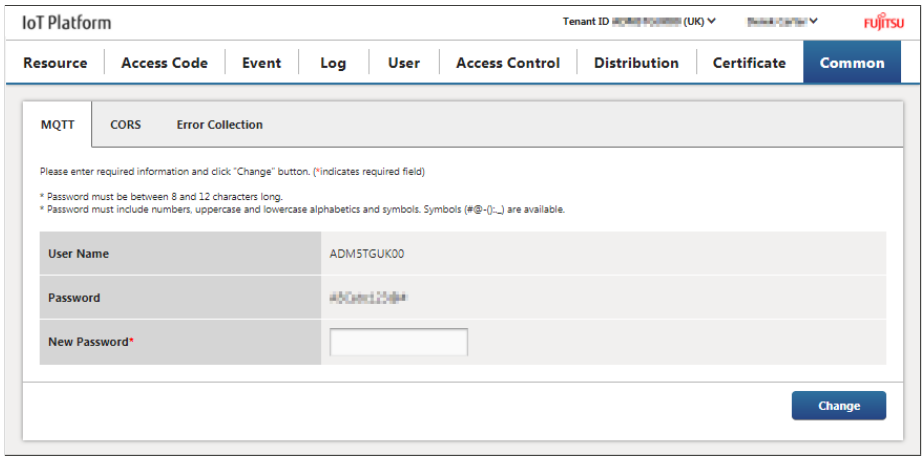


Figure : Fujitsu IoT Platform Portal - common dialogue, MQTT information

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

Topic Template

Navigate to the 'Resource' tab, and select the intended resource for the Meshlium's messages to view a Resource MQTT Topic:

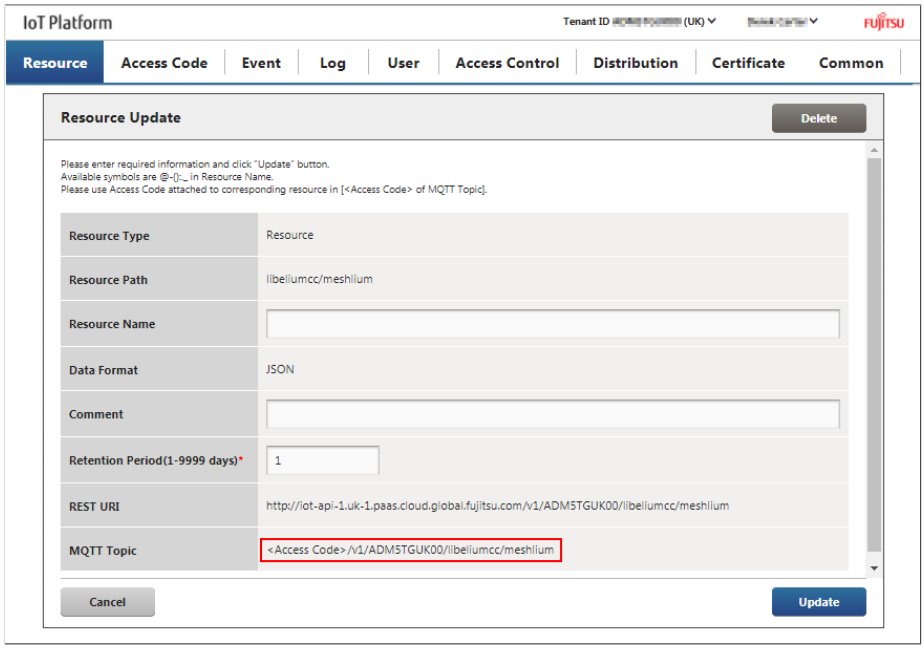


Figure : Fujitsu IoT Platform - resource information update dialogue, MQTT topic

It should be noted that the 'Access Code' used in association with the 'Resource' should replace the term '<Access Code>' when entered as a topic template on the Meshlium. In order for the MQTT messages to be correctly identified against a specific Fujitsu IoT Platform 'Resource', the topic template should be that of the MQTT Topic as identified on the Fujitsu IoT Cloud Service (K5) platform portal.

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

REST Configuration

The following configuration settings apply to the IoT Connector when the Transmission Type is set to REST:

- **Rest Resource**: The URL path of the Fujitsu IoT REST service. The Rest Resource is required to match that of the Fujitsu IoT Cloud Service (K5) platform portal 'Resource' in order for REST messages to be correctly processed.
- **JSON template**: JSON format data structure of REST message. The user can use the following substitution-expressions to create a customized message content:
  - #ID# : Unique identifier for data.
  - #ID_WASP# : Identifier for Waspmote.
  - #ID_WASP# : Secret identifier.
  - #SENSOR# : Sensor identification.
  - #VALUE# : Value obtained from the sensor.
  - #TS("c")# : Date with custom format. The parameter passed in this expression corresponds to those used by Java's SimpleDateFormat (see Date and Time Patterns in https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html).
- **Bearer Token**: Token given to authorize the client with the Cloud Service Access Code. The bearer token configured on the Meshlium is required to match the 'Access Code' associated with the Fujitsu IoT Service (K5) platform portal 'Resource'.
- **Bulk Message**: Identifies if bulk REST messages are used:
  - **YES**: Transmit multiple sensor data point messages in a single bulk REST request.
  - **NO**: Use an individual REST request for each sensor data point.

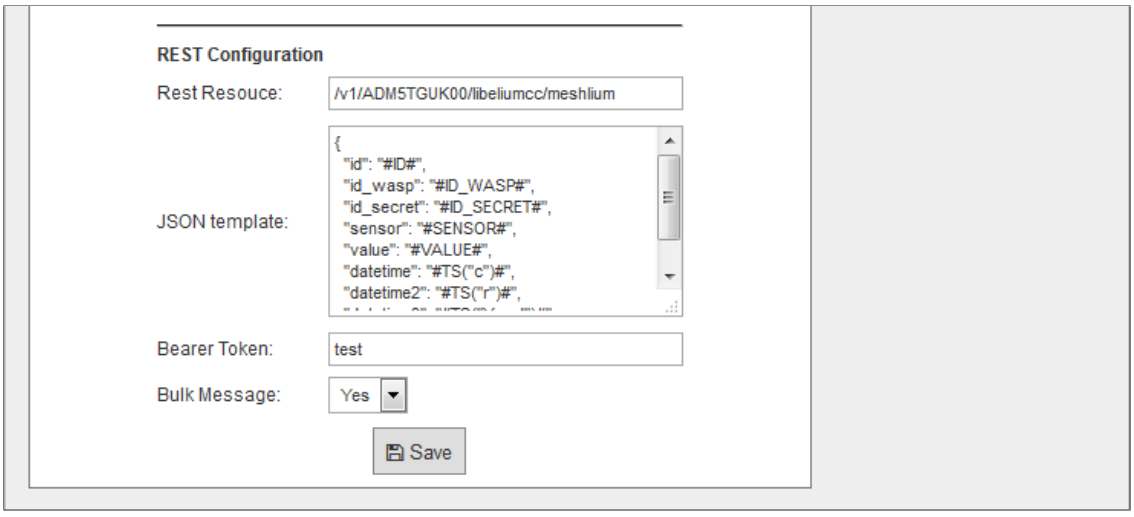


Figure : Meshlium Fujitsu IoT Connector - REST configuration parameters

Some parameters must be configured in the Fujitsu IoT Cloud Service (K5) platform portal, navigate to the following URL and login with a valid user:

https://iot-portal-1.uk-1.paas.cloud.global.fujitsu.com/LoginShow

Please refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

REST Resource

Navigate to the 'Resource' tab on the Fujitsu IoT Cloud Service (K5) platform portal, and select the intended resource for the Meshlium's messages to view the REST Resource to use:

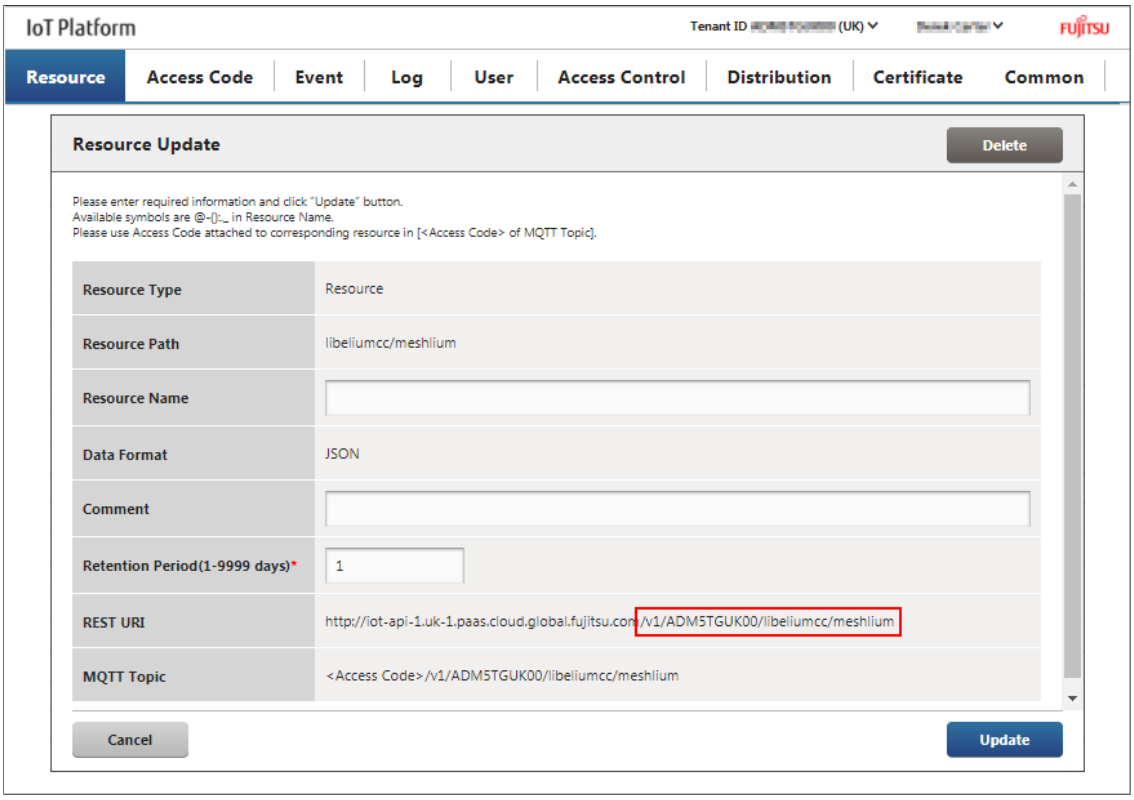


Figure : Fujitsu IoT Platform - resource information update dialogue, REST URI

It should be noted that Rest Resource is all the symbols following the hostname, including the initial '/' character.

Please, refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

Bearer Token'

Navigate to the 'Access Code' tab on the Fujitsu IoT Cloud Service (K5) platform portal, and select the 'Access Code' associated with the 'Resource' intended to receive messages from Meshlium:

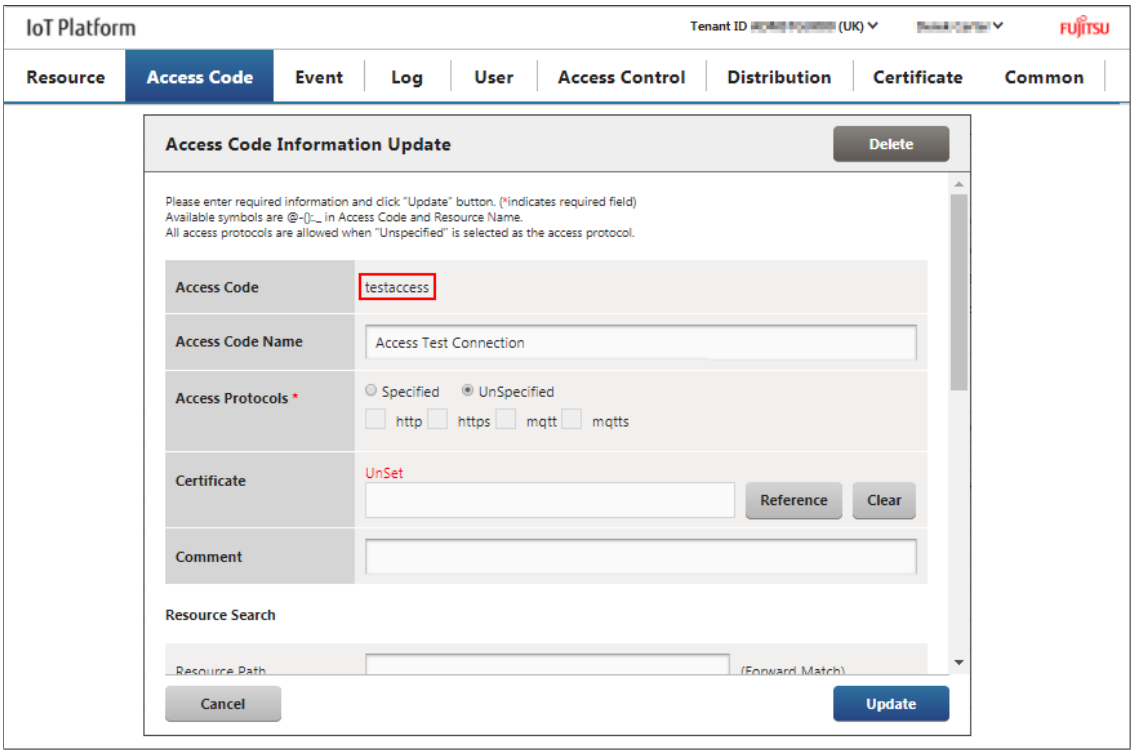


Figure : Fujitsu IoT Platform Portal - Access Code information update dialogue, Bearer Token

Please, refer to the Fujitsu IoT Cloud Service (K5) documentation for a detailed description.

Controlling status

Once the connector's parameters are configured, the user can launch the Fujitsu IoT Connector. The Service will then periodically poll for received frames on the Meshlium's local database, and send them to the Fujitsu IoT Cloud Server either via MQTT or REST depending upon the connector's configuration. The status indicator displays the current state, via a green or red symbol next to 'Fujitsu Cloud Connector'.

Figure : Meshlium Fujitsu IoT Connector - stopped

Once the Fujitsu Cloud Connector is running, it can be stopped by clicking the "Stop" button.

Figure : Meshlium Fujitsu IoT Connector - running

Viewing logs

In order to ensure the Fujitsu IoT Cloud Connector is correctly configured and identify issues, the user can select to view the most recent log output of the Cloud Connector.

At the top of the Fujitsu IoT Connector configuration settings page are tabs for “Configuration” and “Logs”. Selecting the Logs tab a snapshot of the log output of the Cloud Connector is shown. The view can be updated to show the most recent log messages by selecting the “Refresh” button. The user can also select to clear the current logs by selecting the “Delete” button.

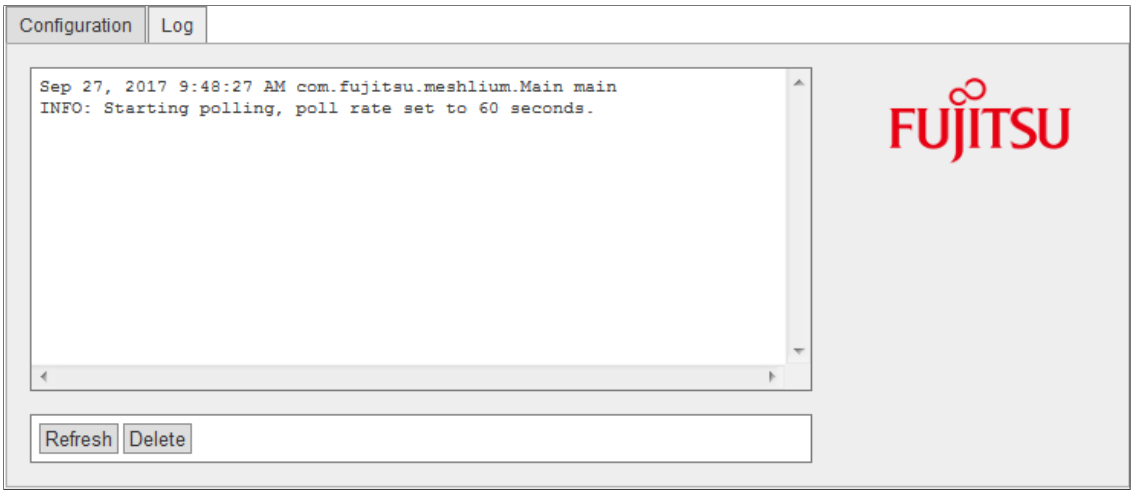


Figure : Fujitsu IoT Connector logs view

12.3.16. HaibuSmart

HaibuSmart is an IoT platform for business. The services include Enterprise Business Applications that involves near real-time information and the integration of sensors as primary data acquisition point.

For more information you can reach us at:

http://www.haibusmart.com

HaibuSmart Cloud

This IoT platform was design to be simple and scalable and Meshlium fits perfectly in HaibuSmart's solutions catalog including Agriculture and other associated services.

The following diagram shows the interaction between Meshlium and HaibuSmart.

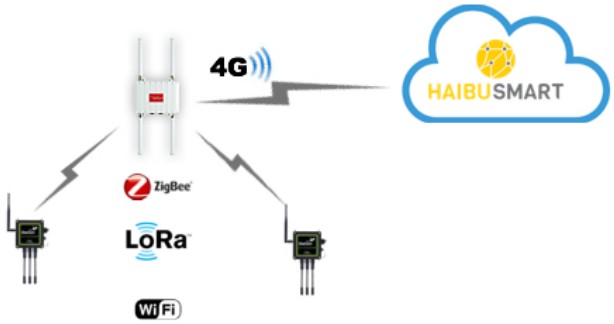


Figure : HaibuSmart architecture

Registering your Meshlium with HaibuSmart

Add your API KEY in the HaibuSmart panel of the Cloud Connector tab in the Manager System. Enter your API KEY, choose the Log Level and press "Save".

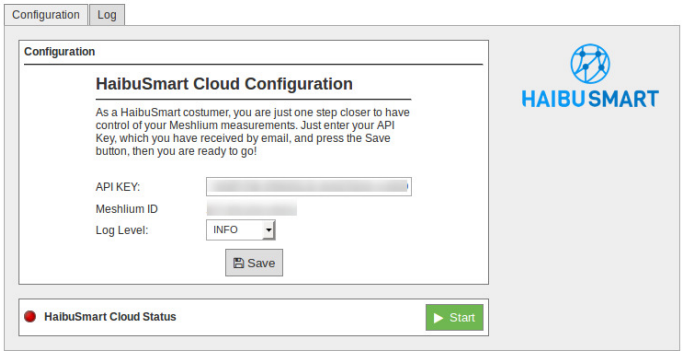


Figure : HaibuSmart Cloud configuration

Synchronization services

Pressing the "Start Button", the HaibuSmart Cloud Connector will be launched. Every time that Meshlium receives frames that are stored on the local database, the HaibuSmart Cloud Connector will send them to the HaibuSmart Cloud Platform. You can check if the service is running with the message on the left and the red (stopped) / green (running) indicators.

You can find further information on our website at http://www.haibusmart.com/meshlium/#/connector.

12.3.17. IoT-Ticket

IoT-Ticket is one of the world's most complete, advanced and easy to use Industrial Internet of Things platforms with over 1.6 million users mainly in the energy and mobile machinery industry. Using IoT-Ticket you can build IoT applications in your web-browser in minutes, no plug-ins required. You can create dashboards, reports, analytics or augmented reality based on big-data collected from your things.

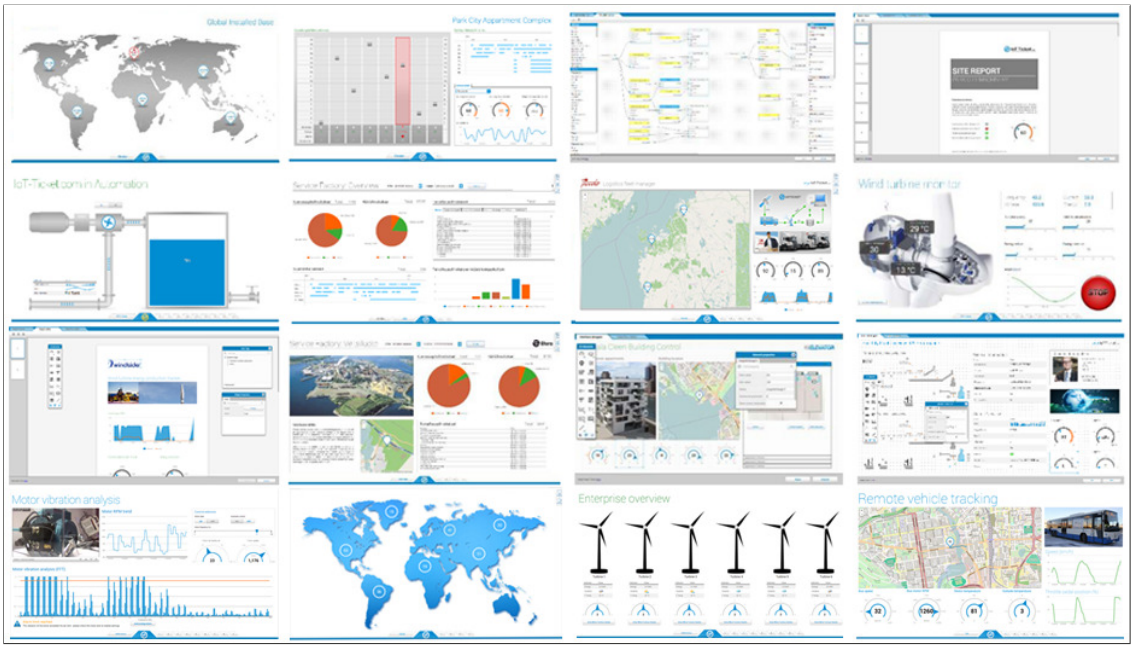


Figure : IoT-Ticket panel examples

Some benefits offered by IoT-Ticket:

- **Complete, up-to-date solution**: IoT-Ticket is a complete remote management system which includes the electronics, software and server. The platform is continuously developed further with new features and options.
- **Easy to get started and integrated**: You can use the platform as a service (SaaS or PaaS) or deploy to your own servers. We can integrate IoT-Ticket with any of your other information systems.
- **Flexibility and choice**: Use the whole IoT-Ticket platform or part of it. Use IoT-Ticket specific electronics or use your own, already deployed, electronics. Easy to use API in many programming languages allows a huge selection of devices to be easily connected.
- **Easy to use and Customizable**: The IoT-Ticket web dashboards allows you to be up and running in minutes using only your web browser. IoT-Ticket can be customized to meet your unique needs, even the look and feel can be made to match your corporate brand identity.

More information can be found at www.iot-ticket.com.

IoT-Ticket Meshlium integration

Once Libelium's Cloud Connector has been configured, all your available data will show up automatically in your IoT-Ticket web-based dashboard / report designer from where you can easily design Internet of Things applications.

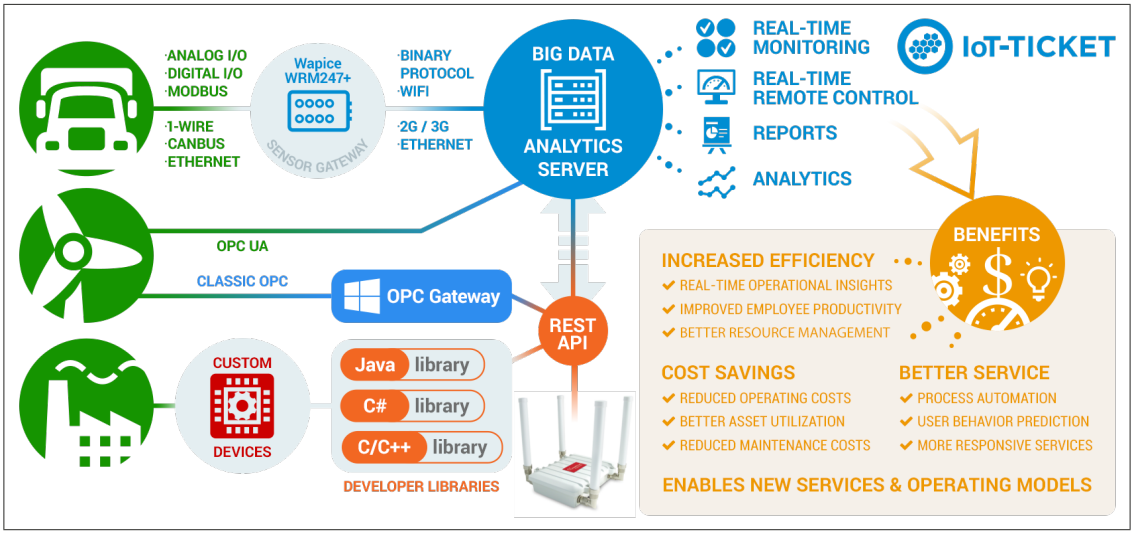


Figure : IoT-Ticket Meshlium integration

The IoT-Ticket cloud connector settings can be found under the IoT section of the browser-based Meshlium Manager System. The configuration is split into three parts Login Configuration, Connector Settings and Waspmote Filtering, as well as a section for information about the current status of the connector with controls to start and stop the program.

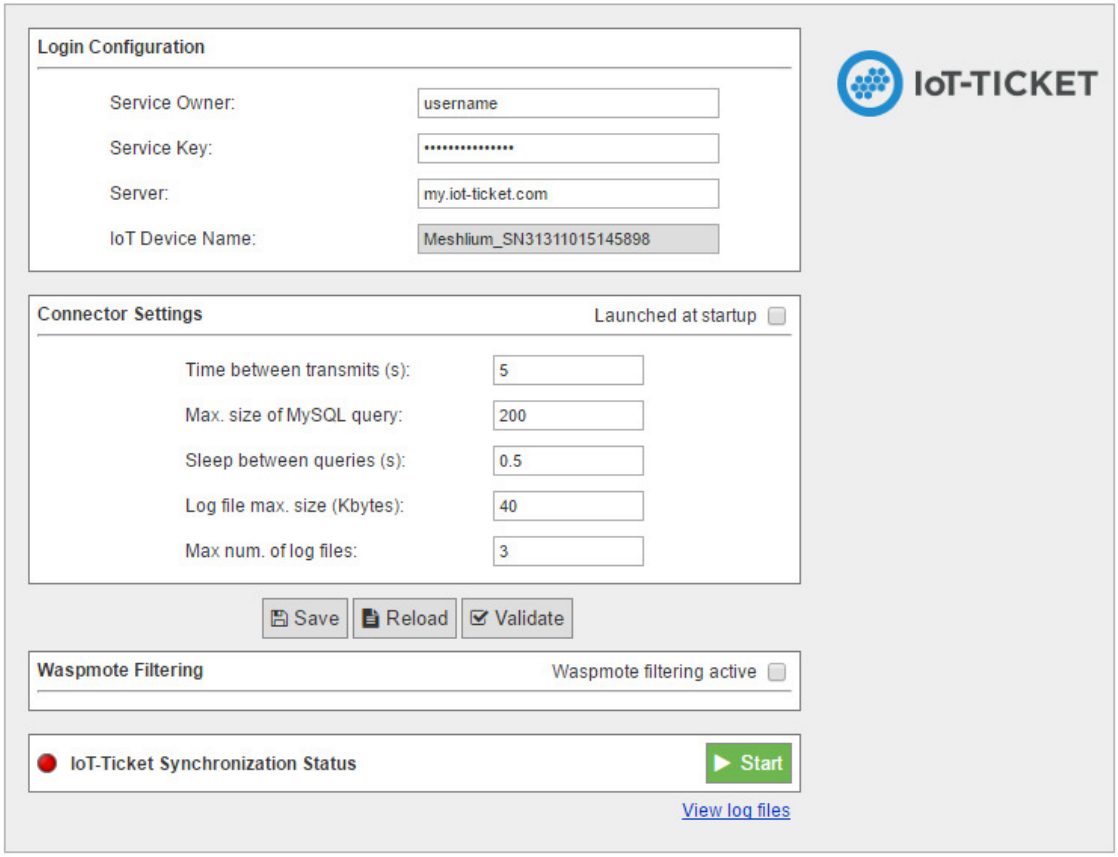


Figure : IoT-Ticket plugin panel

The Login Configuration section sets up the information for your IoT-Ticket account, and consists of four parts:

- **Service Owner/Key**: gives the username and password of the my.iot-ticket.com service account to which you wish to connect your Meshlium device.
- **Server**: specifies the IoT-Ticket server to use, by default **my.iot-ticket.com**.
- **IoT Device Name**: is a read-only field showing the IoT-Ticket device name used for that Meshlium unit. It is set when the connector is started and is empty if no name has yet been set (in such a case, use browser "Refresh" after connector has started to see the name).

The Connector Settings section has parameters for the operation of the connector itself. These values affect time between updates to IoT-Ticket as well as size of transmitted batches of data. More frequent data updates may come at the cost of increased system resource usage.

- **Launch at start**: This checkbox indicates whether the connector is set to start automatically when the Meshlium is powered on.
- **Time between transmits**: gives the minimum elapsed time between transmissions to IoT-Ticket. Values less than 60 seconds may consume high system resources.
- **Sleep between queries**: is the time the program sleeps between SQL queries, in order to conserve resources.
- **Max. size of SQL query**: is the maximum number of results for a single SQL query to the Meshlium database. Values greater than 200 may lead to high system load.

The section also allows for configuring connector logging:

- **Log file max. size (kbytes)**: The maximum size of a single log file in kilobytes.
- **Max. num. of log files**: is the number of log files that can be written before the logging handler begins overwriting the first.

Save, load and verify

These buttons allow saving, loading and validating entered configuration data to a local file on the Meshlium disk which is read by the connector. The saved data includes both the Login and Connector settings as well as any entered Waspmote filtering rule (see "Waspmotes" section below).

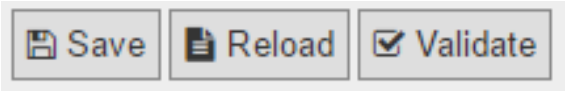


Figure : Save, Load and Verify buttons

- **Save**: validates the data entered into the form and saves to it to disk.
- **Reload**: reads data back from disk, erasing any fields that have been changed since last save.
- **Validate**: runs a check that entered fields are of the correct type and connects to IoT-Ticket to check the entered username and password. If verification fails for a field, it will be marked in red and an error message appears.

Validation of settings

Configuration settings are validated to make sure the entered data fields make sense (e.g. numeric fields such as sleep and query size must be numbers). Additionally, the validation will issue warnings if any parameters might cause high system load on Meshlium.

Additionally the "Validate" button checks entered login information with the IoT-Ticket server and shows a warning if settings are incorrect and a green confirmation message if they are correct.

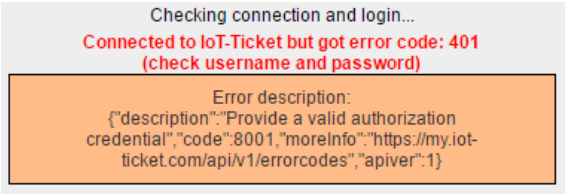


Figure : Validation error

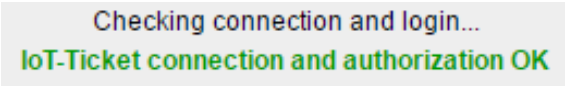


Figure : Validation success

Waspmote filtering

This section allows filtering of which Waspmote data is synchronized to IoT-Ticket. The section is enabled by ticking the Waspmote filtering active checkbox.

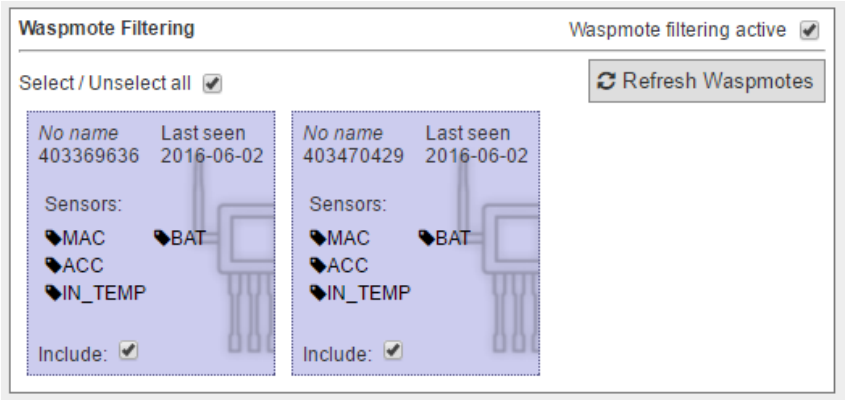


Figure : Waspmotes list

- **Refresh**: Waspmotes reloads the list of Waspmotes from the Meshlium database.
- **Select / Unselect**: All allows for quick selection or deselection of all present Waspmotes.

The Waspmote infobox contains the following values:

- **Name**: in the top-left shows the name a Waspmote broadcasts to the Meshlium with its readings (this is set in Waspmote code). If not set, No name is displayed instead.
- **Last Seen**: is the last date at which a sensor entry was sent from the Waspmote to the Meshlium.
- **Sensors**: is a list of sensors present on that Waspmote device. Only the latest detected set is displayed here, in case sensors are changed. Full names may be seen by hovering the mouse over the abbreviated names in the list.
- **Include**: specifies whether the Waspmote should be included in the data transmitted by the connector. Deselected Waspmotes have their info box greyed out.

Synchronization status

This section allows the user to start and stop the connector and displays information about its current status. When the connector is off, the indicator marker is red.

Figure : Start button

After clicking "Start", the connector shows a startup sequence, and when finished the running status will be indicated by the status icon turning green. The start button becomes a red "Stop" button.

Figure : Stop button

The link "View log files" will allow you to see the status of the running connector via its log files. A filtered set of this logging data will also be available as a data node in your IoT-Ticket enterprise.

IoT-Ticket view

Once the connector is running you can use your web browser to see the Meshlium data coming into your IoT-Ticket Dashboards and Enterprise Manager on my.iot-ticket.com.

In your IoT-Ticket enterprise the Meshlium device will be viewable as an IoT-Ticket device under your enterprise and can now easily be used in Enterprise Dashboards to create views of your incoming data, even mixing it with data coming from other IoT sources.

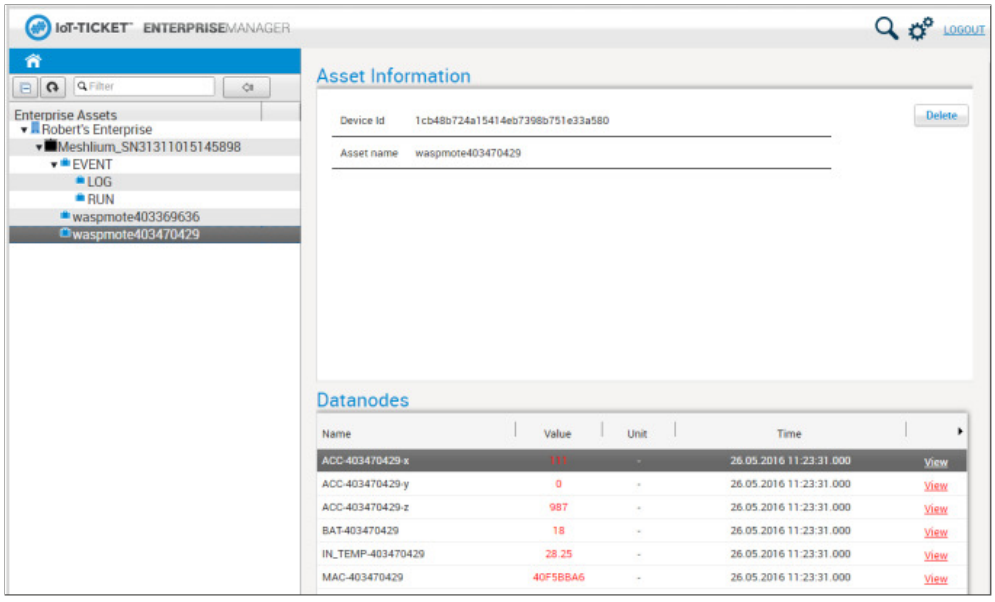


Figure : IoT-Ticket panel

You can now configure your own dashboard with sensor data and have it up and running in a matter of minutes.

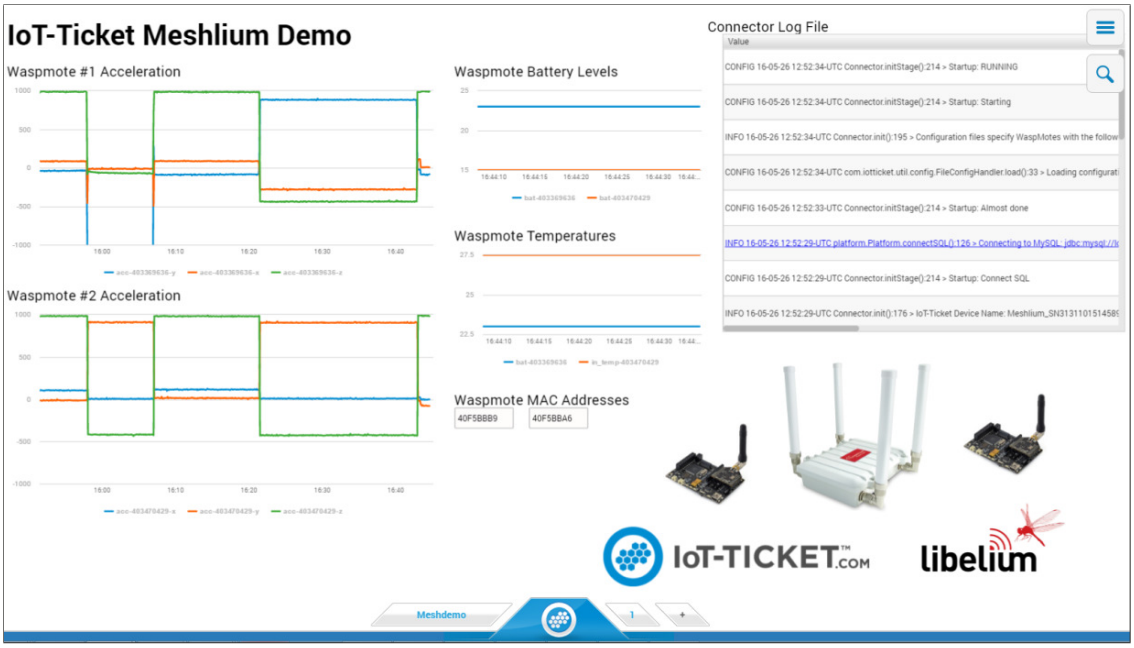


Figure : IoT-Ticket panel

12.3.18. IoTSens

IoTSens (http://www.iotsens.com/) is a horizontal platform for the development of smart cities which provides functionalities for gathering, integrating, storing and analyzing data from the city from a global point of view, so managers and citizens know what is happening and can immediately act.

IoTSens seamlessly integrates with Meshlium devices by means of MQTT queues so the connector will send all the sensors data to your IoTSens platform in order to be processed.

Configuration

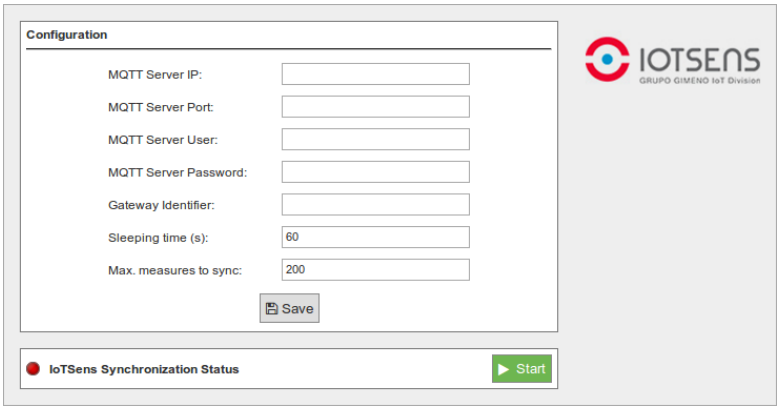


Figure : Configuring IoTSens in Meshlium

The IoTSens provider will supply you with the MQTT connection configuration attending your particular deployment:

- **MQTT Server IP**: IP address where the MQTT Server is deployed.
- **MQTT Server Port**: Port number where the MQTT Server is listening for connections.
- **MQTT Server User**: User name for connecting to the MQTT Server. This field can be empty if no user is required.
- **MQTT Server Password**: Password for connecting to the MQTT Server. This field can be empty if no user is required.

Additionally, the IoTSens plugin supports the configuration of some parameters regarding how the synchronization process works:

- **Sleeping time**: The synchronization process sleeps some time between executions. This parameter configures how many seconds it will sleep before starting the synchronization process again once it has finished. The sleeping time must be long enough to give time to other device processes to do their work.
- **Max. measures to sync**: This parameter configures how many sensor measures are synchronized at most in every synchronization process. The number of measures to synchronize must be limited in order to avoid the synchronization process to overload the system for a long time.

Controlling synchronization

You can start and stop the synchronization of the data to the IoTSens service. In the interface, you can see an indicator of whether the IoTSens service is running or not. If you click on "Start", the synchronization will begin.

Figure : IoTSens synchronization service is running

You can stop at any moment clicking on "Stop" button.

Figure : IoTSens synchronization service is stopped

12.3.19. Kii

Introduction

Kii Cloud is an MBaaS (Mobile Backend as a Service) and an IoT (Internet of Things) cloud platform provided by Kii Corporation.

Kii offers a cloud service that provides various server-side functions as versatile APIs for mobile apps and IoT solutions. By leveraging these APIs, the user can provide services making mobile apps and things Internet-ready without the server-side implementations and operations.

To learn more about the advantages of using the Kii Cloud see this page:

http://docs.kii.com/en/starts/merits

In order to make it easier for developers and system integrators to use these APIs, Kii offers a set of SDKs (http://docs.kii.com/en/references) for multiple platforms.

Internet of Things (IoT)

For IoT scenarios Kii offers the Thing-IF SDK (https://docs.kii.com/en/guides/thingifsdk). Thing Interaction Framework (Thing-IF https://docs.kii.com/en/starts/iot-functions/basic_function) is a framework positioned above the Kii Cloud SDK. It is a combination of selected features of the Kii Cloud SDK to accelerate IoT solutions.

Devices which are part of IoT solutions are called things in the Kii Cloud environment.

Connector basics

The Kii Cloud connector runs on Meshlium as an agent/daemon that periodically checks the local database for incoming sensor data (eg. frames coming from remote Waspmotes). The Meshlium unit itself is registered as a gateway against the Kii Cloud and the nodes sending sensor data are dynamically registered as end-nodes of the gateway. The sensor data itself provided by the nodes is sent to the Kii Cloud as Thing-IF states (http://docs.kii.com/en/functions/thingifsdk/thingifsdk/model/states) which reflect a node's sensor data snapshot at a specific point in time.

Once the nodes are sending states on the Kii Cloud you could for example browse the node's current and historical sensor values on a mobile application by using the Thing-IF SDK for Android (https://docs.kii.com/en/guides/thingifsdk/android) or iOS (https://docs.kii.com/en/guides/thingifsdk/ios).

Connector configuration and operation

You can configure the Kii Cloud connector in the Cloud Connector tab of the Meshlium Manager System. Click on Basic Cloud Partner and select Kii.

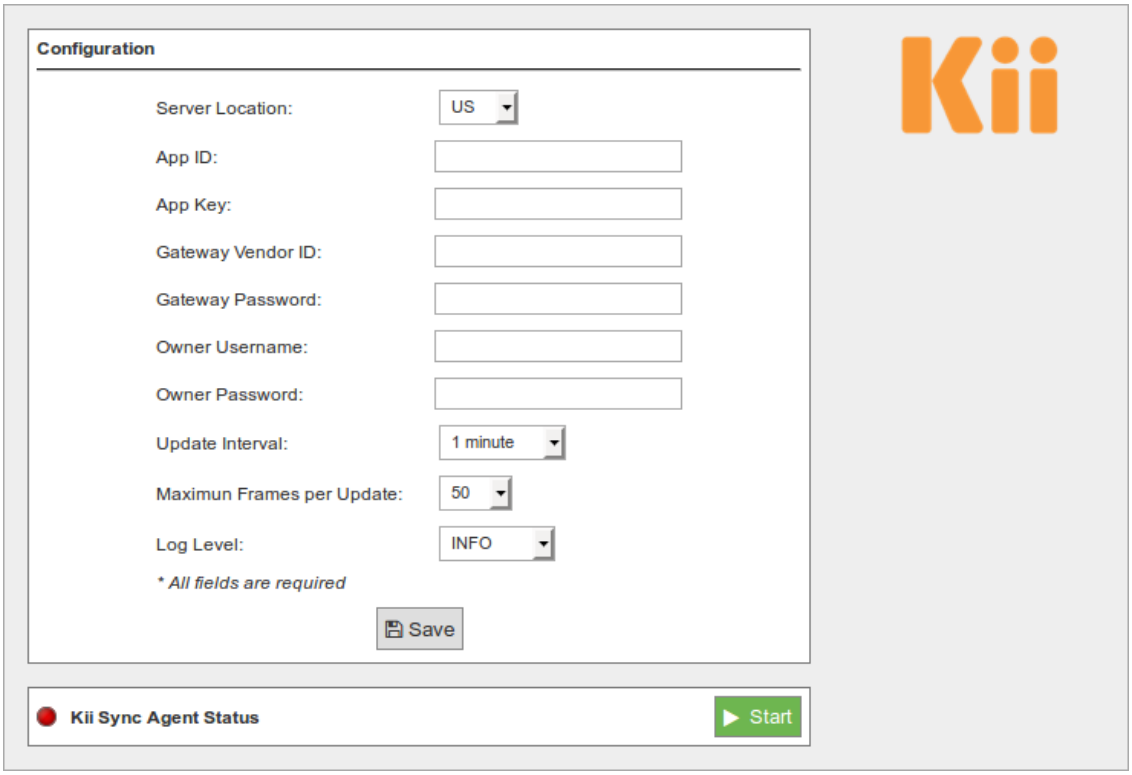


Figure : Kii Cloud Connector

Then fill in the fields as follows:

- **Server Location**: Enter server location of your app (either US, JP, EU, CN3 or SG) created at developer.kii.com. For more info on creating an app see this page.
- **App ID**: Enter the App Id of your app created at developer.kii.com. For more info on creating an app see this page.
- **App Key**: Enter the App Key of your app created at developer.kii.com. For more info on creating an app see this page.
- **Gateway Vendor ID**: Enter a unique name for this gateway (Meshlium). It will be created if it does not exist on the Kii cloud.
- **Gateway Password**: Enter a password for this gateway (Meshlium). If the gateway already exists on the Kii cloud, the password must match the previous registration.
- **Owner Username**: Enter a username to define which Kii user will own the gateway and nodes. It will be created on the Kii cloud if it does not exist.
- **Owner Password**: Enter a password for the user you specified above. If the user already exists on the Kii cloud, the password must match the previous registration.
- **Update Interval**: Enter the frequency in which the Kii agent queries the local database for new sensor data (frames). Minimum and default is 60 seconds.
- **Maximum Frames per Update**: Enter the maximum number of frames to fetch from the local database per update cycle. Default is 50, maximum is 200. Only frames not previously synchronized will be fetched. As a general rule of thumb we advise you at least allow for 1 second per frame in order to allow the daemon to keep up (eg. if you choose 200 frames, select an interval of at least 200 seconds).
- **Log Level**: Enter the log level. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG, REPORT. Default is OFF.

You can start and stop the service by using the “Start/Stop” button in the Manager System plugin:

Figure : Configuring Kii in Meshlium

Figure : Configuring Kii in Meshlium

How to verify that the Kii Cloud Connector is working properly

You can manually verify that the Kii Cloud Connector is working properly doing a visual inspection of the devices and sent data at our developer console. These are the steps:

- Go to developer.kii.com and sign in with the same credentials you used to create the app as described in the previous section of this document.
- Select the app you created before. You will see a set of icons representing the different services provided by Kii.
- Click on the Things icon (the cube), then click on the Console tab and make sure you can see the Meshlium (and the nodes that you tested to submit data) in the list.
- Click on the Objects icon (the cylinder), then click on the Data Browser tab. In the combo box select Normal bucket and then click on Application scope. Now click on states. Make sure you see a row that when expanded shows the data from one frame (this is fine, you can see them all here). Make sure the version column shows a number equal to the number of frames you sent (if you click on Refresh on the top left, this number should increase on each interval time by the number of maximum frames that you configured on the connector configuration page).
- Click again on the Objects icon (the cylinder), then click on the Data Browser tab. In the combo box select Time series bucket and then click on Thing scope. In the second combo box select Vendor Thing ID and in the text box below, type the name of the node you want to verify. A pop-up box will appear with the node and you must click on it. Now click on ts_history and you will see a list of sensor entries for that node. Click on each of them to verify the sensor data.

12.3.20. Labeeb

Labeeb IoT is a Cloud and On-Premise Internet of Things (IoT) Services Enablement Platform. It provides an attractive environment (i.e. platform-as-a-service, open APIs, development tools, and documentation) for entrepreneurs, third party developers and companies to accelerate the development and deployment of new IoT services, and to help them achieve faster time-to-market.

For more information: http://www.labeeb-iot.com.

Figure : Labeeb IoT plugin

Register Meshlium in Labeeb IoT Platform

Create a Labeeb IoT account previously to register Meshlium in Labeeb IoT Platform:

http://mea.labeeb-iot.com

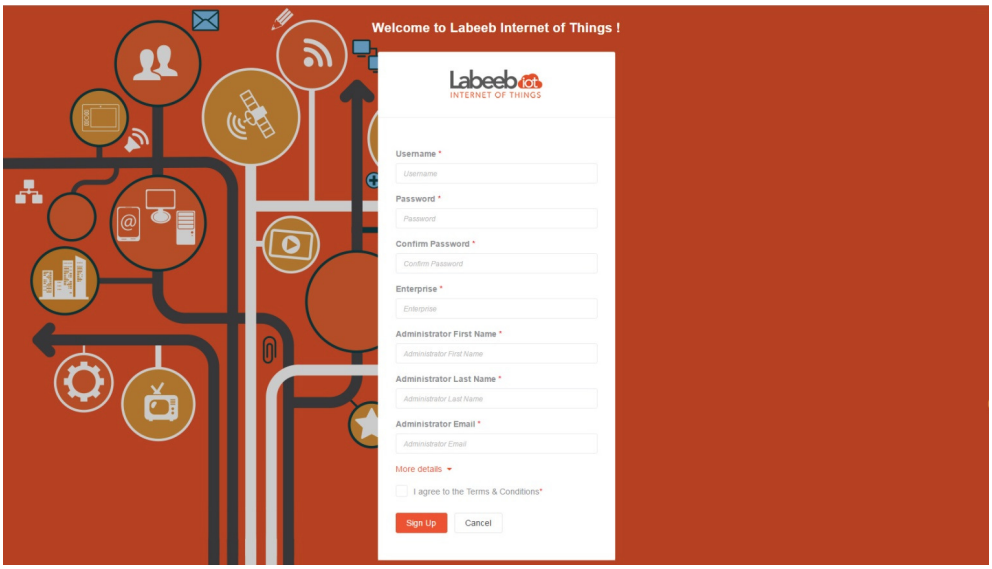


Figure : Creating a new Labeeb account

Configuration

After creating your account, you can use your credentials (enterprise name, username and password) to configure Meshlium.

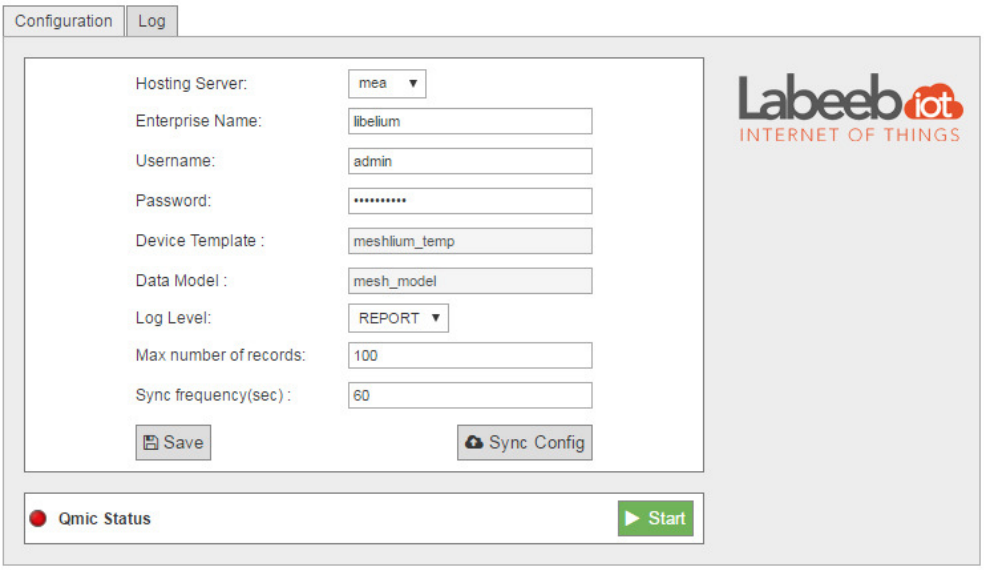


Figure : Labeeb IoT configuration panel

1. Fill the configuration fields with your Labeeb IoT credentials:
   - **Hosting Server**: the server where your Labeeb IoT account is hosted in (default: MEA for mea.labeeb-iot.com).
   - **Enterprise Name**: enterprise name used to create portal account.
   - **Username**: username of your portal account.
   - **Password**: password of your portal account.
   - **Log level**: used for debugging level, errors, reports, etc.
   - **Max number of records**: The maximum number of records to be synchronized to Labeeb IoT every some time interval.
   - **Sync frequency (sec)**: specific time interval to perform synchronization, defined in seconds.
2. Save the configuration by pressing the button “Saves”. This will store them locally and create all needed devices and sensors automatically on Labeeb IoT platform.
3. Whenever you make a change, add a sensor or a device, press on “Sync Config” to update these parameters on Labeeb IoT portal.

Controlling synchronization

Once the connector is configured, the user can launch the Meshlium Labeeb IoT script ("Start" button). The program will search for the received frames on the local database, and will send them to the Labeeb IoT Platform. The status indicator displays the current state: "Running" or "Stopped".

The Meshlium Labeeb IoT cloud connector will start sending the data of any new Waspmote device to Labeeb IoT Platform after a maximum of 60 min.

Figure : Labeeb IoT sender is running

You can stop the Labeeb IoT script anytime by clicking on the "Stop" button.

Figure : Labeeb IoT sender is stopped

You can get all the data sent from the Waspmote devices on the Labeeb IoT portal:

Click on Data > Data retrieval > select the related device or data type to retrieve collected data.

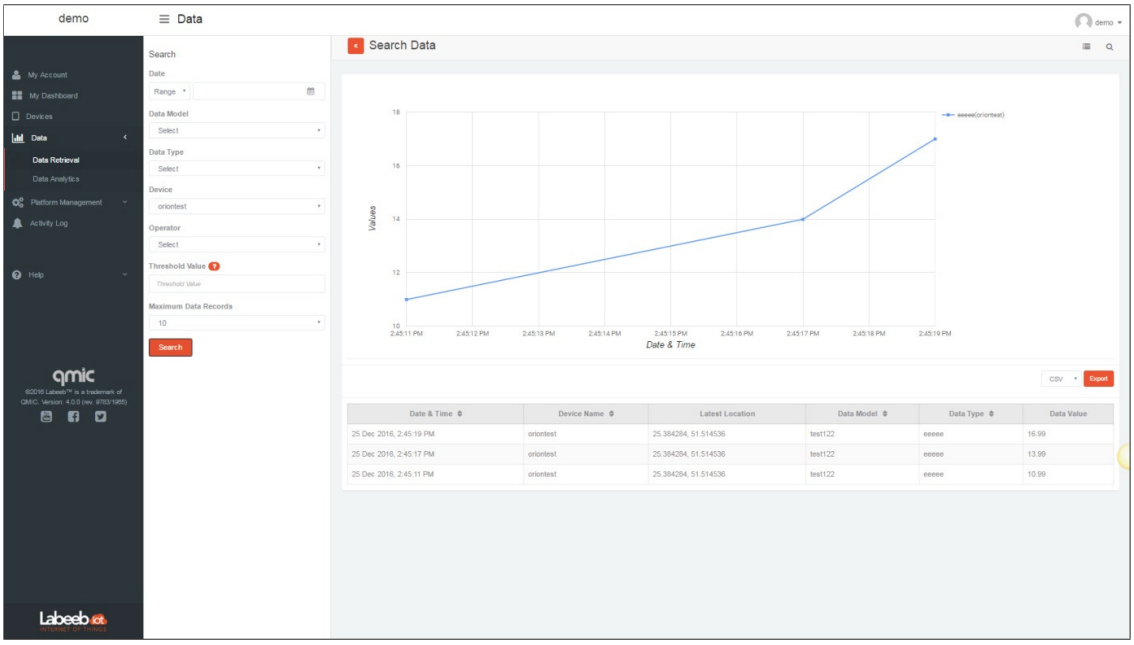


Figure : Labeeb IoT Platform data retrieval

12.3.21. MQTT

MQTT is a publish/subscribe, extremely simple and lightweight messaging protocol, designed by IBM for constrained devices and low-bandwidth, high-latency or unreliable networks, where battery power is critical. Due to its features of delivery assurance and bandwidth reduction, MQTT is being used by some Cloud platforms such as IBM or Carriots, which means that Waspmote data can be stored inside them or in any other one based on this protocol.

More information: http://mqtt.org/faq.

With this plugin, Waspmote sensor data can be directly integrated with a MQTT broker.

Figure : MQTT plugin

Configuration

The broker is a key agent in MQTT protocol. The broker is a server which receives all the frames and distributes each one of them to the subscribers clients.

In Server/Broker Configuration, the user can set:

- **IP Address**: Server IP address.
- **Port number**: Server port number.
- **User**: Server user name to log in the MQTT system.
- **Password**: Server password to log in the MQTT server.
- **Topic template**: Topic of your message. The user can use these wild-cards creating a personalized structure:
  - #MESHLIUM#: Identifier for Meshlium.
  - #ID#: Unique identifier for data.
  - #ID_WASP#: Identifier for Waspmote.
  - #SENSOR#: Sensor identification.
- **Message template**: Data structure of your message. The user can use these wild-cards creating a customized content:
  - #ID#: Unique identifier for data.
  - #ID_WASP#: Identifies the Waspmote unit.
  - #ID_SECRET#: Secret identifier.
  - #SENSOR#: Identifies the sensor.
  - #VALUE#: Value obtained from the sensor.
  - #TS("c")#: Date with custom format. The parameter passed in this wild-card corresponds to the same ones you can use in PHP date function (see format parameters in http://php.net/manual/es/function.date.php#refsect1-function.date-parameters).

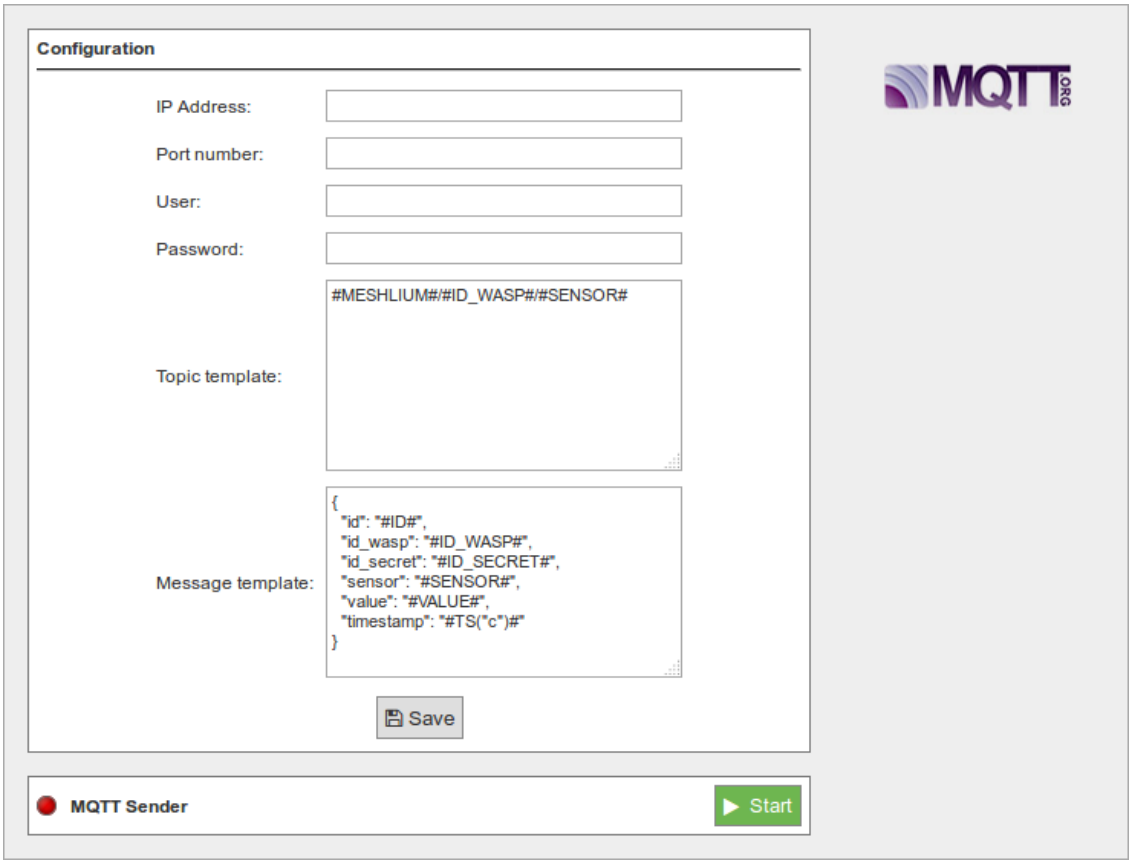


Figure : Server/Broker Configuration

Examples about MQTT Servers/Brokers:

- http://mqtt.org/wiki/doku.php/brokers.
- http://mosquitto.org/.
- http://mqtt.io/.

***Note**: in this example, the broker was running on a computer inside our local network for test purposes only. For professional use, it is recommended to work with a 24/7 server with static IP address.*

Controlling status

Once configured the server/broker, the user can launch the Meshlium MQTT program (Start button). The program will search for the received frames on the local database, and will send them to the broker via MQTT protocol. The status indicator displays the current state, saying “Running” or “Stopped”.

Figure : MQTT sender is running

You can stop the MQTT sender anytime clicking on the “Stop” button.

Figure : MQTT sender is stopped

Platforms using MQTT

MQTT has been widely implemented across a variety of industries. As of March 2013, MQTT is in the process of undergoing standardization at OASIS protocol stack. The protocol specification has been openly published with a royalty-free license for many years, and companies such as Eurotech (formerly known as Arcom) have implemented the protocol in their products.

Here are a number of notable projects that have made use of MQTT and related technologies. Companies like Cisco, Eclipse Foundation, Eurotech, IBM, Kaazing, M2Mi, Red Hat, Software AG, TIBCO and Carriots, among other companies, are working with this protocol.

More information about examples and uses: http://mqtt.org/wiki/doku.php/example_uses.

12.3.22. NEC Connexive

Connexive is a cloud platform developed by NEC that gives a scope of services to use the cloud.

Configuration

The Connexive cloud connector is located in:

Cloud Connector → Basic Cloud Partner → Connexive

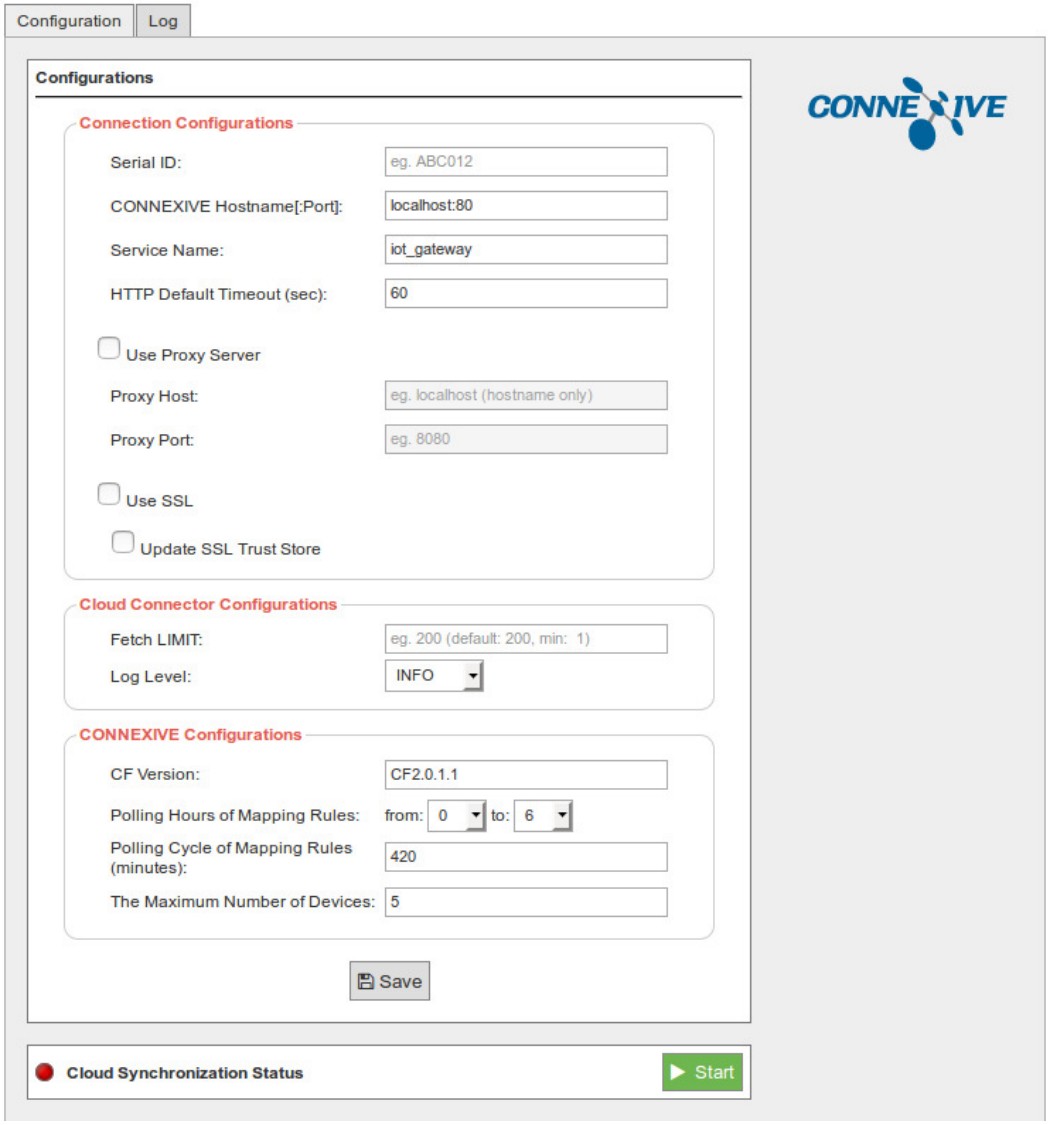


Figure : Connexive synchronization service configuration panel

The “Connection Configurations” section is the general configuration for the Connexive cloud connector.

- **Serial ID**: ’IoT-GW ID’ or the unique ID for the Meshlium which was set in Connexive when registering IoT-GW.
- **Connexive Hostname[:Port]**: Connexive hostname and port number.
- **Service Name**: service name to access Connexive. Default is “iot_gateway”.
- **HTTP Default Timeout (sec)**: default timeout of the reply for an HTTP transmission (in seconds units).
- **Use Proxy Server**: turn on this button if you are using a proxy server.
- **Proxy Host** : IP address or host name of the proxy server.
- **Proxy Port** : port number of the proxy server.
- **Use SSL**: turn this button on if using SSL.
- **Update SSL Trust Store** : turn this button on when updating the SSL trust store file.
- **Trust Store File**: trust store file for SSL.
- **Trust Store Password**: password for the trust store file.

The “Cloud Connector Configurations” section sets up the synchronization characteristics:

- **Fetch LIMIT**: the maximum number of results for a single SQL query to the Meshlium database. Values greater than 200 may lead to high system load.
- **log Level**: log level.

The “Connexive Configurations” section is the specific configuration for the Connexive cloud:

- **CF Version**: configuration version to connect with Connexive. Please always input the fixed value “CF2.0.1.1”.
- **Polling Hours of Mapping Rules**: the time period to poll, in order to acquire mapping rules. For example, in the case of “1-3”, it polls Connexive from 1:00 to 3:59.
- **Polling Cycle of Mapping Rules (minutes)**: the interval which is polled to acquire mapping rules.
- **The Maximum Number of Devices**: the maximum number of devices.

Click on the “Save” button for storing the configuration fields.

Controlling synchronization

Once configured the cloud connector, press the “Start” button to start the synchronization service.

Figure : Connexive synchronization service is running

You can stop the synchronization service anytime clicking on the “Stop” button.

Figure : Connexive synchronization service is stopped

12.3.23. Orchestra

The Orchestra platform IoT enables the possibility to collect all the data from sensors attached to Meshlium over MQTT and manage them in a user-friendly dashboard.

How to get your own API-key

For getting your own API-key you have to send a mail to services@orchestra.it with the subject "[Libelium - Activation] New activation request", signaling the number of sensors and the numbers of venues or Meshliums being used, and Orchestra will provide the account to access the platform where you can get the needed information.

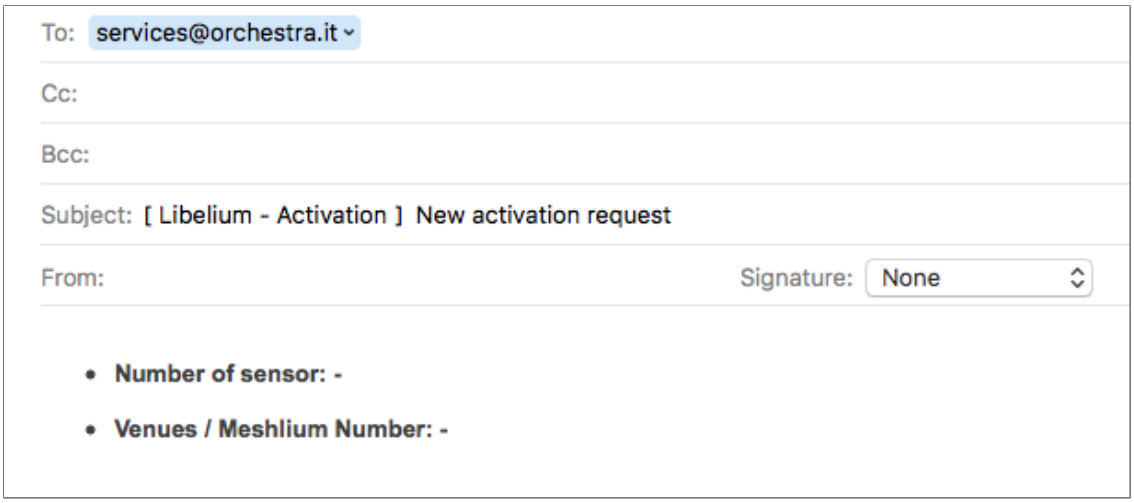


Figure : Configuring Orchestra

Below there are a couple of images showing where you will find the API information on the Orchestra website.

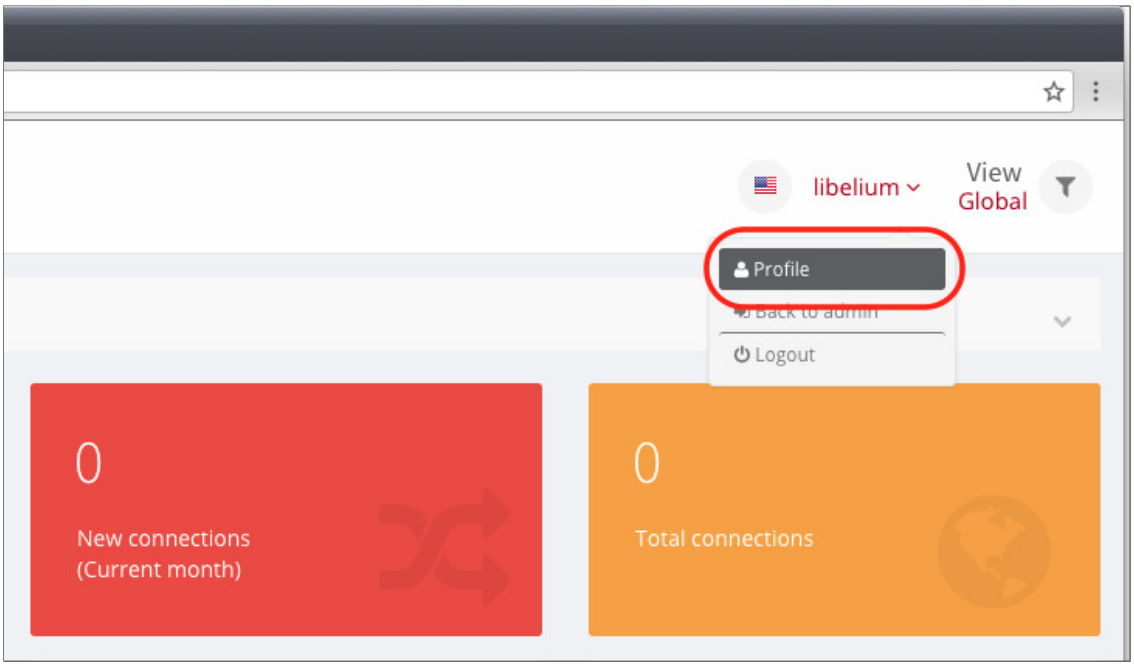


Figure : Orchestra platform user panel

To activate the API secret key and get the client Id, you have to go to the “Profile” section of the Orchestra account and then select the checkbox in the “Orchestra API” section and click the “Generate key” button.

Then you have to save the information generated by clicking on the “Save” button in the same section.

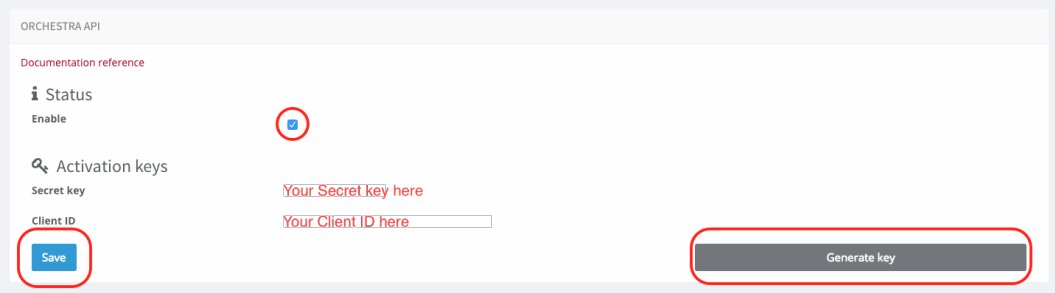


Figure : Orchestra platform API panel

Configuration

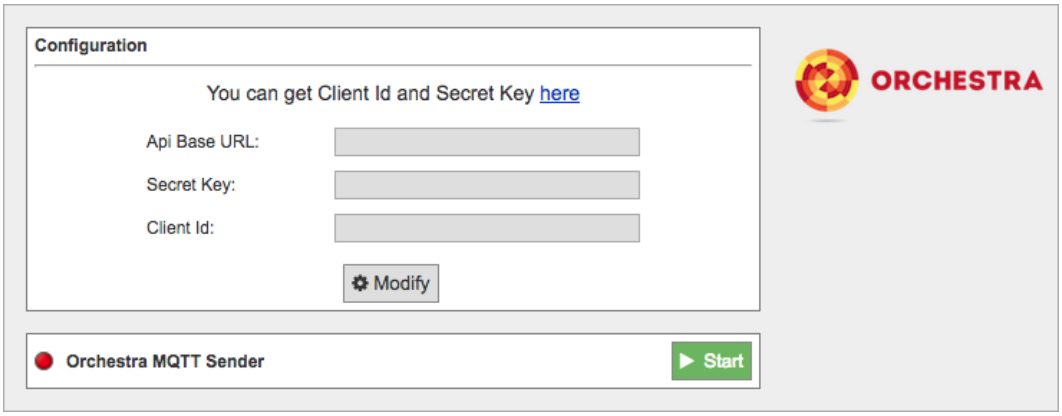


Figure : Orchestra Cloud Connector configuration panel

- **API Base URL**: You must enter the API URL that you are going to use for collecting the Meshlium data.
- **Secret Key**: The secret key to access the API.
- **Client Id**: Security key used for validating the access to the Host.

Click on the “**Modify**” button to enable the fields and insert the values previously obtained.

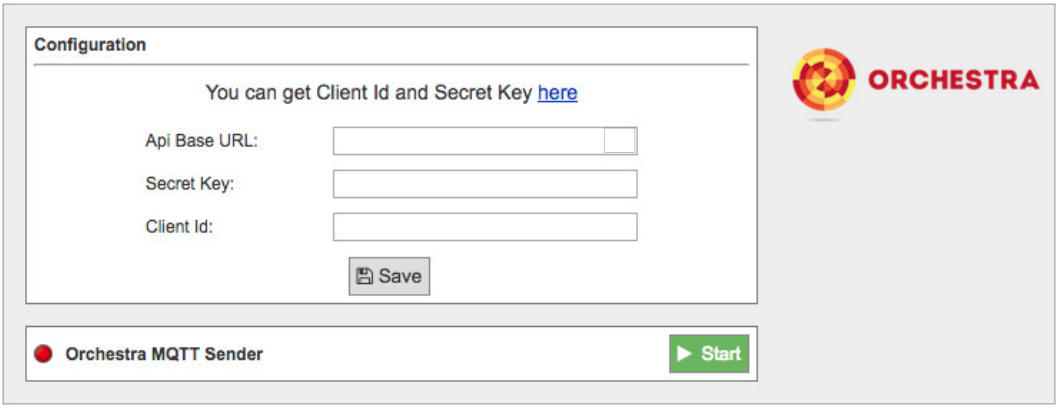


Figure : Orchestra Cloud Connector configuration panel

Once you click the "**Save**" button, a select box will appear with the venues you can choose.

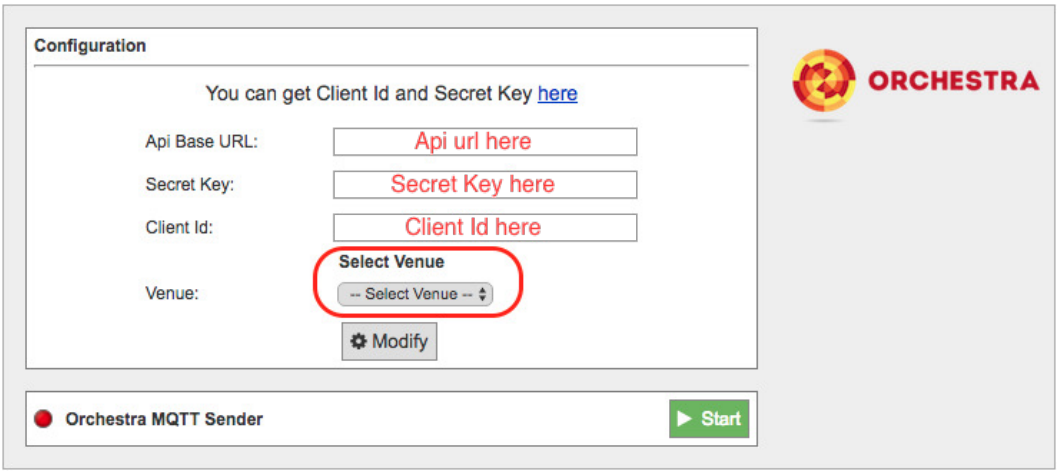


Figure : Orchestra Cloud Connector configuration panel

After that, press the "**Start**" button, and you will start to receive data from the configured Meshlium.

If you want to stop the event sending, just press the "**Stop**" button.

12.3.24. Redd

Redd is a company focused on delivering telemetry and telecontrol solutions for clients assets, with the aim of optimizing and simplifying operations, and most importantly, providing them with the necessary information to make quick and timely decisions.

For more information, please contact iot@reddsystem.com.

Configuration

You can locate the Redd Cloud Connector at:

Cloud Connector → Basic Cloud Partner → Redd Cloud Connector

Inside the configuration panel you can fill 2 parameters with the information provided by Redd. These are:

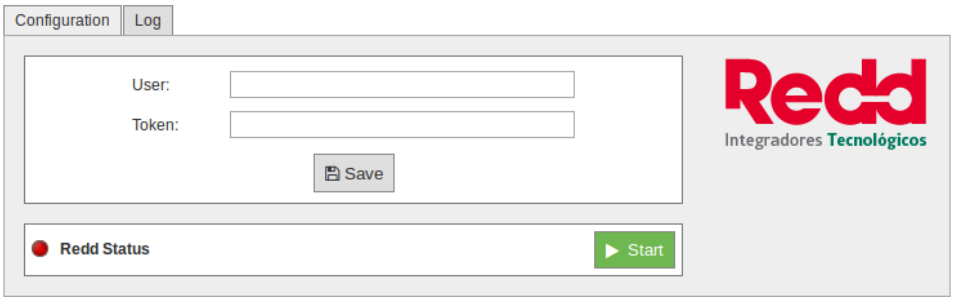


Figure : Redd Cloud Connector configuration panel

- **User**: this field recognizes the information sent to Redd by you.
- **Token**: unique identifier that enables access of your information to Redd.

Click on the "Save" button for storing the configuration fields.

The tab "log" will allow you to see the status of the running connector via its log files.

Controlling synchronization

Once configured the Redd Cloud Connector, you can launch the sync script by pushing the "Start" button. The synchronization will search for the received frames on the local database, and will send them to the Redd IoT platform via TCP protocol.

Figure : Redd Cloud Connector synchronization service in standby

After clicking on the "Start" button, it becomes a red "Stop" button. Click again and you will stop the synchronization.

Figure : Redd Cloud Connector synchronization service running

12.3.25. RIOT Platform

RIOT is a Sensing as a Service Platform developed by REDtone IOT. It manages connectivity from sensors or data sources with various communication protocols including MQTT and RESTful.

Please visit http://riot.com.my/ for more information.

Configuration

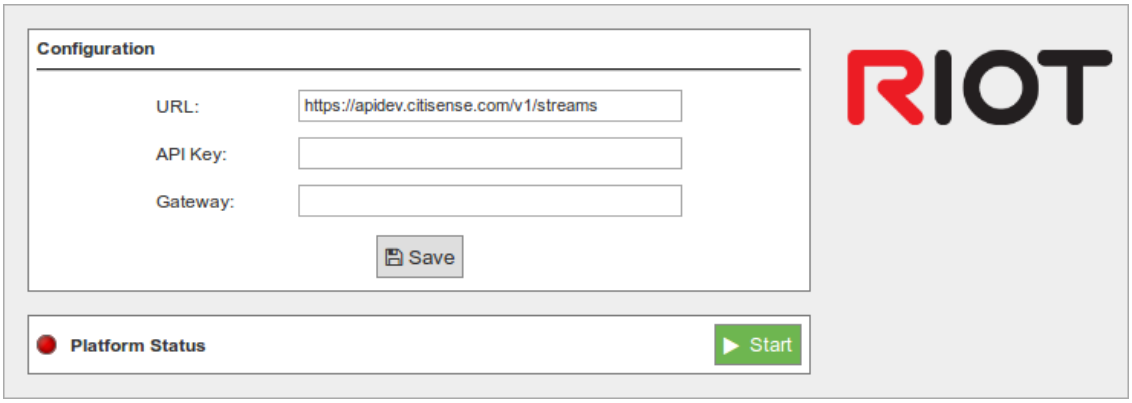


Figure : RIOT plugin Configuration

The RIOT plugin is configured with the following parameters:

- **URL**: The IP address or the URL will be provided by REDtone IOT.
- **API Key**: An API Key will be provided by REDtone IOT as one of the authorization information in order to send data to the RIOT platform.
- **Gateway**: Field to identify which Meshlium device the data came from when the platform receives the data, so that we can trace the location.

***Note**: In the Waspmote sensor board, make sure you identify each board with their own ID so that we can trace which sensor board the data originated.*

Controlling synchronization

The synchronization will be executed for all the data that has not been synchronized in the Sensor Parser database table. You can start and stop the synchronization process. In the interface of our service, you can see if the status of our service is either running or not. If you click on "Start", the synchronization will start.

Figure : RIOT sender is running

If you want to stop the synchronization process, you can simply just click on "Stop" and the process will stop.

Figure : RIOT sender is stopped

12.3.26. RoboMQ

RoboMQ is a Hybrid Integration Platform (HIP) that can connect any device, sensor, IoT gateway, enterprise application, or any cloud to allow you to build business workflows across networks and clouds. What makes RoboMQ a great choice for your IoT needs is that your IoT devices or applications can easily integrate with and leverage your IoT data by making it part of business workflow involving cloud, SaaS and on-premise enterprise systems. RoboMQ platform is built with containerized microservice architecture that can easily scale vertically and horizontally and runs in a distributed multi hybrid cloud setup.

To learn more about the benefits of integration with RoboMQ, visit www.robomq.io or contact sales@robomq.io.

Get a trial or paid subscription to RoboMQ iPaaS

To use this connector, the user first must get a trial account or paid subscription.

Configure RoboMQ connector

Once you have your RoboMQ subscription credentials, fill out the following configuration fields for the RoboMQ connector.

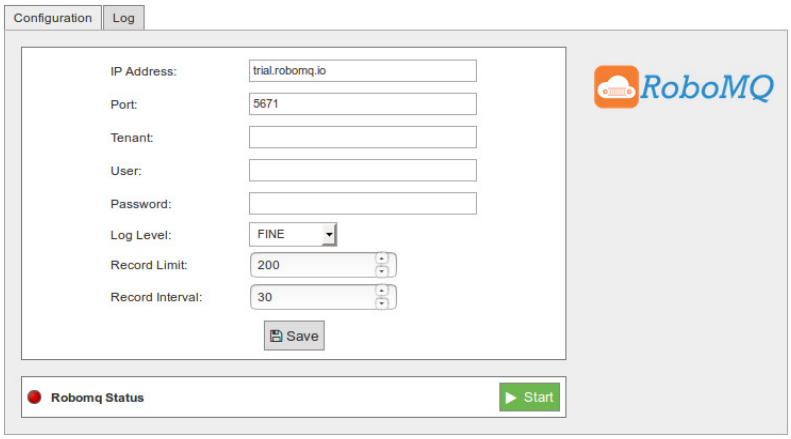


Figure : RoboMQ connector configuration panel

- **Host**: RoboMQ message broker URL. For free trial accounts use “trial.robomq.io”.
- **Port**: RoboMQ Message broker port for AMQP protocol. It supports SSL encrypted traffic, which is recommended. Use port 5672 for non-SSL and port 5671 for SSL traffic respectively.
- **Tenant**: Tenant is the Tenant ID or your account with RoboMQ iPaaS that is provided in the provisioning email.
- **Username**: username that is provided in the provisioning email.
- **Password**: password that is provided in the provisioning email.
- **Log Level**: Log Level that is set for the connector with the following settings.
  - OFF: No new logs will be printed to the log tab on Meshlium gateway.
  - FINE: The most verbose logging level. All logging statements are visible in the logging window.
  - INFO: Very verbose logging level but omits extremely detailed logs.
  - WARNING: Only shows warning and severe logs to the user.
  - SEVERE: Only shows severe logs.
- **Record Query Limit**: The number of records that is queried at every record query interval. The recommended maximum number of records to be queried is 200. A value greater than 200 is accepted but can affect the performance of the Meshlium gateway.
- **Record Query Interval**: The number of seconds that will elapsed in between each record query. The recommended interval between record queries is 60 seconds any record query interval below 30 seconds will not be accepted since this could affect the performance of the Meshlium gateway.

Connector operation and status panel

Once you have configured the connector, you can launch the Meshlium's RoboMQ connector and start sending data to RoboMQ by clicking the green "Start" button. The following image is displayed after you click the "Start" button.

Figure : RoboMQ Connector is running

To stop the connector, press the "Stop" button and wait for a few seconds for the connector to fully stop. After you do this, a "Start" button should appear once again.

Figure : RoboMQ Connector has stopped

Log tab

Once you start the connector you can view logs for this connector in the log window. The following is the illustration of the log window after the connector was started. You can refresh these logs by clicking the "Refresh" button. You can clear the logs by clicking the "Delete" button.

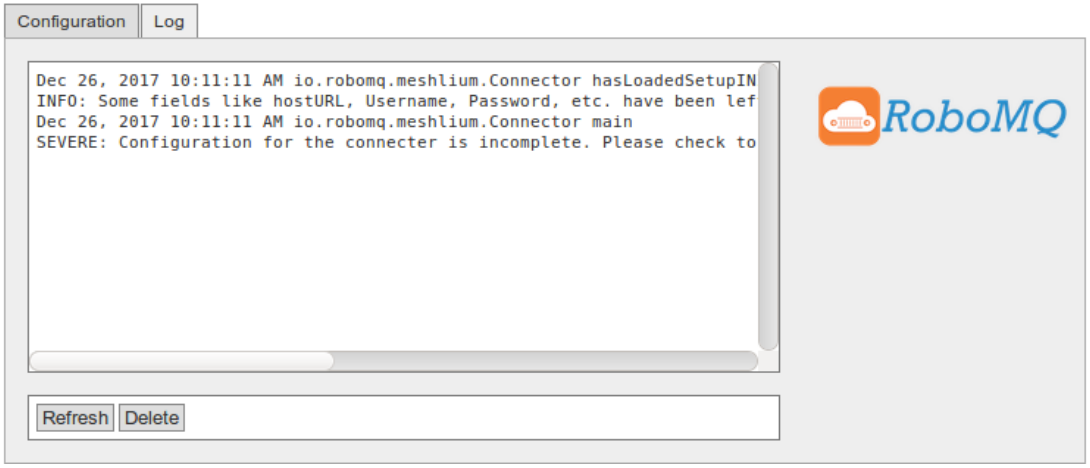


Figure : RoboMQ Connector logging in action

12.3.27. scriptr.io

scriptr.io is the Internet of Things Application Platform and Marketplace of extensible IoT Applications. It combines a visual environment and an extensive set of APIs for the rapid design, integration and development of applications that execute on a scalable, secure and robust runtime. scriptr.io is compatible with all major IoT data platforms to accelerate your IoT Digital Transformation.

Using Meshlium with scriptr.io

To use Meshlium with scriptr.io you need the following: (1) Sign-up to scriptr.io, (2) Create a channel in your scriptr.io account, (3) Create a script in your scriptr.io account to receive the data sent by Meshlium, (4) Create a device on scriptr.io to authenticate the requests sent by Meshlium.

Sign-up to scriptr.io

You can sign-up for an account by selecting one of the available registration methods from:

https://www.scriptr.io/register

After signing-up, or if you already had an account, sign-in to scriptr.io from https://www.scriptr.io/login using your credentials. Once authenticated, you will be directed to scriptr.io's Web IDE.

Create a channel in your scriptr.io account

scriptr.io's channels are used to broadcast messages to your scripts, widgets or to the client applications that are using your services. To create a channel from scriptr.io's Web IDE (https://www.scriptr.io/workspace), click on the drop-down arrow near your username at the top-right corner of the screen, then select "Settings".

In the configuration dialog that is displayed, select the "Channels" tab. Click on "+Add Channel", then enter a name (e.g. "libelium"). Keep the checkboxes unchecked so that only authorized entities can publish or subscribe to this channel. Click on the check sign on the right to validate your changes.

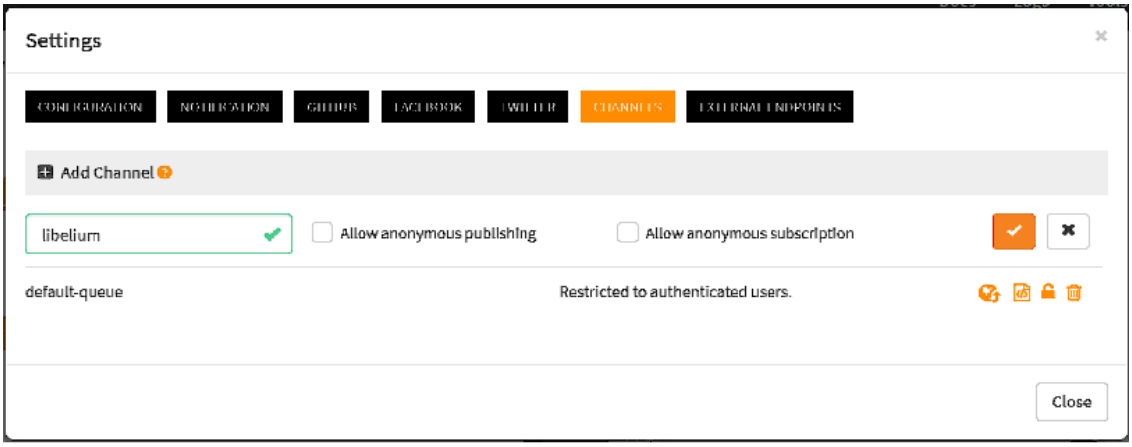


Figure : Configure your channel

Create a script to receive data

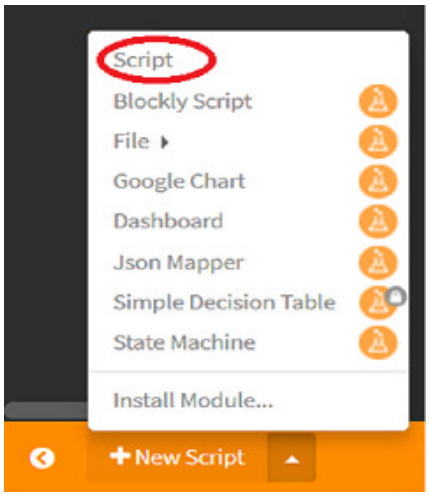


Figure : Create a new script

Scripts are used to implement the logic of your IoT applications, in JavaScript. In the current case, we need a script that receives the data sent by the Meshlium device. To create a new script, click the "+New Script" option on the bottom-left corner of the Web IDE; or select "Script" from the drop-down.

You can configure Meshlium to use HTTP or AMQP to send data to your scriptr.io account. The below code caters to both configurations and parses the received data into a JSON object ("payload" in the below example). Copy & paste the following into the script editing area of the Web IDE:

```
// require the scriptr.io's log module and configure it to the "info" level
var log = require("log");
log.setLevel("info");
try {
    var payload = null;
    // Check if the payload is available in request.parameters
    // (AMQP message) or in request.body (HTTP message)
    if (request.parameters && Object.keys(request.parameters).length > 0) {
        payload = JSON.parse(request.parameters.data)
    }else {
        payload = JSON.parse(request.body);
    }
    log.info("Received the following payload:\n" + JSON.stringify(payload));

    // ADD YOUR CUSTOM LOGIC HERE

}catch(exception){
    log.error("Something went wrong\n" + JSON.stringify(exception));
}
```

Give a name to your script (e.g. “libelium”) and save it.

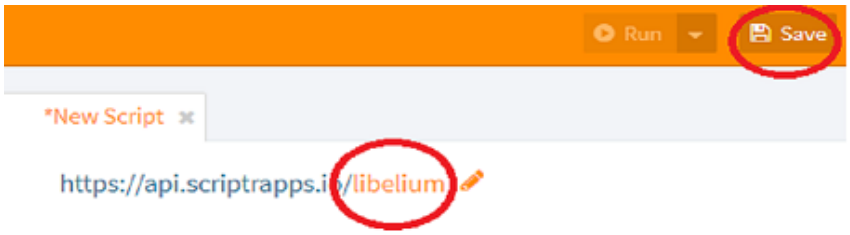


Figure : Name and save your script

Create a device to authenticate the requests

To access your resources on scriptr.io, third party clients - such as your Meshlium gateway - should authenticate. For that, they will use the credentials obtained after having been identified as devices in your scriptr.io account.

To create a new device, click on the drop-down near your username in the top right corner of the Web IDE, then select “Device Directory”. Enter an ID and a password for your device. Confirm the password, then validate your changes by clicking on the check button on the right.

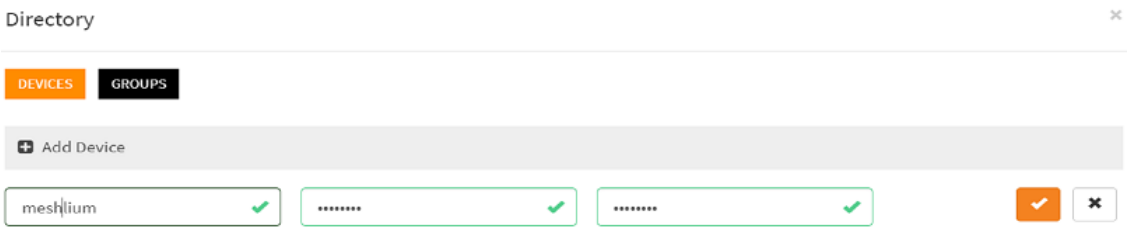


Figure : Create a new device

Your device is added and an authentication token has been generated for it.

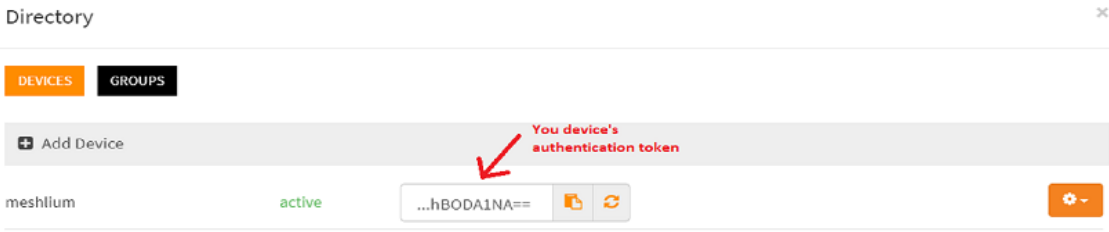


Figure : Every new device gets an authentication token to authenticate against scriptr.io

Configuration

To configure the scriptr.io cloud connector on Meshlium's Manager System, go to:

Cloud Connector → Basic Cloud Partner → scriptr

The configuration form is divided into 3 sections: a common section and 2 sections that are specific to the protocol (HTTP or AMQP) you will decide to use.

***Note**: if you have signed-up for a free scriptr.io account you should only choose HTTP.*

Shared configuration

In the "Common" section, fill the following:

- **Script name**: enter the name of the script you created in the "Create a script to receive data" section (e.g. "libelium").
- **Message template**: configure the message to send to scriptr.io (this depends on the measures sent by your sensors).
- **Log level**: specify a log level.
- **Interval**: specify a time in seconds between 2 synchronizations with your scriptr.io account. This value cannot be less than 30.

Using HTTP to send data

In the "Common" section, fill the following:

- **Server adddress**: type "api.scriptrapps.io".
- **Protocol**: choose "HTTP" (default value).

In the "HTTP" section, fill the following:

- **Auth token**: enter the value of the authentication token that was generated by scriptr.io for your Meshlium device, as described in the "Create a device to authenticate the requests" section.

Using AMQP to send data

***Note**: if you have only signed-up to a free scriptr.io account don't use AMQP.*

In the "Common" section, fill the following:

- **Server address**: type "amqp.scriptr.io".
- **Protocol**: choose "AMQP".

In the "AMQP" section, fill the following:

***Note**: check next paragraph, "Obtaining AMQP configuration from scriptr.io", for the values obtained from scriptr.io.*

- **User**: paste the value of the "Username" field from scriptr.io.
- **Password**: paste the value of the "Password" field from scriptr.io.
- **Virtual host**: paste the value of the "Virtual host" field from scriptr.io.
- **Exchange**: paste the value of the "Exchange" field from scriptr.io.
- **Routing key**: paste the value of the "Routing key" field from scriptr.io.

Obtaining AMQP configuration from scriptr.io

From scripr.io's Web IDE, click on your username on the top-right corner of the screen and select "Queuing". In the resulting panel:

1. Select "AMQP" as protocol.
2. In the "Credentials" section, select the device you have created in the "Create a device to authenticate the requests".
3. In the "Publishing Details" section, select the channel you have created in "Create a channel in your scriptr.io account".

Copy the value of "Virtual host", "Username", "Password", "Exchange name" and "Routing Key to invoke" and paste them in the "AMQP" section of Meshlium's configuration.

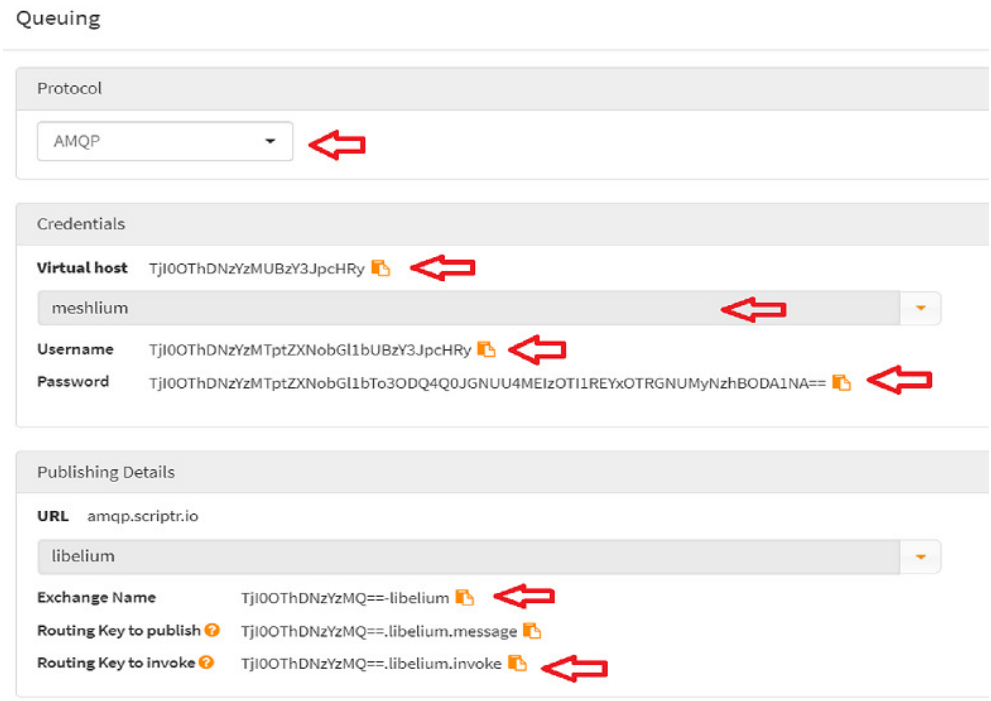


Figure : AMQP configuration from scriptr.io

In Meshlium Manager System, click the "Save" button for storing your configuration.

Controlling the synchronization

Once the cloud connector is configured, the user can launch it using the "Start" button. At specified intervals, the frames received from the sensors and stored in the local database will be sent to your script on scriptr.io. Check the status indicator of the cloud connector to know if it is "Running" (green light) or "Stopped" (red light).

Figure : scriptr.io cloud connector is stopped

You can stop the scriptr.io cloud connector anytime clicking on the "Stop" button.

Figure : scriptr.io cloud connector is running

12.3.28. SensorUp IoT Platform

SensorUp provides an open standard IoT platform that enables information from all different kinds of sensors accessible in a single platform.

Configuration

You can access the SensorUp IoT platform plugin from the Cloud Connector menu, and setup all the information needed to connect Meshlium to the SensorUp IoT platform.

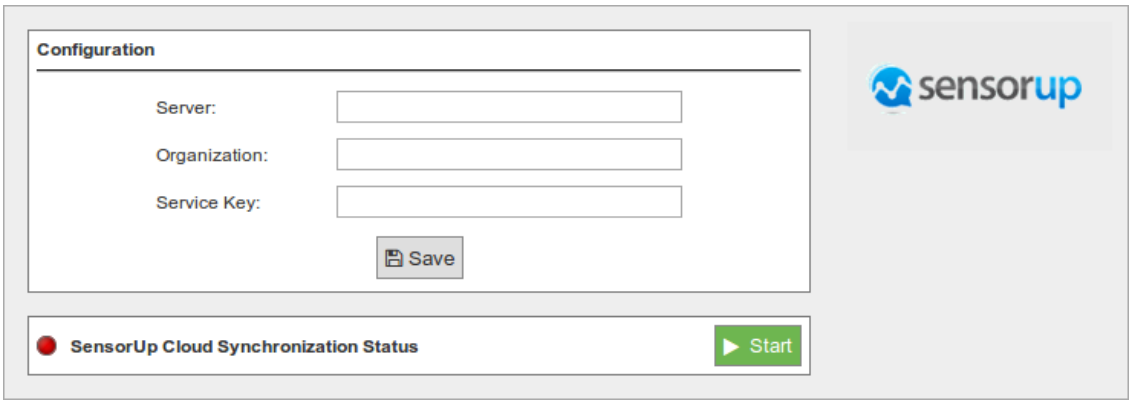


Figure : SensorUp configuring plugin

- **Server**: SensorUp IoT platform server.
- **Organization**: Identifier of your organization.
- **Service Key**: Key used to access SensorUp IoT platform.

All these parameters are provided by SensorUp.

Controlling synchronization

Once you have saved the configuration, you can send your data to the SensorUp IoT platform by pressing the “Start” button. You will notice about it because the screen shows a spinning wheel when the process starts and displays a “running” status.

Figure : SensorUp status “Running”

If you want to stop this process, just press the “Stop” button. You can start/stop this process anytime.

Figure : SensorUp status “Stopped”

12.3.29. Sentilo

Sentilo is an open source sensor and actuator platform designed to fit in the Smart City architecture of any city who looks for openness and easy interoperability. It is built, used, and supported by an active and diverse community of cities and companies that believe that using open standards and free software is the first smart decision a Smart City should take.

Configuration

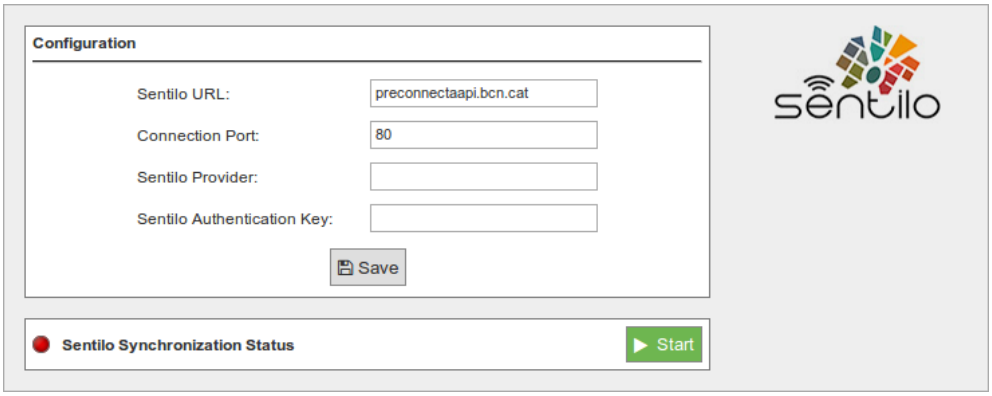


Figure : Configuring Sentilo in Meshlium

Inside the "Sentilo" plugin, you have a form to introduce your credentials to access your Sentilo system. You have to enter here these parameters:

- **Sentilo URL**: Address of the API service of Sentilo. This address should be provided without the "http://".
- **Connection Port**: The port in which the API listens to connections.
- **Sentilo Provider**: The provider is the identity of who is sending data to Sentilo.
- **Sentilo Key**: The security key to send data to Sentilo.

This data will be provided by the administrators of the Sentilo system you are using.

Sensor, Types and Components are not created automatically, you need to manually create them in Sentilo as a previous step.

Controlling synchronization

The synchronization will be done in packs of 100 data at a time, so the system is not overloaded. You can start and stop the synchronization of the data to the Sentilo service. In the interface, you can see an indicator of whether the Sentilo service is running or not. If you click on "Start", the synchronization will begin.

Figure : Sentilo synchronization service is running

You can stop at any moment clicking on "Stop" button.

Figure : Sentilo synchronization service is stopped

12.3.30. Simfony

Simfony's IoT Platform is focused on providing the core set of tools that enables the rapid roll-out of any IoT project or service. Companies can use the service to easily and rapidly design, prototype and deploy IoT projects that match their exact needs and requirements, rather than looking for an off the shelf product that fits best. The platform provides the following services: global mobile data connectivity, SIM management and control, device authentication and authorization, a visual service designer, data storage, reporting and visualization, IoT VPN and an extensive API exposing all of these capabilities for enterprise integration. All these services are available on a "pick and choose" basis enabling maximum flexibility and optimizing costs.

More information can be found at www.simfonymobile.com.

Configuration

The Simfony Cloud Connector is capable of self-configuration using data already provisioned from the Simfony Cloud Platform. This function requires that the user is authenticated and authorized into the Cloud Service with a specific set of credentials provided through the Self-Care portal. These credentials are not stored on the Meshlium or by the Connector and will have to be entered manually each time an Auto-Configuration action is requested through the web GUI. This functionality runs on the user's browser that is connected to the Meshlium device and requires Internet access, i.e. from the browser to the Cloud Platform API.

Once the user is authenticated, a list of all the Cloud provisioned devices is available for selection. If one of the devices is selected in the drop-down list, the web GUI will automatically fill in or overwrite the following parameter: Client ID, Device ID, Device Name, and Device Password.

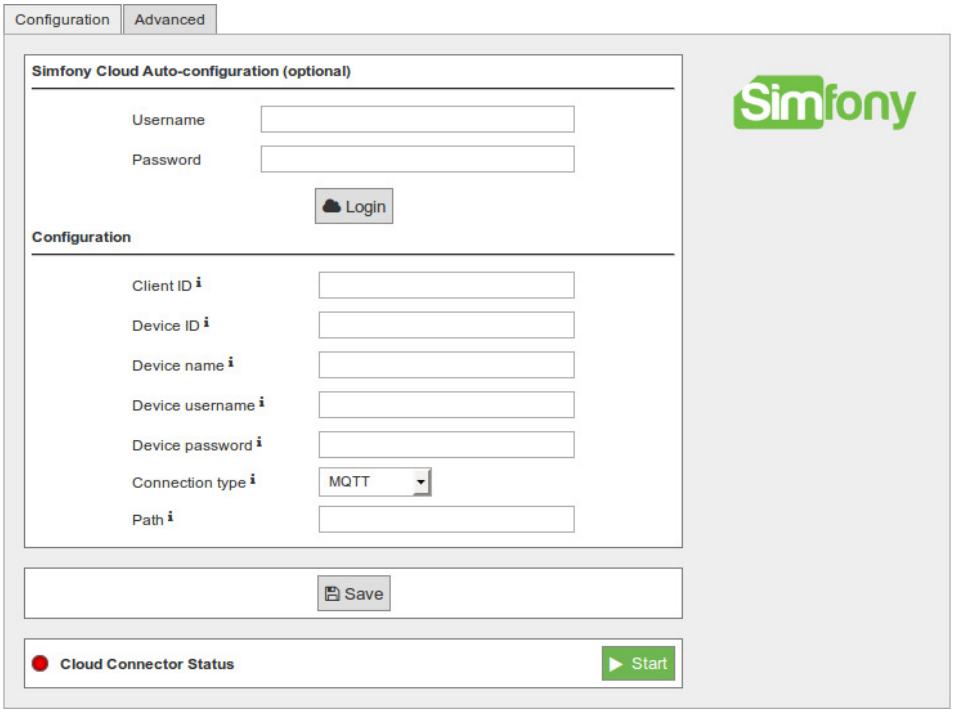


Figure : Simfony cloud connector configuration panel

If the Cloud Connector is already configured with a valid Device ID the Auto-Configuration feature will automatically retrieve the Cloud provisioned data corresponding to that Device ID and fill in the parameters mentioned above. Any previous configuration is overwritten. This functionality can be used to resync the data provisioned in the Cloud with the configuration data of the Connector.

All the data retrieved automatically from the Cloud can also be entered manually.

The user must use the Save button to save any newly configured data or apply any changes to it.

Advanced configuration

The advanced configuration window of the Connector allows the setting of the following parameters:

- **Connection retries**: Controls the number of connection testing retries before suspending operations and going to the sleeping phase (see the Functional description chapter, Test connectivity phase in "Simfony Meshlium Connector- User Guide"). **Default: 3**.
- **MQTT QoS**: Controls the QoS of the MQTT PUBLISH messages. **Default: 1**.
- **Permanent MQTT connection**: In case of MQTT connections, controls if the Connector will close the MQTT connection or not during the sleep phase. **Default: false**.
- **Refresh Interval**: The number of seconds the Connector will suspend its operations (sleep time) before starting a new extract and transmit cycle (see the Functional description chapter in "Simfony Meshlium Connector- User Guide"). **Default: 300**.
- **Maximum transmit interval**: The number of Refresh Intervals after which the Connector will transmit the data independently of the number of new database records found and the "Minimum number of DB records" parameter value. **Default: 5**.
- **Minimum number of DB records**: The minimum number of new database records that will trigger a sending procedure of the Connector. If the found new number of records is lower (strictly) than the value of this parameter, the transmit phase will be suspended until the number or records reaches the threshold or the condition expires (see Maximum transmit interval in "Simfony Meshlium Connector- User Guide"). **Default: 1**.
- **Aggregate sensor data**: Controls the way the Connector aggregates the sensor data found in the DB. If "true", the connector will aggregate sensor data from the same Waspmote frames into a single message. If "false" the Connector will transmit the data individually as extracted from the database. **Default: true**.

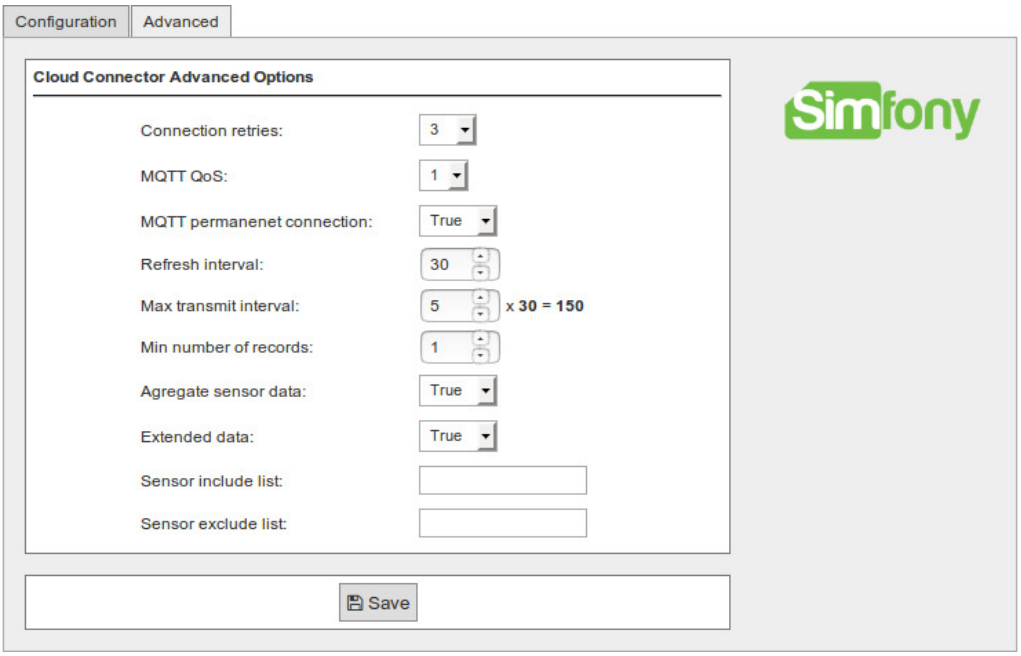


Figure : Simfony cloud connector advanced configuration panel

- **Extended sensor data**: Controls the number of parameters the Connector will transmit to the cloud. If "true", the whole data extracted from the database will be sent. If "false", only a subset of the data stored in the database is sent. **Default: false**.
- **Include sensor list**: The sensor ID list (section "Sensor list") that the Connector will look for when extracting data from the Meshlium database. The "Include" and "Exclude" lists are exclusive with the Exclude list having higher precedence. **Default: empty**.
- **Exclude sensor list**: The sensor ID list (section "Sensor list") that the Connector will exclude when extracting data from the Meshlium database. The "Include" and "Exclude" list are exclusive with the Exclude list having higher precedence. **Default: empty**.

Running the connector

After the entire configuration is complete, the user can start the connector using the “Start” button of the GUI. The Connector will be started and run seamlessly in background.

Figure : Simfony start button

The Status box will show the Connector’s state whenever the page is viewed by the user.

To stop the Connector, the user can press the “Kill” button that will stop the connector from running.

Warning: The “Kill” operation will terminate the Connector process and all it procedures abruptly independently of the stage they are in, i.e. extracting, transmitting, etc.

Figure : Simfony stop/kill button

To stop gracefully the connector the “Stop” button can be used. This will not interrupt any ongoing operations but rather wait for the connector to finish any ongoing activities. The Connector will look for this graceful stop signal each time it is starting or finishing the sleep cycle.

Integration with Simfony’s IoT platform

The Simfony Cloud Connector is intended to work with any type of connectivity provided by the Meshlium device it is deployed on. The Connector has two standard protocols available for communicating with the Simfony Cloud: MQTT and HTTP. Both of them are available in the encrypted version also, i.e. MQTT+SSL and HTTPS. The customer it is able to choose the most appropriate protocol for his application.

Each Cloud Connector/Meshlium device must be individually authenticated and authorized before it can send data to the Simfony Cloud Service. The IoT platform will perform protocol specific authentication and authorization procedures and will allow the connectors to send data only if these are successful. The Connector configuration data must contain these credentials before the Connector can run properly.

Before data can be sent from the Connector to the Cloud Platform, a Cloud IoT Application must be deployed in order to listen for data. Customers can easily create, test and deploy their own applications via the Application Designer GUI. Each application can have a specific entry point for the data coming from the sensor and connectors. This entry point is defined by the used Protocol (MQTT or HTTP) and a custom target (MQTT-topic; HTTP-path). This entry point must be also configured in the Connector via the “Connection Type” and “Connection Path” parameters. Once the application is deployed, the connectors can start sending data into it and the custom business logic will be triggered.

Find out more about running the connector in the “Simfony Meshlium Connector- User Guide” at http://www.simfonymobile.com.

12.3.31. SmartCityPlatform

The SmartCityPlatform connects all the core elements of a successfully managed city. It allows the SmartMayor to manage urban development, socio-economic development and technological development of the city, measuring and reporting progress in real time.

By using this cloud connector, you connect to the Sense module, which is a part of SmartCityPlatform. It allows an overview of the city's pulse by gathering, measuring & monitoring happenings in the city.

www.smartiscity.eu

Configuration

To use the SmartCityPlatform cloud connector, you need to register your Meshlium in the authentication server. The server provides you with credentials, which you will enter in the form below. When you finish configuring, click the "Save" button to save the configuration.

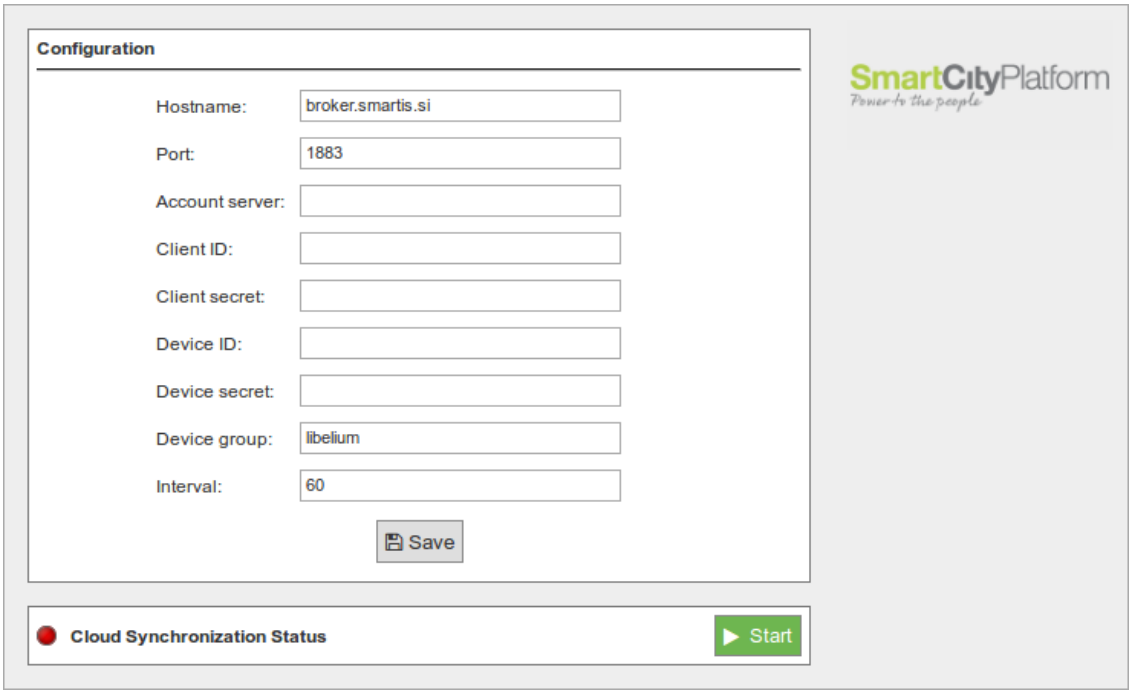


Figure : SmartCityPlatform cloud connector configuration panel

- **Hostname**: the IP or hostname of the sensor broker.
- **Port**: the port where the sensor broker is listening for connections.
- **Account server**: the IP or hostname of the account server.
- **Client ID**: client identification provided by the authentication server.
- **Client secret**: client secret provided by the authentication server.
- **Device ID**: device identification provided by the authentication server.
- **Device secret**: device secret provided by the authentication server.
- **Device group**: the name you set for your device group.
- **Interval**: time duration in seconds between synchronizing data batches.

Controlling synchronization

With the configuration saved, you can start using the cloud connector. To start the synchronization, press the green “Start” button on the right.

Figure : SmartCityPlatform start button

You get a “loading” status inside the synchronization control section, indicating the synchronization is starting.

Figure : SmartCityPlatform loading button

When the cloud connector starts, a green dot on the left appears, indicating the synchronization is running. To stop the synchronization, simply click on the red “Stop” button on the right.

Figure : SmartCityPlatform stop button

12.3.32. SmartPlants

Smartplants Cloud integration enables secure communications between the devices connected to the Meshlium device and the cloud.

Configuration

You will receive the configuration information that is required to connect your Meshlium to the SmartPlants system via an e-mail from Smartplants.

Controlling synchronization

Once you configured the server/broker, the user can launch the Meshlium SmartPlants script (clicking on the "Start" button). The program will search for the received frames on the local database, and will send them to the Smartplants Cloud platform via an MQTT protocol. The status indicator displays the current state, saying "Running" or "Stopped".

Figure : Smartplants sender is running

You can stop the Smartplants program anytime clicking on the "Stop" button.

Figure : Smartplants sender is stopped

12.3.33. Sofia2

Sofia2 is a middleware developed by Indra that allows the interoperability of multiple systems and devices, offering a semantic platform to make real world information available to smart applications (Internet of Things).

It is multi-language and multi-protocol, enabling the interconnection of heterogeneous devices. It provides publishing and subscription mechanisms, facilitating the orchestration of sensors and actuators in order to monitor and act on the environment.

Configuration

The plugin to connect Meshlium to Sofia2 platform is in the Manager System menu:

Cloud Connector → Premium Cloud Partner → Sofia2

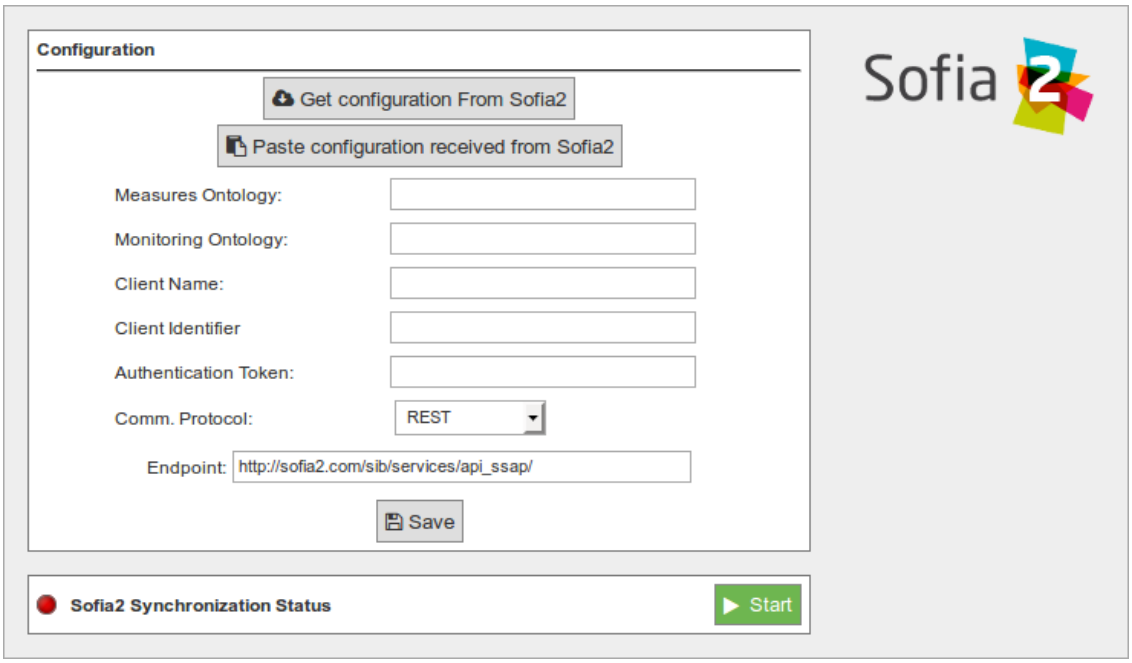


Figure : Sofia2 configuration

Registering the Meshlium device in Sofia2 is a previous step required to connect the Meshlium with Sofia2. At the end of the registration process you will obtain the configuration parameters needed to set up the Meshlium plugin properly.

This configuration includes the following information:

- **Measures Ontology**: Collection (table or storage) where the measures from Waspmote sensors sent by Meshlium (the gateway) will be stored into the platform.
- **Monitoring Ontology**: Collection (table or storage) where monitoring values (internal temperature, battery level...) of the Waspmotes connected to Meshlium, will be stored into the platform.
- **Client Name**: Name of the Meshlium unit to be identified by Sofia2 platform, checking if it has permission to write on the ontologies.
- **Client Identifier**: Identifier of the Meshlium unit to differentiate between several Meshliums using the same Client Name.
- **Authentication Token**: Token to authenticate the Meshlium device during the establishment of a session with Sofia2 platform.

Register Meshlium in Sofia2

To register Meshlium in Sofia2, click on the link Get Configuration From Sofia2.

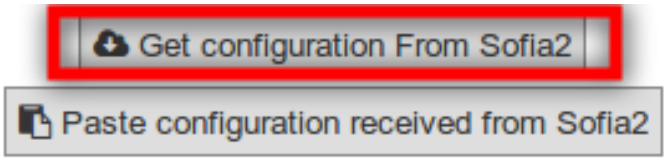


Figure : Sofia2 configuration link

You will be redirected to the following page:

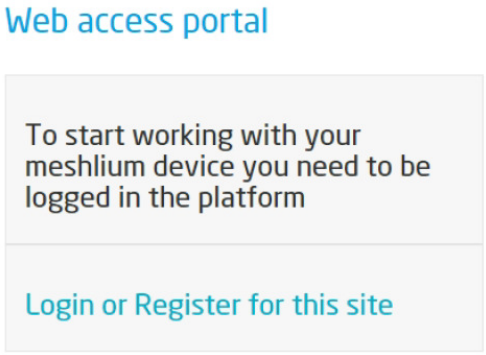


Figure : Sofia2 configuration portal

Where:

- Login using your Sofia2 account.
- Create a new account if you do not have a valid login user.

Logging in Sofia2 platform will redirect to the following page that suggests a name for your collections of measures and monitoring data, and for your gateway identifier (KP in Sofia2 terms):

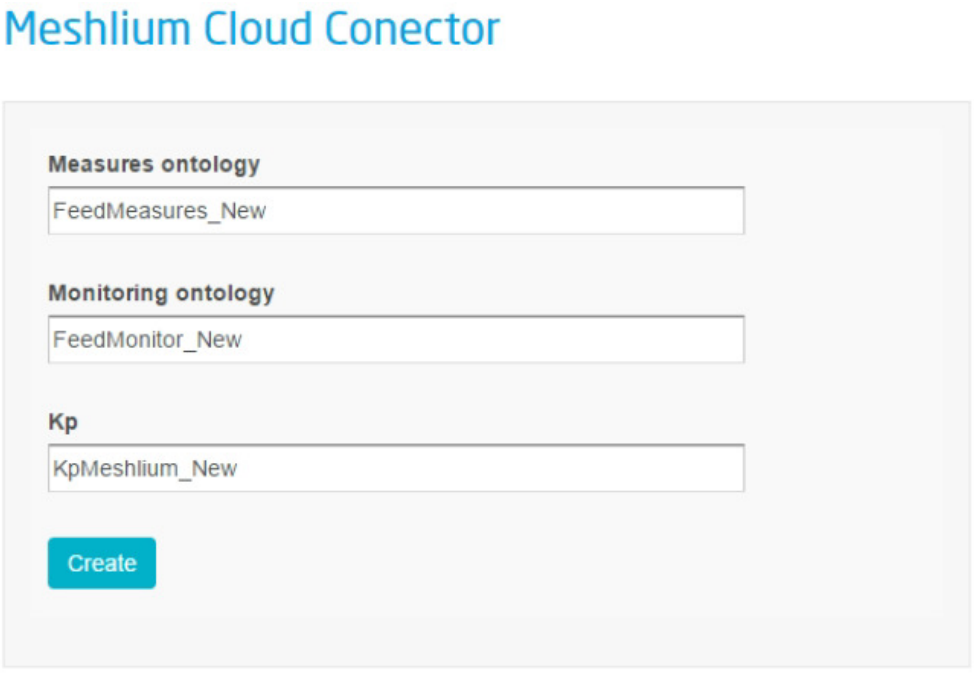


Figure : Sofia2 configuration information

Finally, after creating the configuration, you will obtain a JSON file containing all configuration values ready to be pasted on the Manager System, in order to setup the Sofia2 Cloud Connector:

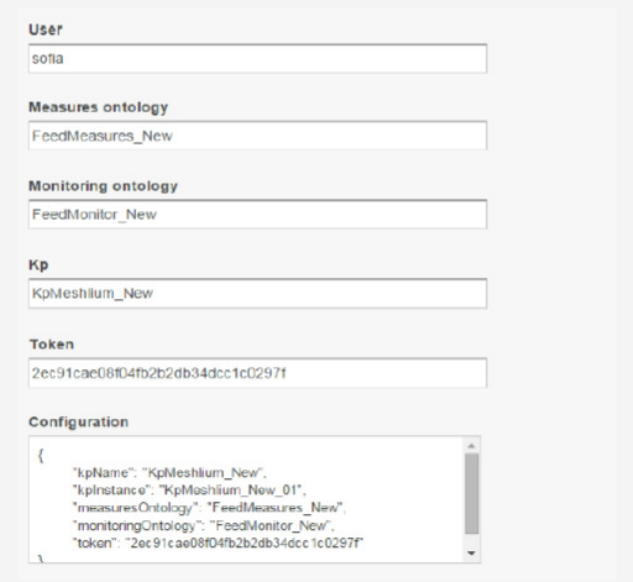


Figure : Sofia2 JSON configuration

Configure the Cloud connector

The configuration from Sofia2 can be setup in Meshlium just by clicking Paste configuration received from Sofia2 and pasting the JSON generated in the previous step.

Figure : Sofia2 pasting JSON configuration

Configuring the plugin this way, the fields for Measures Ontology, Monitoring Ontology, Client Name, Client Identifier and Authentication Token will be completed.

Alternatively, these fields can be filled in manually, with the information received from the configuration page of Sofia2 showed in the first step.

Select communication protocol

REST

REST is a stateless communication protocol over HTTP. Using this protocol, the Cloud Connector is a client of the Sofia2 platform, that periodically opens a connection with the platform to send an HTTP POST operation containing the sensor measurements.

The parameter of the REST protocol are:

Endpoint: URL of the REST Gateway of Sofia2 platform. It is the REST server that will receive requests from clients.

Figure : Sofia2 REST protocol

MQTT

MQTT is a stateful communication protocol over TCP. Using this protocol, the Cloud Connector is a client of the Sofia2 platform, that initially opens a connection with the platform, maintains it alive during that time, and periodically sends an MQTT packet containing the sensor measurements. In case of disconnection, the connector periodically tries to reconnect.

The parameters of the MQTT protocol are:

- **Server**: IP or machine name of the MQTT gateway in the Sofia2 server.
- **Port**: Port of the MQTT gateway in the Sofia2 server.
- **KeepAlive**: Interval in seconds that the connector will use to check the status of the connection.
- **Connection Timeout**: Timeout to establish a connection.
- **Response Timeout**: Timeout to wait response from the Sofia2 server.
- **Auth user**: Optional. MQTT protocol authentication user.
- **Auth password**: Optional. MQTT protocol authentication password.

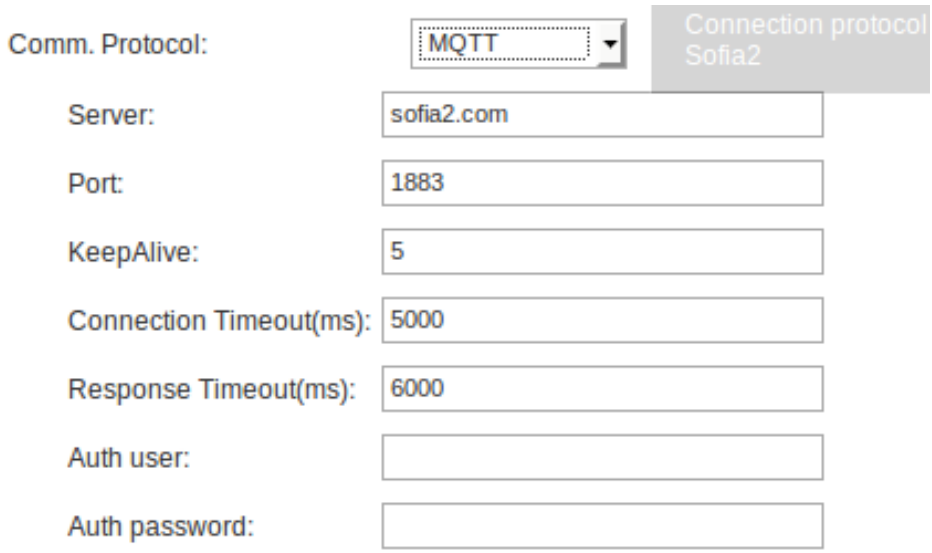


Figure : Sofia2 MQTT protocol

Websocket

It is a stateful communication protocol over HTTP. Using this protocol, the Cloud Connector is a client of Sofia2 platform, that initially open a connection with the platform, maintains it alive during a defined time, and periodically sends a HTTP packet containing the sensor measurements. In case of disconnection, the connector periodically tries to reconnect.
The parameters of the Websocket protocol are:

- **Endpoint**: URL of the Websocket gateway of Sofia2 platform. It is the server that will receive requests from clients.
- **Timeout**: Timeout for any operation with the server.

Figure : Sofia2 Websocket protocol

Save the configuration and start the connector

Once all configuration and connection parameters are setup, they can be stored and the connector can be started to send information to Sofia2.

To save the configuration, click on the "**Save**" button:

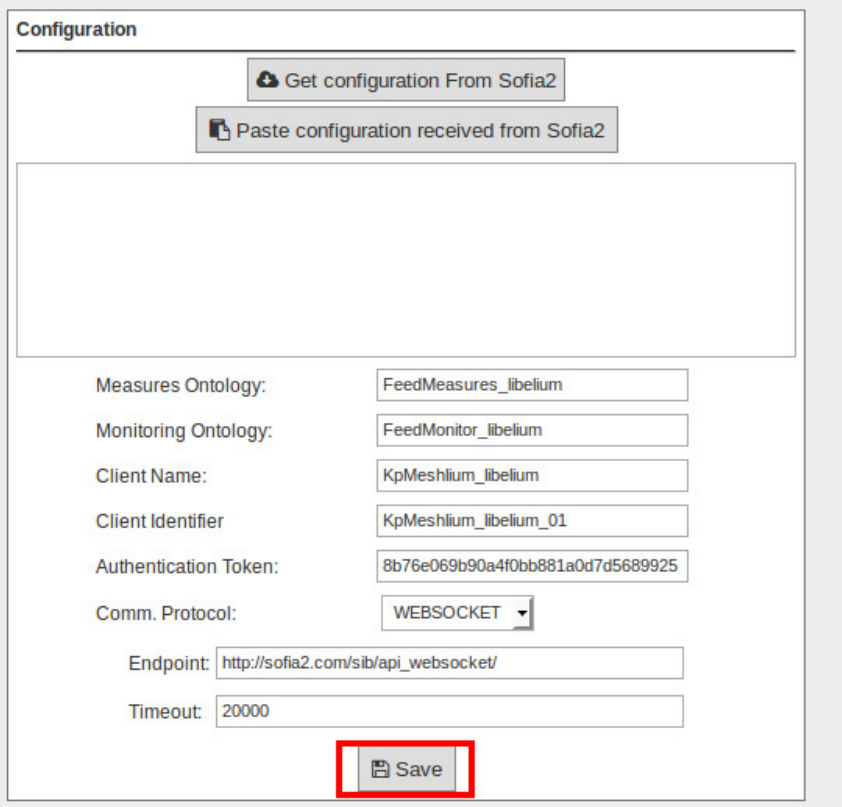


Figure : Sofia2 save configuration button

After saving the configuration, the Cloud Connector can be started by clicking on the "Start" button:

Figure : Sofia2 synchronization service running

You can stop the process at any moment by clicking on the "Stop" button.

Figure : Sofia2 synchronization service stopped

12.3.34. Sparkcompass

This guide will take you through connection to the Sparkcompass platform. Prior to use the Sparkcompass platform with Meshlium, set you up your own sub-domain (e.g. yoursubdomain.pacificfjord.com or yoursubdomain.sparkcompass.com). When you log into your platform instance, select the app you want to receive data from your Meshlium.

How to set up Sparkcompass to receive data from the Meshlium hub

Select the MQQT tab in the Sparkcompass App where you want to receive the data coming from your Meshlium. Your Access UID and Access Key will be generated automatically. Create an MQTT device using the button shown and give it a unique Device ID (up to 23 characters).

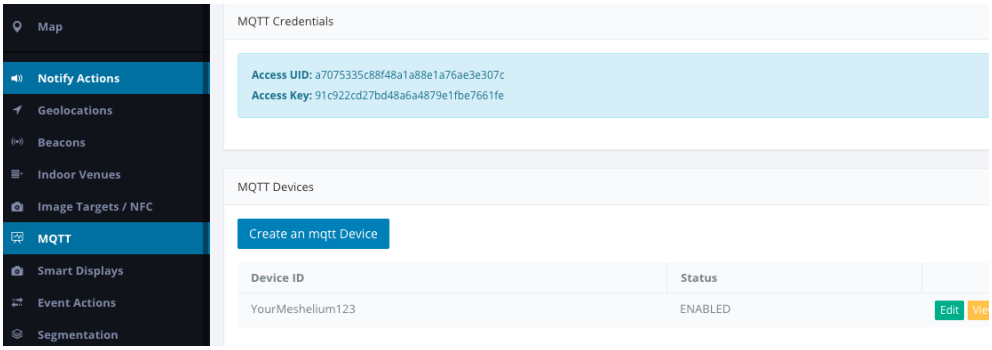


Figure : Retrieve your access credentials from the Sparkcompass Platform

Configuration

You will use the values obtained from the Sparkcompass platform to configure your Meshlium.

In the Configuration panel, the user can set:

- **Host**: yoursubdomain.sparkcompass.com (you will get this from Sparkcompass).
- **Port**: 1883 (unless otherwise directed).
- **Access UID**: as above (e.g. a7075335c88f48a1a88e1a76ae3e307c).
- **Access Key**: as above (e.g. 91c922cd27bd48a6a4879e1fbe7661fe).
- **Device ID**: as created above (e.g. YourMeshlium123).

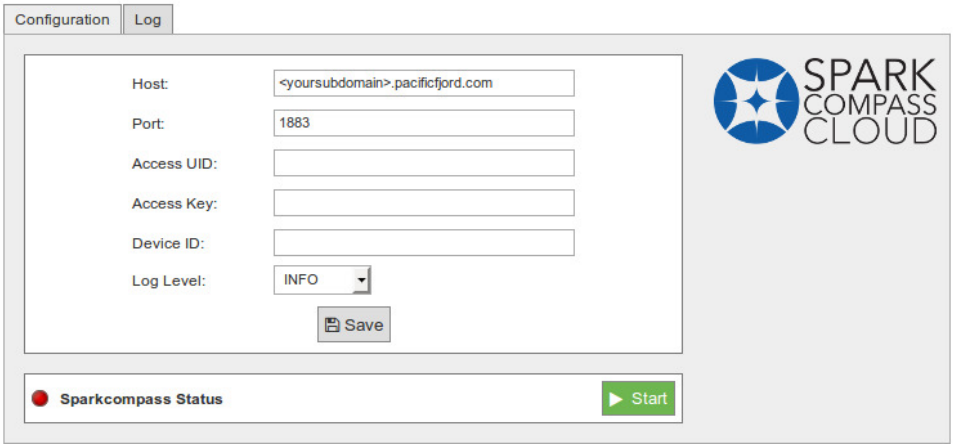


Figure : Sparkcompass plugin configuration panel

Click the “Save” button for storing the configuration fields.

Controlling synchronization

Once configured the connector, the user can launch the Sparkcompass plugin by pressing the “Start” button. The program will search for the received frames on the local database, and will send them to the Sparkcompass. The status indicator displays the current state, saying “Running” or “Stopped”.

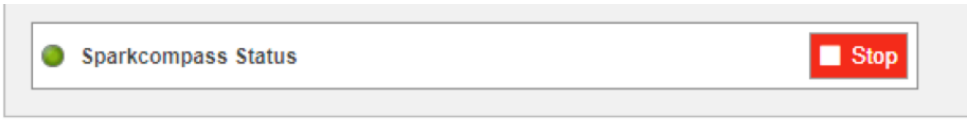


Figure : Sparkcompass sender is running

You can stop the Sparkcompass plugin anytime by clicking on the “Stop” button.

Figure : Sparkcompass sender is stopped

12.3.35. Sparkster

The Sparkster cloud platform simplifies setup, just login and set rules for what data should be sent to the cloud.

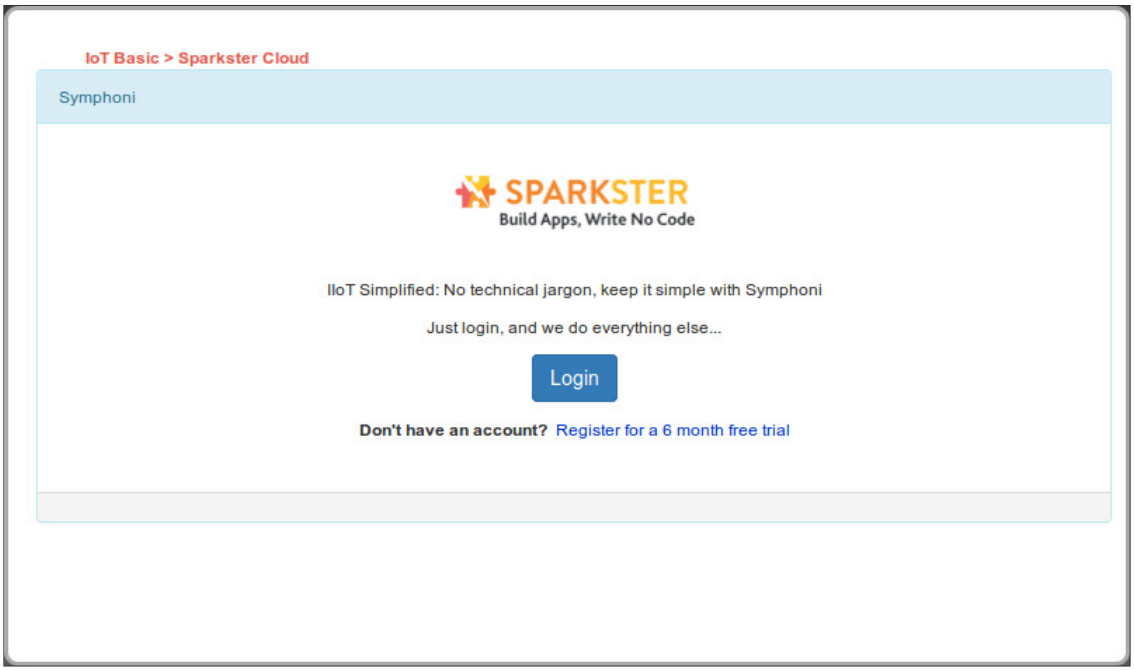


Figure : Sparkster login panel

Register Meshlium

Login to the Sparkster and name your Meshlium, then click the “Register” button.

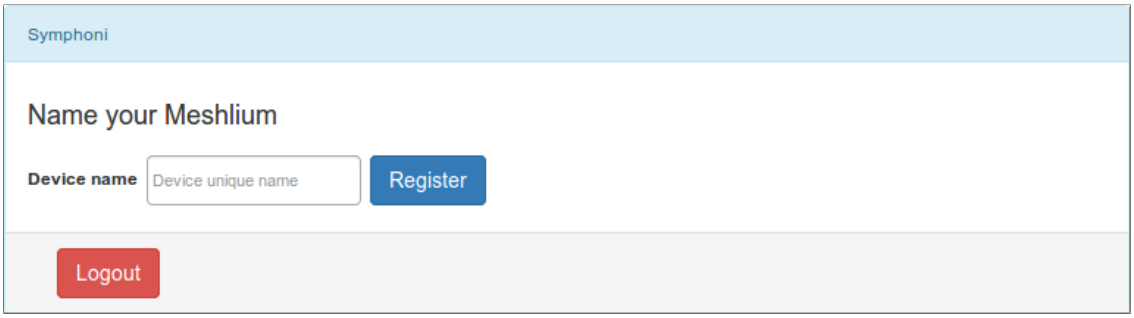


Figure : Meshlium name

Name your sensors

Name your sensors and specify the location of sensors, then click on the “Update” button to save.

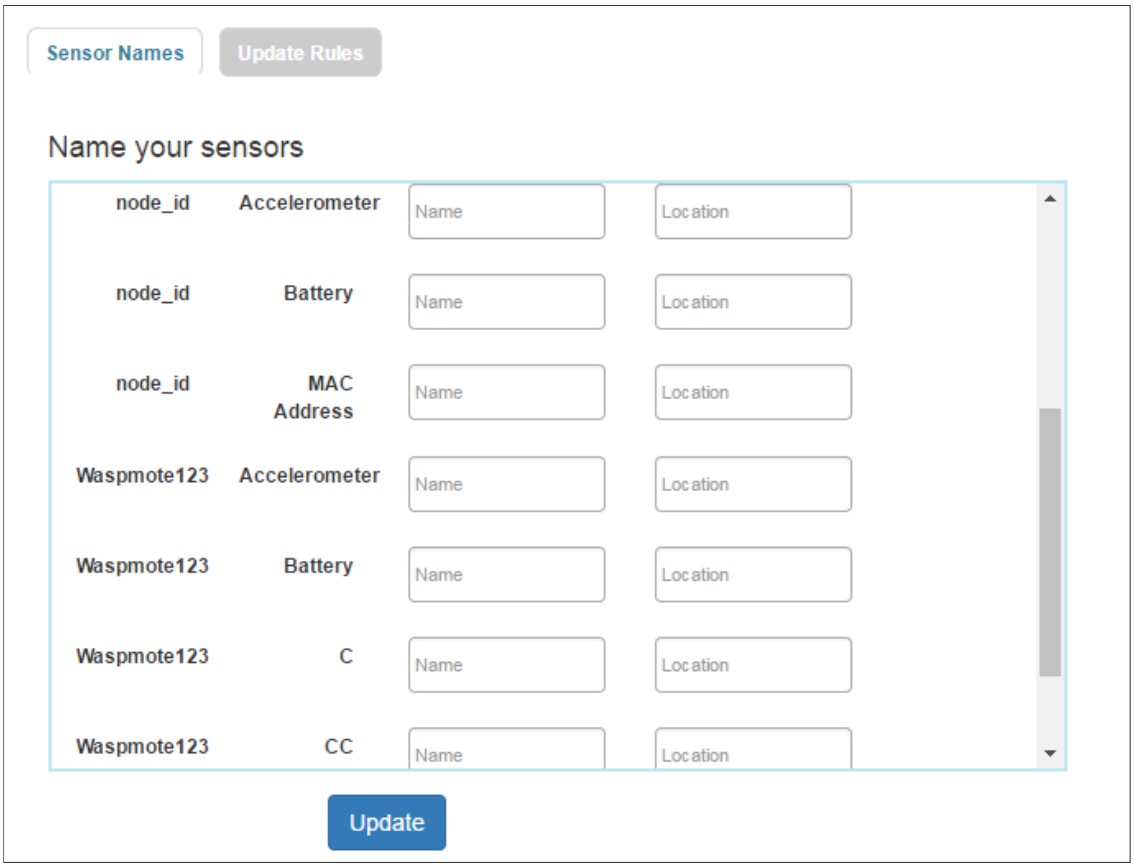


Figure : Sensor names

Create/update rules

Determine how frequently you would like your data to be updated to the cloud, click the “Update” button to save. The lower the frequency, the lower the bandwidth consumed.

Create the rules of what data needs to be sent from the device to the cloud and click on the “Update” button to create/update rules. Examples of these rules include the maximum temperature detected within the update period or notifying the cloud if the battery level is less than 50%. In addition to the value, you may also send a message to the cloud.

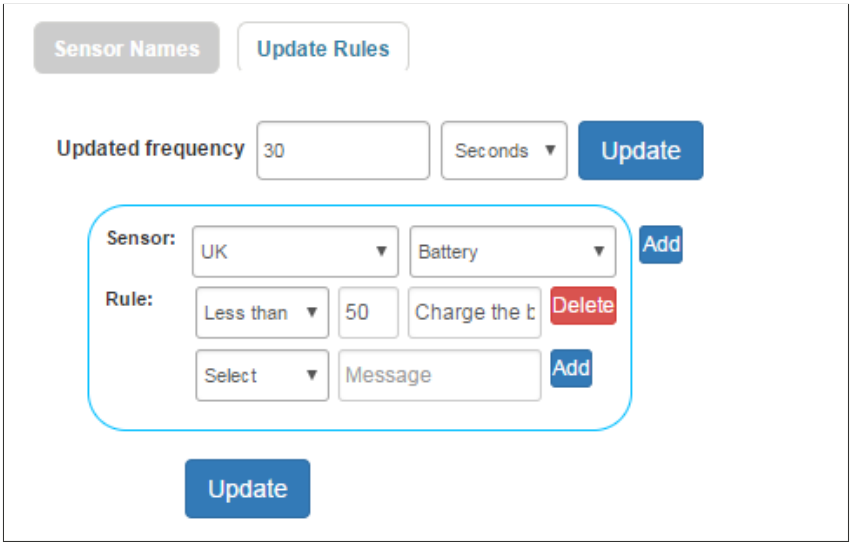


Figure : Sparkster rules

Controlling synchronization

Once you have created all your rules, click the “Start” button to begin the transmission to the cloud. Click the “Stop” button to stop the data transmission.

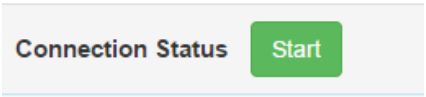


Figure : Connection status

Controlling the application

Click on the “Stop App” button to stop the Sparkster application on your Meshlium.

***Note**: Clicking on the “Stop App” button will log you out and stop any data transmission to the cloud. This will also terminate the Sparkster Application.*

Click on “Logout” button to logout from the application

***Note**: Clicking on the “Logout” button will log you out and stop data transmission to the cloud.*

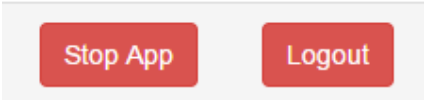


Figure : Stop or log out

12.3.36. TechEdge SAP HANA

SAP HANA Cloud Platform is a platform-as-a-service open (PaaS) that provides unique services for databases and applications in memory. It is the cloud platform that allows you to quickly develop new applications or extend existing ones. Allowing anyone to extend SAP applications within minutes, all in the cloud.

With SAP HANA Cloud Platform you can:

- Deploy in the cloud and their existing on-premise applications. You can quickly add a new functionality to their existing applications in the cloud and on-premise.
- Connect your cloud and on-premise applications to eliminate data silos and make a simple, secure and scalable digital access.
- Create and run new applications in the cloud to solve new problems, make new customers and new income.
- It allows to connect the business processes with field devices through Internet of things (IoT) services.

This platform allows:

- Enable remote services management of devices.
- Communicate through secure protocols with field devices.
- Manage devices and their messages remotely through programming interfaces (API).

Configuring SAP HANA

To make the connection between the platform and the SAP HCP Gateway Meshlium by Libelium, so we can receive the data sent from the gateway, a pre-configuration of Things Internet service is required.

For more information about how to configure the Things Internet service, please contact your TechEdge contact.

Configuration

Press on SAP HCP to access the service configuration screen of TechEdge cloud connector for SAP HANA Cloud Platform.

By accessing the SAP HCP connector configuration screen, a form with the necessary fields to configure is displayed.

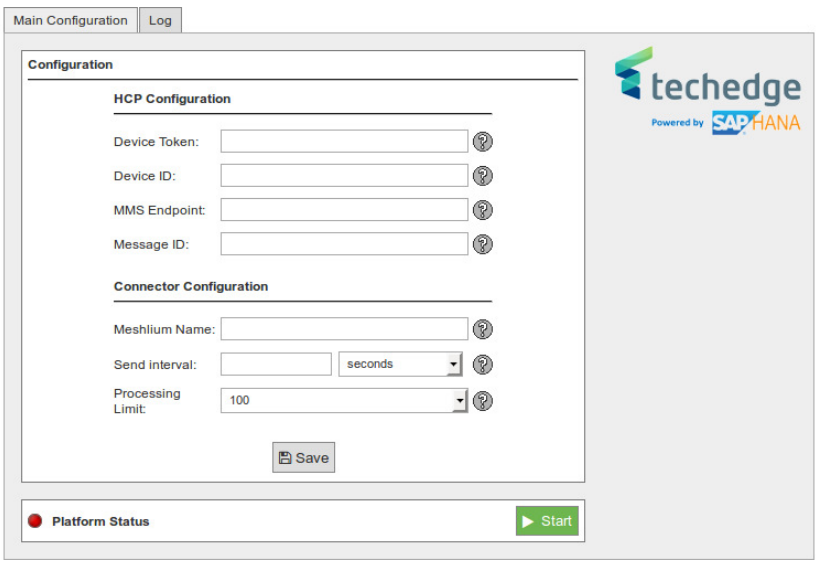


Figure : TechEdge configuration plugin

- **Device Token**: The Token OAuth2.0 provided by HCP to the device create (you must configure SAP HCP IoT before, as described in the previous sections).
- **Device ID**: Identifier of the device created in HCP.
- **MMS Endpoint**: Data Endpoint of HCP MMS services.
- **Message ID**: ID of message type created in HCP.
- **Meshlium Name**: Meshlium unit identifying name (free field).
- **Send Interval**: Defines the connector's space of time to wait between each HCP cloud deliveries. Each delivery contains between 100 and 200 sensors traces, and in order to not saturating the Gateway memory, the minimum accepted is 5 seconds.
- **Processing Limit**: Limit of simultaneous messages processing in each of HCP cloud deliveries, the figures are considered between 100 and 200, these are the figures recommended by Libelium to ensure a high performance in the Gateway.

After setting all fields described above, it is necessary to save the changes by clicking the "Save" button at the bottom of the configuration form.

Controlling synchronization

After saving the configuration, you can now start the service. To do this click on the "Start" button.

Figure : TechEdge status "Running"

When the service has started, "Platform Status" is displayed in green, and the "Start" button changes to "Stop".

***Note**: The Data synchronization with Meshlium will be held as maximum packet size defined in "Processing Limit" field. All those new data received since the last delivery will be synchronized, if the data exceeds the maximum size set to "Processing Limit" several deliveries will be made to complete the synchronization.*

With the service has started, to stop the service you must click on the "Stop" button on the HCP configuration screen in Meshlium.

Figure : TechEdge status "Stopped"

Pressing the "Stop" button, a stop will start in a controlled manner, allowing you to stop the service without incident (close files, ends active processes of polling, etc.), ensuring a proper functioning of the Meshlium gateway.

When the service has stopped, "Platform Status" is displayed in red, and the "Stop" button will change by "Start".

12.3.37. Telefonica IoT Platform

Telefonica provides an M2M cloud to collect and analyze data. This platform is based on assets and models and you can optimize your business processes implementing rules and notifications, and subscribing to data from different hosts.

Configuration

A new option is shown in M2M Platform menu, in the Cloud Connector main option. If you expand it, you can see this form with 3 fields in it:

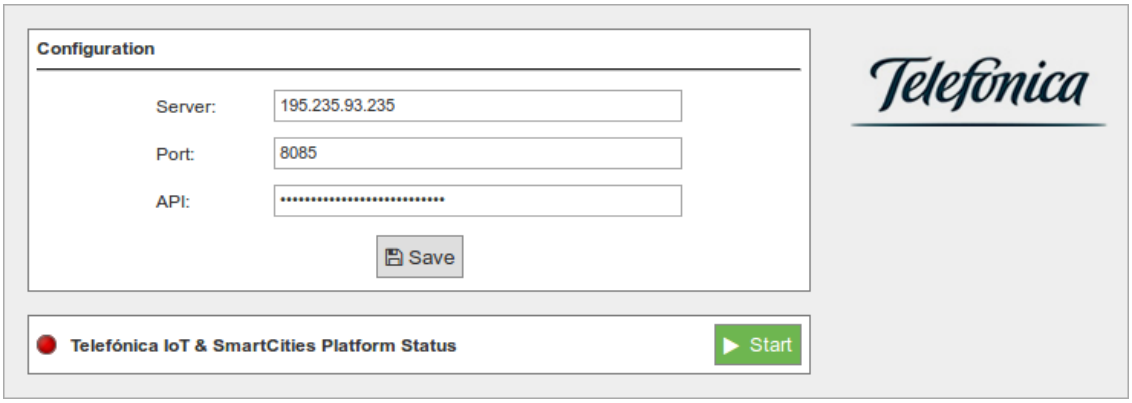


Figure : Telefonica IoT setup example on Manager System

- **URL**: Address of the API service of Telefonica IoT. This address should be provided without the "http://", usually int.dca.tid.es.
- **Port**: The port in which the API listens to connections.
- **API**: The security key to send data to Telefonica IoT.

All this data are provided by Telefonica service administrators.

Controlling synchronization

The synchronization will be done in packs of 100 data at a time, so the system is not overloaded. You can start and stop the data synchronization to the Telefonica service. In the interface, you can see an indicator of whether the Telefonica service is running or not. If you click on "Start", the synchronization will begin:

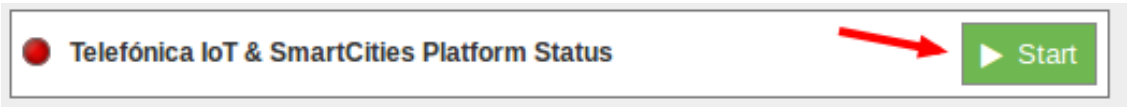


Figure : Telefonica IoT Start button

You can stop at any moment clicking on "Stop" button.

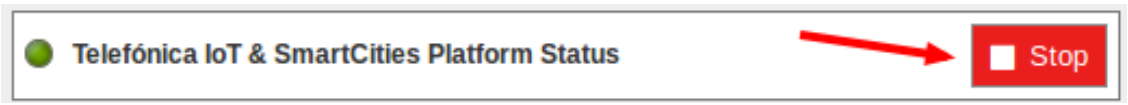


Figure : Telefonica IoT Stop button

12.3.38. ThingPlus

Thing+ allows customers to build their own IoT services with high speed, reliability, scalability, and cost competitiveness, connected by a SaaS or PaaS IoT platform. When Thing+ Embedded devices connect to the Thing+ Cloud (public or private), customers can visualize various data graphs and charts from sensors directly on the Thing+ Portal. Device registration is easy as the Thing+ Portal provides dashboard widgets, a trigger-condition-action-based rule engine for alert notifications or to control actuators, and results in the form of an event timeline.

Get API Key

- Get the “Gateway ID” to register:
  - Open the Meshlium Manager System.
  - Click Cloud Connector.
  - Open the ThingPlus plugin.
  - You can see ThingPlus configuration and “Gateway ID”.
  - Copy the “Gateway ID”.
- Get the API Key:
  - Go to your ThingPlus service (if you have no registered service, register your service) https://yourservice.thingplus.net.
  - Go to Gateway Management page (via upper right menu).
  - Click the + button in the upper right corner.
  - Click “Request for Gateway certificates or API Key” button.

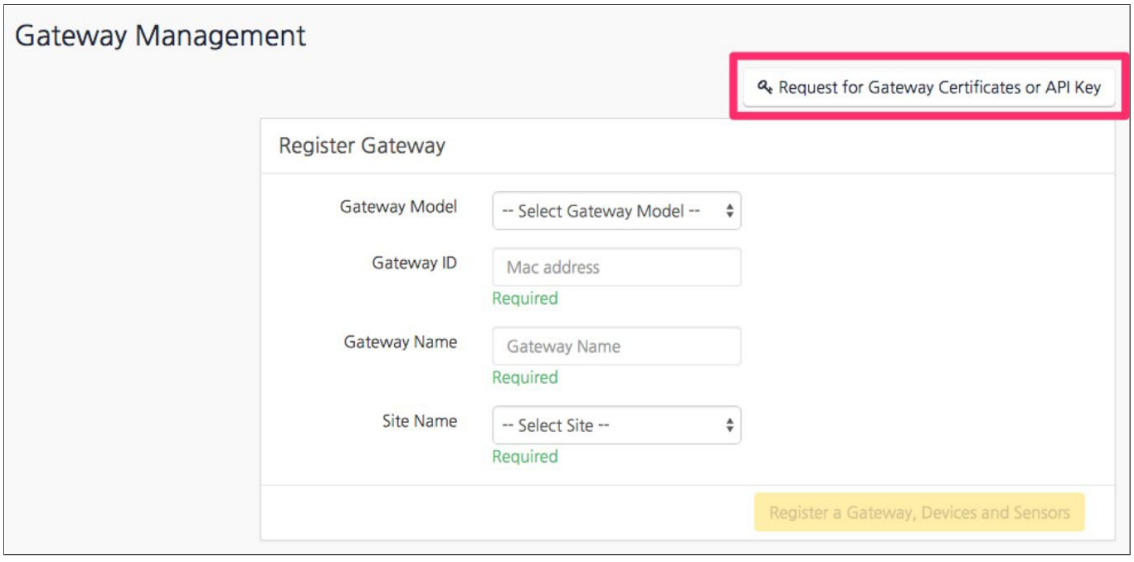


Figure : Request API Key in the ThingPlus panel

- Fill form.
  - Gateway ID.
  - Select “API Key” (Authentication Type).
- Click “Get API Key”.
- Copy the “API Key”.

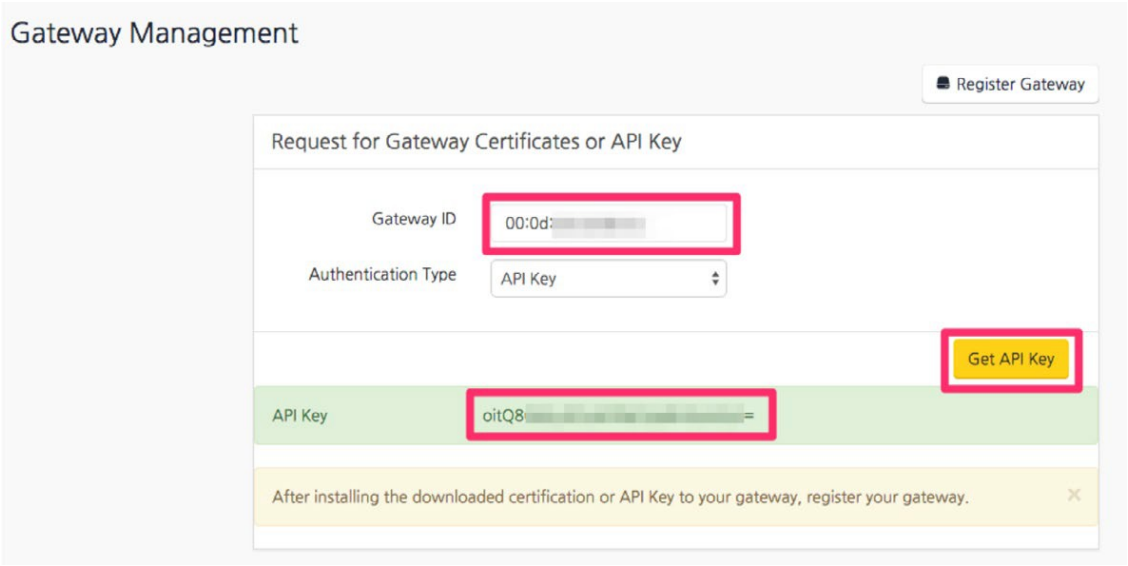


Figure : Get the API Key in the ThingPlus panel

- Set “API Key” in the Meshlium Manager System.
  - Go to ThingPlus configuration again.
  - Paste the “API Key”.

Figure : Enter API Key in the ThingPlus plugin

- Click the “Save” button (ThingPlus gateway app will restart).
- If the status is “STOPPED”, then click on the “Start” button after saving API Key.

Register Gateway and Sensors

- Go to your ThingPlus service.
- Go to Gateway Management page (via upper right menu).
- Click the + button in the upper right corner.
- Fill the form:
  - Select Gateway Model as “Libelium Meshlium”.
  - Input Gateway ID (the same MAC address when registering gateway).
  - Select Device Model as “Waspmote Basic”.
  - Input Device Address (Waspmote address is the id_wasp field).
  - Input Device Name.
  - Select Sensors to register (all sensors are selected as default).
  - Select Site Name (default).
- Click “Register a Gateway, Devices and Sensors” button:
  - ThingPlus gateway app will restart and send the sensor data in a few minutes.
  - You can see the sensor data at Dashboard or Sensor page.

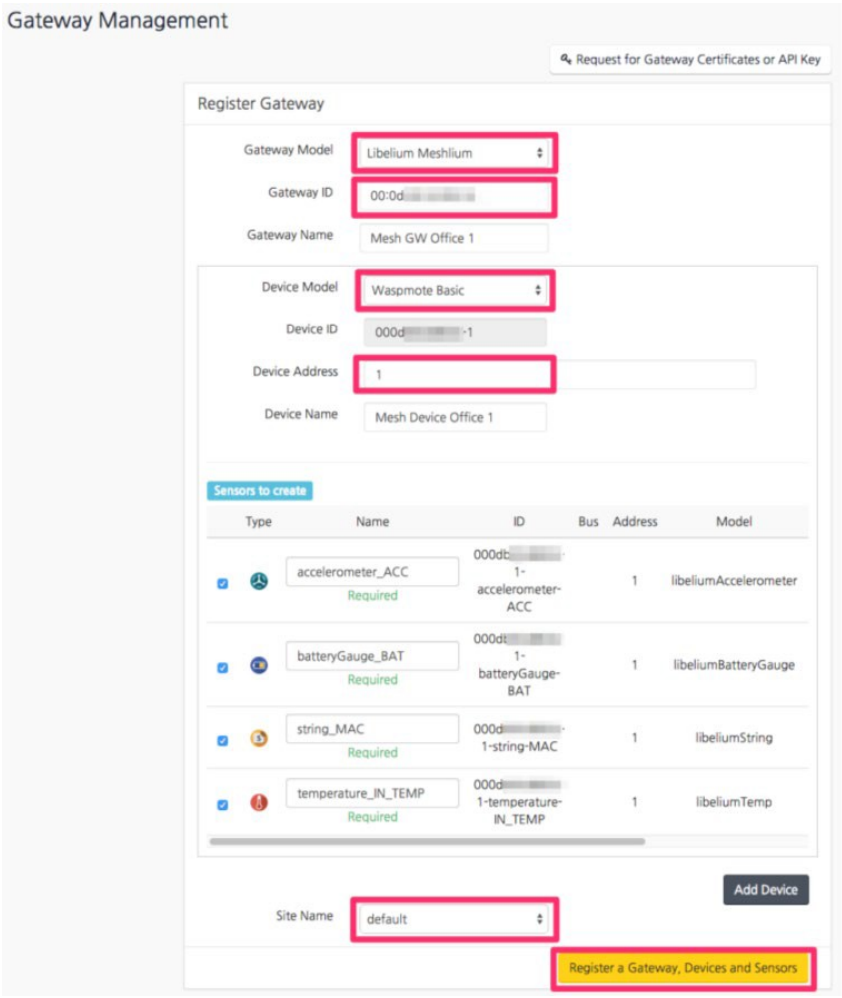


Figure : Registering a Gateway in ThingPlus service.

12.3.39. ThingSpeak

ThingSpeak™ provides instant visualizations of data posted by your sensors to the ThingSpeak cloud. With the ability to execute MATLAB® code in ThingSpeak, you can perform online analysis and processing of the data as it comes in. Use the ThingSpeak Cloud Connector to view your data configuration and send your Waspmote data to ThingSpeak for analysis and display.

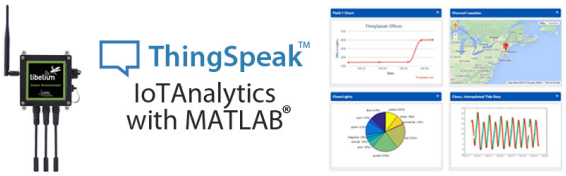


Figure : ThingSpeak cloud

Prerequisites

To use ThingSpeak, you must have a MathWorks account. If you do not already have one, create a MathWorks account (https://thingspeak.com/users/sign_up).

You need a license for commercial use (https://thingspeak.com/prices/thingspeak_standard) or you can sign up for a free evaluation. To use the ThingSpeak Cloud Connector, you must have a programmed Waspmote that is sending data to your Meshlium.

Configuration

In the Meshlium Manager System, select the Configuration tab. The Waspmotes units and sensors connected to your Meshlium device are listed on the configuration tab.

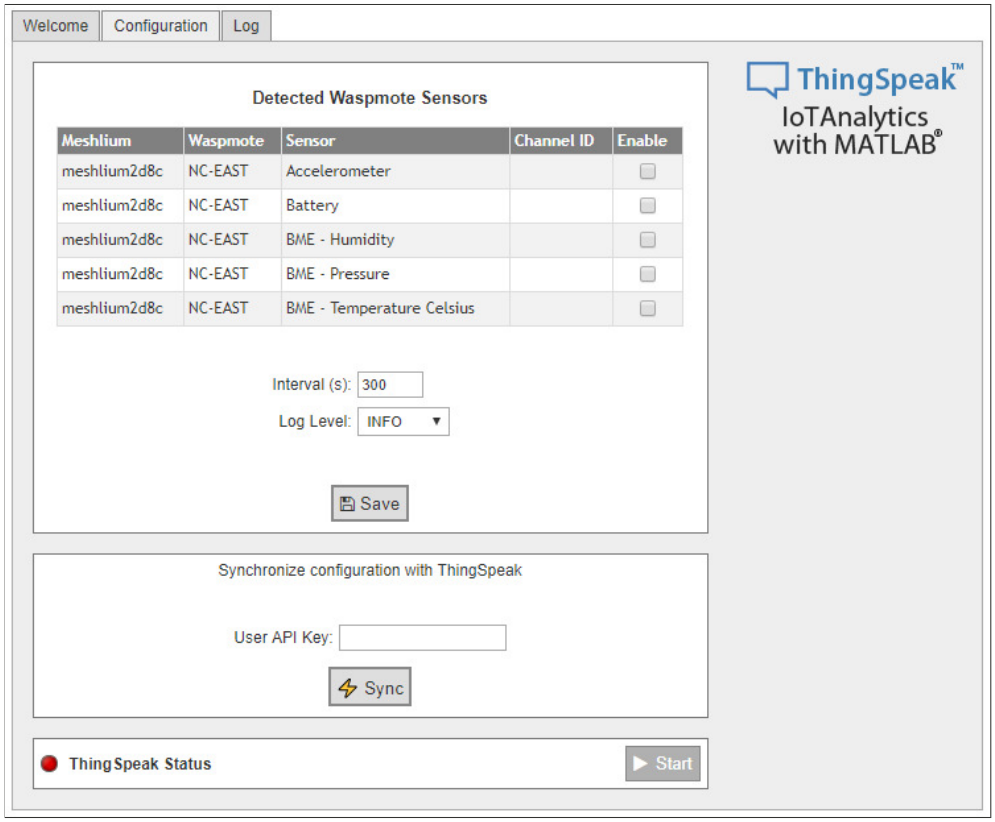


Figure : Configuration tab of the ThingSpeak Cloud Connector in the Meshlium Manager System

The ThingSpeak Cloud Connector has 3 basic operations:

1. **Save**: Select the devices to enable, and then save the settings to the Meshlium Manager System.
2. **Sync**: Synchronize changes to your ThingSpeak configuration. This operation creates new channels for sensors not previously enabled. You need to enter your ThingSpeak User API Key to sync.
3. **Start**: Enable storing your Meshlium data to ThingSpeak. As soon as you press **Start**, data from your Meshlium is sent to your ThingSpeak channels.

Additional Settings

- **Channel ID**: ThingSpeak stores data in channels. A channel is created for each unique sensor. The channel is private by default, but you can make it public. Use the channel link that appears after you synchronized with ThingSpeak to see channel contents, or go to My Channels at ThingSpeak.com (https://thingspeak.com/channels).
- **Enable**: Include data from this sensor when updating ThingSpeak. Each time you change the Enable value, you must save the configuration.
- **Interval**: Set the interval between updates to ThingSpeak. The smallest interval is 30 seconds. Updates happen in a batch, so this value is different than the sensor measurement frequency.
- **Log Level**: Choose the level of information written to the log.
- **User API Key**: Log in to www.thingspeak.com, and go to **Accounts > My Profile** (https://thingspeak.com/account/profile). Copy the key and paste it in the User API Key box. For security, the User API Key is not stored in the connector. You need to enter it every time you use the **Sync** button.

To set up your ThingSpeak connection

1. In the Configuration tab, select **Enable** 

for each sensor from which you want to record.
2. Select the **Interval** and **Log Level** 

.
3. Click **Save** 

. The ThingSpeak Cloud Connector indicates that synchronization is needed.
4. Retrieve your ThingSpeak User API key:
   - Log in to www.thingspeak.com.
   - Select **Account > My Profile** (https://thingspeak.com/account/profile).
   - Copy your User API Key to the ThingSpeak Cloud Connector.

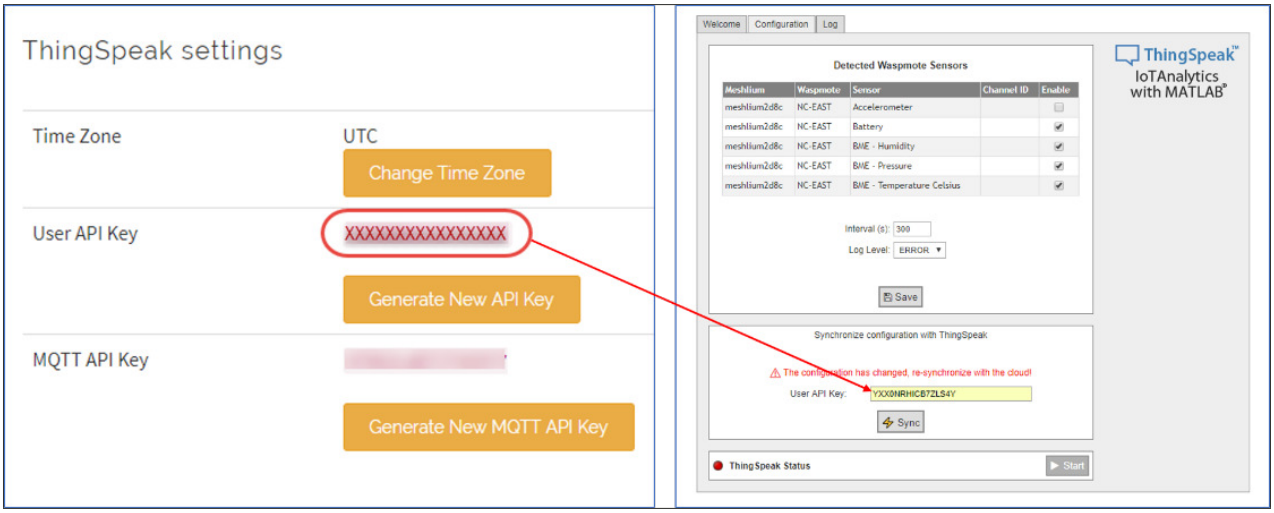
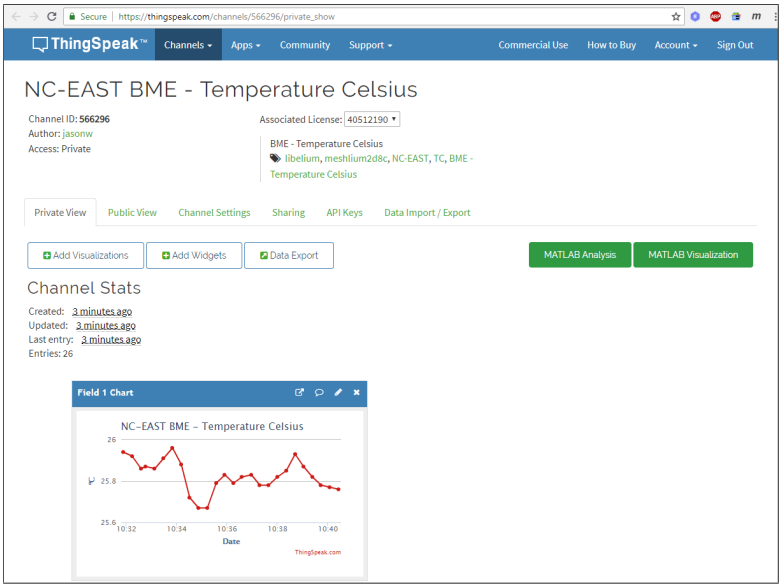


Figure : The ThingSpeak User API Key is required to synchronize your Meshlium settings

- Press **Sync** to create channels on ThingSpeak and map them to sensors on the Meshlium device.

5. Press **Start**. ThingSpeak starts logging the data. Your existing Waspmote data is visible in your ThingSpeak channels immediately.



6. Analyze and visualize your data with ThingSpeak:
   - **Regularize** or **smooth** your data to remove outliers. (https://www.mathworks.com/help/thingspeak/regularize-irregularly-sampled-data.html) (https://www.mathworks.com/help/thingspeak/remove-high-frequency-noise-in-measured-data.html)
   - **Analyze** data to find underlying insights and **predict trends**. (https://www.mathworks.com/matlabcentral/fileexchange/47049-analyzing-weather-data-from-an-arduino-based-weather-station) (https://www.hackster.io/matlab-iot/measure-and-analyze-tide-levels-with-thingspeak-and-matlab-efa405)
   - Use **standard visualizations** or create custom visualizations to showcase your data. (https://www.mathworks.com/help/thingspeak/Compare-Temperature-Data-from-Three-Different-Days.html).

Tips

- Each Waspmote must have a unique node name and each Meshlium must have a unique hostname.
- On your ThingSpeak channels, do not remove any tags or edit the metadata fields for the auto generated ThingSpeak Cloud Connecter channels. These fields are used to communicate with the Meshlium Manager System. You can change any other fields or settings.
- The maximum number of records selected in a single update is 200. If all your data is not being sent to ThingSpeak, decrease the interval to send updates more often.
- Use the channel **tags feature** to rapidly sort your sensor channels on ThingSpeak. Your channels are automatically tagged by the sensor name and Waspmote name. Enter a value and the channel view is filtered to show only channels with that tag.

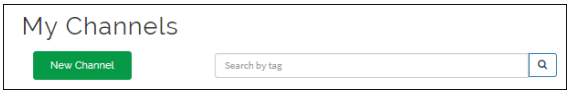


Figure : Use tag feature: https://www.mathworks.com/help/thingspeak/channel-settings.html#channels-search-by-tag

For example, entering the name of a Waspmote unit shows only channels associated with that Waspmote. You can add tags but do not remove the automatically generated tags.

- If a Waspmote fails and needs to be replaced, the existing ThingSpeak channels can be reused, just give the new Waspmote the same node name as the failed device.
- If you reset the API key of a channel, use the **Sync** button to update the new keys.
- If you accidentally delete a ThingSpeak channel, use the **Sync** button to create a new channel.

13. Device Connectors

The aim of this chapter is to introduce the user to the Meshlium's Device Connector functionality. This section will help you to connect your Meshlium to a 3rd party device platform.

3rd party certified connectors are linked with Meshlium by an IP interface.

What is a device platform?

Devices are equipments which could be easily correlated with Meshlium through an Ethernet or wireless connection. Devices perform actions like taking images, activation on systems, industrial control, etc. Interfacing with 3rd party devices allows Meshlium to execute actions manually or automatically responding to events detected in the info sent by Waspmote or Plug & Sense! and received on Meshlium.

Meshlium Device Connector

Meshlium runs the software necessary for implementing the analysis of its internal database and the control of devices. In other words, this software checks the occurrence of events and performs the rules specified to execute actions on the device. This software is called Device Connector.

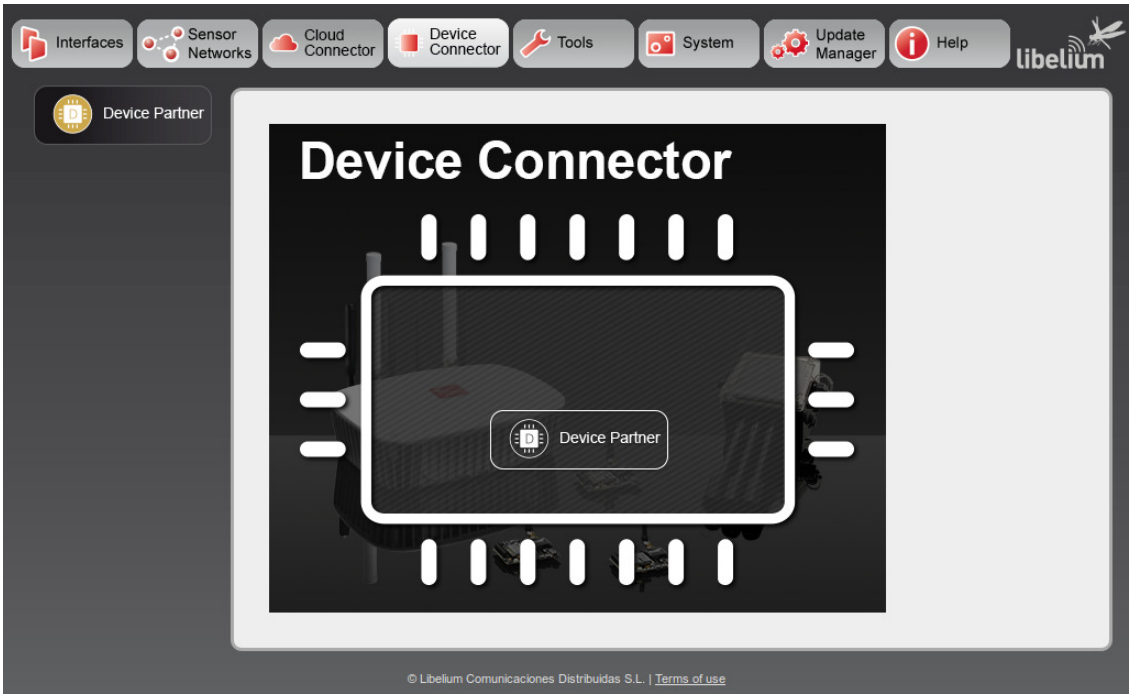


Figure : Device Connector main menu on the Manager System

13.1. Device Partners

13.1.1. Axis

Axis offers a wide variety of network videocameras and advanced analytic applications. More information: https://www.axis.com/global/en/products/network-cameras.

Thanks to this plugin, the sensor data received on Meshlium can trigger actions on Axis cameras.

Figure : Axis plugin

Configuration

Prior to configure Meshlium, make sure you set these parameters on the camera: IP, username, password and presets.

For information about how to configure your camera, please check the installation guide of your camera model.

The Axis plugin is located in:

Device Connector → Device Partner → Axis

In the "Configuration" panel, the user can set:

- **Log Level**: Generate log messages. From fewer to more details, the levels are: OFF, ERROR, INFO, DEBUG and REPORT. The option by default is OFF.
- **Execution interval**: Time interval between checking camera's rules.

In the "Add Camera" panel, the user can set:

- **Name**: Device rule name.
- **IP**: IP address of the camera.
- **User**: User name configured on the camera.
- **Password**: The password configured on the camera by the user (defined in the previous step).
- **PTZ Preset**: Preset of the camera to be used when the rule is executed. For obtaining a list of available presets on the camera you have to fill the parameters above (IP, user and password). They will be used for getting the presets when you press the button "Get Presets". After pushing the button, the list of presets will be shown and you will be able to select one of them.
- **Threshold**: Threshold in a sensor value for taking a picture. This field is not mandatory, but if you want to add a threshold, you have to select one:
  - Waspmote sensor: Available sensor which will trigger the rule.
  - Operation: Available operations are: greater, less, greater or equal, less or equal, equal than or different from.
  - Value: Value to be compared to the sensor selected.

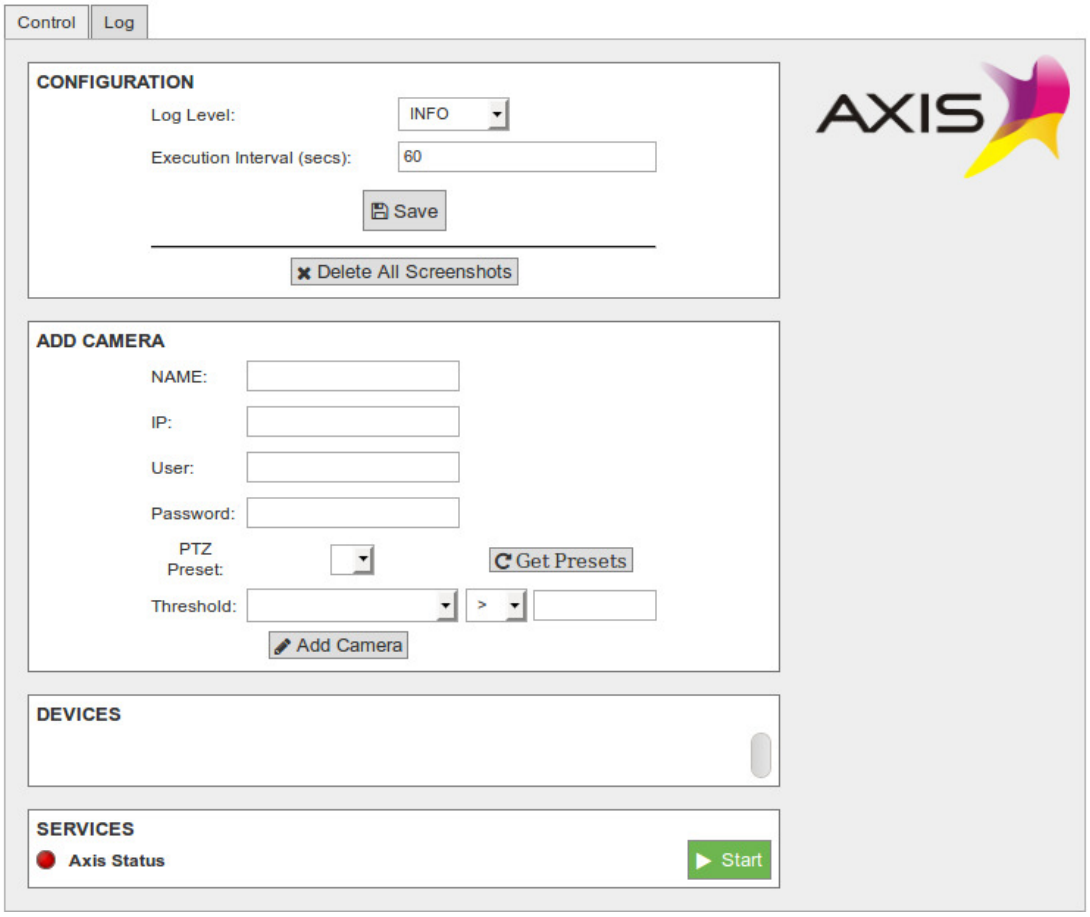


Figure : Axis configuration panel

When all the parameters have been configured, press the “Add Camera” button. This action will add a new rule in the “Devices” section.

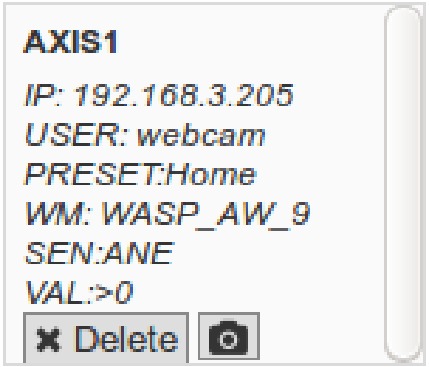


Figure : New device

Devices

In the “Devices” section you can see all the rules that will be evaluated for each device. Apart from checking that the rules are properly configured, you can execute 2 actions:

- **Delete**: Delete the device. It will be disappear from this section and the rule will not be evaluated.
- **Take a picture**: The camera will take a picture in the predefined preset.

For accessing to the pictures taken (both manual or automatic) you have to access to the FTP server of the Meshlium unit. The folder “axis” inside the FTP main directory contains the images taken. The image names will vary depending if the picture was taken manually or automatically:

- **Manually**: the name of the file will be **NAME_YYYY-MM-DD_HH-MM-SS.jpg**.
- **Automatically**: the name of the file will be **NAME_WASPNAME_SENSORNAME.jpg**.

Controlling synchronization

Once the devices are configured, the user can launch the Meshlium Axis service (“Start” button). The program will search for the received frames on the local database, and will check the rules configured. If a rule is evaluated positively, then a picture in the preset position of the rule will be taken and stored. The status indicator displays the current state, red means “stopped” and green means “running”.

Figure : Axis service is running

You can stop the Axis service anytime by clicking on the “Stop” button.

Figure : Axis service is stopped

14. Smartphone detection

Meshlium allows to detect iPhone and Android devices and in general any device which works with WiFi or Bluetooth interfaces.

These devices can be detected without the need of being connected to a specific Access Point, enabling the detection of any smartphone, laptop or hands-free car kit device which comes into the coverage area of Meshlium.

The idea is to be able to measure the amount of people and cars which are present in a certain point at a specific time, allowing the study of the evolution of the traffic congestion of pedestrians and vehicles.

Figure : Smartphone detection

Users have to do nothing to be detected as the WiFi and Bluetooth radios integrated in their smartphones periodically send a “hello!” message telling about their presence. The information read from each user contains:

- The MAC address of the wireless interface, which allows to identify it uniquely.
- The strength of the signal (RSSI), which gives us the average distance of the device from the scanning point.
- The vendor of the smartphone (Apple, Nokia, etc).
- The WiFi Access Point where the user is connected (if any) and the Bluetooth friendly name. Users no connected to an AP will be showed as “free users”.
- The Class of Device (CoD) in case of Bluetooth which allows us to differentiate the type of device (smartphone, hands-free, laptop, LAN/network AP). With this parameter we can differentiate among pedestrians and vehicles.

The coverage areas may be modified by changing the power transmission of the radio interfaces allowing the creation of different scanning zones from a few meters (in order to study a specific point) to dozens of meters (to study the whole street or even the entire floor of a shopping mall).

Applications related to shopping and street activities:

- Number of people passing daily in a street.
- Average time of the stance of the people in a street.
- Differentiate between residents (daily matches) and visitants (sporadic matches).
- Walking routes of people in shopping malls and average time in each area.

The Vehicle Traffic Monitoring is also another important application as understanding the flow and congestion of vehicular traffic is essential for efficient road systems in cities. Smooth vehicle flows reduce journey times, reduce emissions and save energy. Similarly the efficient flow of pedestrians in an airport, stadium or shopping center saves time and can make the difference between a good and a bad visit. Monitoring traffic - whether road vehicles or people - is useful for operators of roads, attractions and transport hubs.

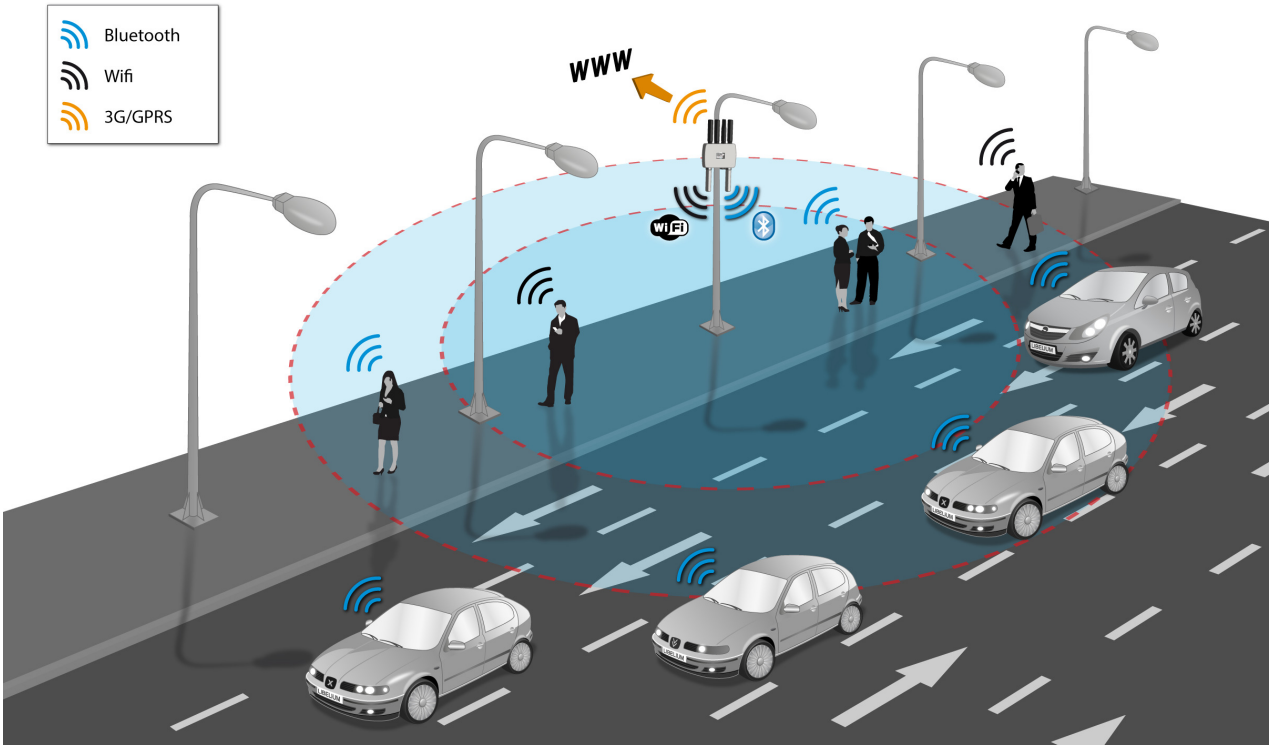


Figure : Vehicle Traffic Detection

Applications for Vehicle Traffic Detection:

- Number of people passing daily in a street.
- Average time of the stance of the people in a street.
- Differentiate between residents (daily matches) and visitants (sporadic matches).
- Walking routes of people in shopping malls and average time in each area.

The monitoring system can also be used to calculate the average speed of the vehicles which transit over a roadway by taking the time mark at two different points.

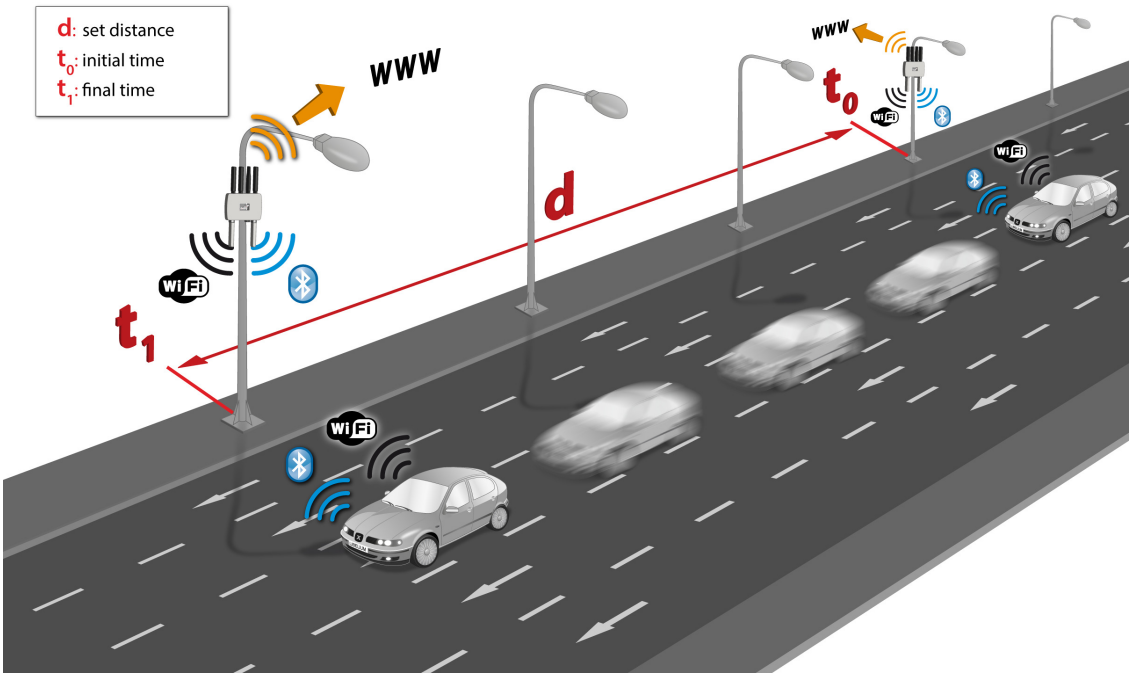


Figure : Calculate the average speed

14.1. Devices detected

Detection includes any of the last models even those that implement low consumption techniques when using the radio interfaces:

- iPhone (*all models*): 4, 4S, 5, 5S, 5C, iPad (2, 3, 4, Air, Mini, Retina).
- Android (*all models*): Nexus, Samsung Galaxy, LG, Sony Xperia, HTC, Motorola, Huawei, Asus...

Figure : Some of the supported smartphones

Vehicle Traffic Monitoring

Due to the reduction of the time between scanning intervals, now vehicle traffic detection rate has increased **from 50% to 80%** even at a speed of 100 km/h (62 miles/h)

- Monitor in real time the number of vehicles passing for a certain point in highways and roads.
- Detect average time of vehicle stance for traffic congestion prevention.
- Monitor average speed of vehicles in highways and roads.
- Provide travel times on alternate routes when congestion is detected.

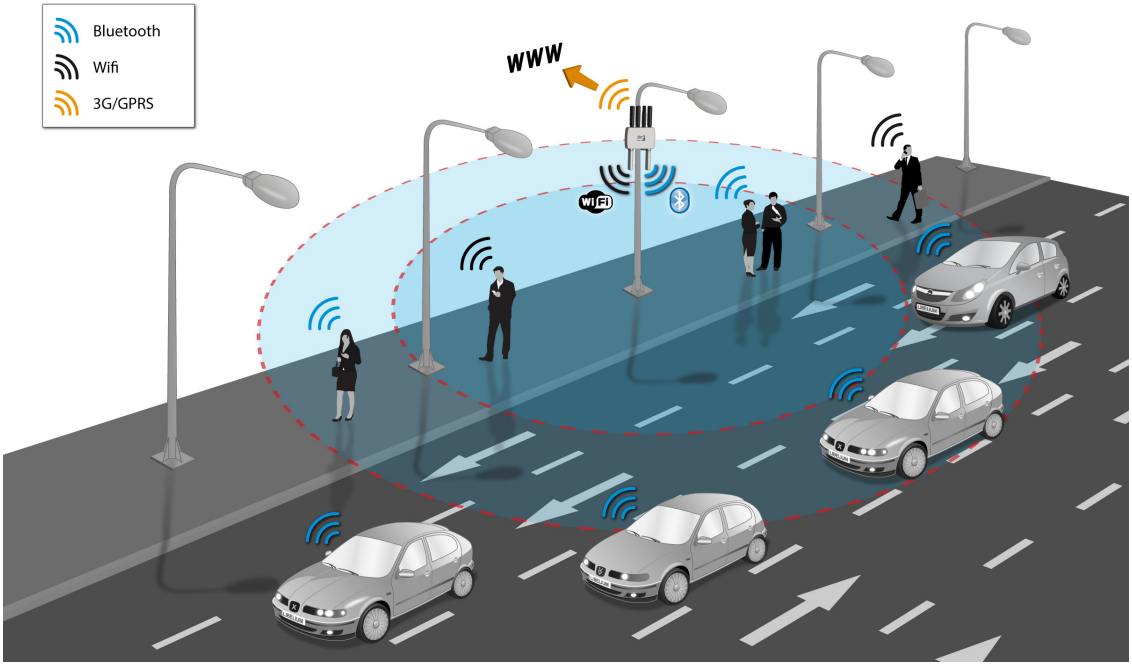


Figure : WiFi and Bluetooth scanning in the street

For Vehicle Traffic Monitoring applications, it is recommended to purchase the special pack of 2 directional antennas which will extend the range of WiFi and Bluetooth scanning in the required direction. The size of one antenna is 17 x 17 x 3 cm. Each one weights about 350 g. The antennas are 14 dBi and come with the needed mounting system, 3 m cables and screw adapters.

Figure : Directional antenna for Meshlium Scanner

Do the users need to have a specific app installed or interact somehow to be detected?

No, the scan is performed silently, Meshlium just detects the "beacon frames" originated by the WiFi and Bluetooth radios integrated in the smartphones. Users just need to have at least one of the two wireless interfaces turned on.

How do we differentiate if the Bluetooth device detected is a car's hands-free or a smartphone?

In the scanning process each Bluetooth device gives its "Class of Device" (CoD) attribute which allows to identify the type of service it gives. We can differentiate easily the CoD's generated by the car's hands-free from the people's phone ones.

How do I control the inquiry area?

In the Bluetooth inquiry there are seven different power levels which go from -27 dBm to 3 dBm in order to set different coverage zones from 10 to 50m. In both WiFi and Bluetooth radios these zones can also be increased or decreased by using a different antenna for the module as it counts with a standard N-Male connector. The default antenna which comes with the scanning modules is an omnidirectional antenna with a gain of 5dBi.

How do I calculate the distance of any of the devices detected?

In the inquiry process we receive the MAC address of the Bluetooth device along with the Received Signal Strength Indicator (RSSI) which gives us the quality of the transmission with each device. RSSI values usually go from -40 dBm (nearest nodes) to -90 dBm (farthest ones). In the tests performed Bluetooth devices at a distance of 10 m reported -50 dBm as average, while the ones placed at 50 m gave us an average of -75 dBm.

What about privacy?

The anonymous nature of this technique is due to the use of MAC addresses as identifiers. MAC addresses are not associated with any specific user account or mobile phone number not even to any specific vehicle. Additionally, the "inquiry mode" (visibility) can be turned off so people have always chosen if their device will or will not be detectable.

How do the Bluetooth, WiFi and ZigBee radios coexist without causing interferences with each other?

WiFi, XBee and Bluetooth work in the 2.4GHz frequency band (2.400-2.480 MHz), however, the Bluetooth radio integrated in Meshlium uses an algorithm called Adaptive Frequency Hopping (AFH) which improves the common algorithm used by Bluetooth (FHSS) and enables the Bluetooth radio to dynamically identify channels already in use by XBee and WiFi devices and to avoid them.

Anyway, in the case of sending 802.15.4 frames from Waspmote or Plug & Sense! to a Meshlium Scanner equipped with XBee-PRO 802.15.4, it is recommended to perform re-tries in the sender application, just to minimize possible interferenc

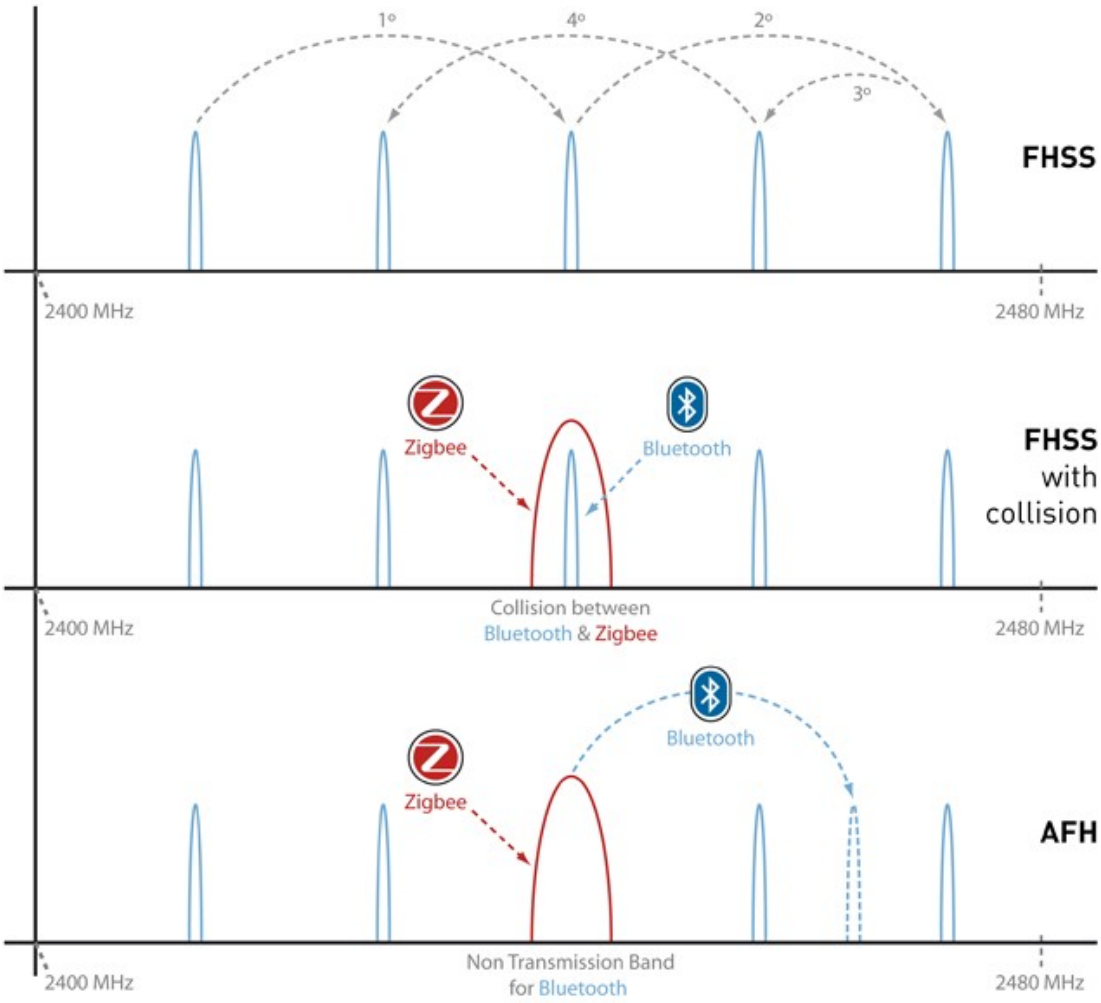


Figure : Bluetooth, WiFi and ZigBee radios coexist

Under which conditions do you get a 95% detection rate of devices?

A set of conditions must be respected to keep the detection rate high.

The devices to be detected must be some meters away from the Scanner and must remain some seconds inside the coverage area to give time to the system to detect them.

The setup of the WifiScan feature in Meshlium is 40 seconds of scan span. This means Meshlium Scanner listens for 40 seconds and then stores the results of the scan.

Android and iOS devices have a special option to disable WiFi connection when the user locks the screen in order to save battery. All cases are studied here. This option changes with the iOS version, but will be present in the majority of iOS devices. It makes iOS devices

When a device is connected to a WiFi Access Point it is easier to detect, as it needs to send radio packets to allow communication.

The results not Android or iOS devices may vary depending on the type of system. Usually, APs are detected easily, as they broadcast the SSID. Hidden SSID are detected too. The only APs that can be hard to detect are the APs that do not broadcast their presence. This APs can only be detected when there is traffic from connected devices.

Regarding other WiFi devices, the individual behaviour will define if they are detectable. As a general rule, every device that broadcasts beams or is connected generating traffic will be detected.

Our tests results are shown in these tables:

- **Android**:

	Screen ON	Screen OFF (power saving off)	Screen ON (power saving on)
WiFi radio OFF	NO	NO	NO
WiFi radio ON (not connected to an AP)	YES (almost every scan cycle)	NO (most of scan cycle)	NO
WiFi radio ON (connected to an AP)	YES	YES	NO

- **iOS**:

	Screen ON	Screen OFF (power saving off)
WiFi radio OFF	NO	NO
WiFi radio ON (not connected to an AP)	YES (after several scans)	SOMETIMES (after a random number of scans)
WiFi radio ON (connected to an AP)	YES	YES (after several scans)

Bluetooth scanning, unlike WiFi scanning, is based on polling, and not in passive listening. This makes Bluetooth detection slower and left the device the chance of avoiding detection (it just needs to ignore the polling request).

Nevertheless, Bluetooth scan is still useful in some applications, like car detection, as most of modern cars have a Bluetooth hands-free device, and these devices are most of the time listening for connections.

Any smartphone can be configured to be visible (or not) by other Bluetooth devices. Putting this option as “NOT VISIBLE” will make the smartphone undetectable by any other Bluetooth device, which includes Meshlium Scanner. Note that some latest-technology hands-free devices implement the “not visible” mode too.

When the Bluetooth interface is set as visible, the phone will be listening for incoming queries. This way the device can be scanned. The visibility setup may be different in different devices. Some of them activate visibility for a limited time (usually 30 seconds), some others have a manual control to enable/disable the visibility.

Different Android and iOS versions have different behaviours about Bluetooth visibility. In most of the modern versions, Bluetooth visibility is disabled when the screen is locked (or even when the user exits the Bluetooth configuration menu). There is no way to detect an Android or iOS phone with the screen off, which makes it very difficult to scan Android or iOS devices in a real environment.

There are a lot of types of Bluetooth devices. Most of slave Bluetooth devices are designed to wait for incoming connections. This makes highly possible to detect devices like hands-free car kits, headsets, HID, etc.

The scanning time is more important in Bluetooth as the devices need some time to reply to the queries.

Device name is not always obtained, as some devices take some time to reply to the name queries. Nevertheless, the device can be easily identified by its MAC address.

How can I calculate the total number of people from the number of detected devices?

It depends. Not all the people have a smartphone. Also, not all the people switch WiFi and/or Bluetooth radios on their smartphones. It all depends on so many economic, social and cultural factors. The percentage of people with WiFi or Bluetooth on depends on the scenario where they are too. For example, if a Meshlium Scanner is installed in a college campus which provides free WiFi service, many students will be detected because they will probably keep their smartphones, tablets or notebooks with WiFi on. The same would happen in a mall, airport or hotel with free WiFi.

Besides, consider that not all the people who could be detected will remain enough time inside of the coverage area of Meshlium Scanner.

Also, keep in mind that some people can carry several WiFi or Bluetooth devices. For example, a driver with smartphone in his pocket and a Bluetooth device in his car can be detected as 2 different users by Meshlium.

To sum up, in Libelium we consider that the total number of people can be approximated multiplying the number of detected devices by a factor, from 3 or 5:

3*Detected devices < Total people < 5*Detected devices

It all depends on a number of variables. The administrator of Meshlium Scanner can perform real tests in order to find the exact value of this factor in the specific scenario under study.

14.2. WiFi Scanner

14.2.1. Concepts

The additional 2nd WiFi radio integrated in Meshlium Scanner allows to scan WiFi devices in a range of action up to 200 m (depending on the line of sight conditions). Meshlium Scanner can detect devices in the 2.4 GHz and 5 GHz frequency bands.

The idea is to search for WiFi devices in a defined interval which can be configured. Meshlium will get the **MAC address**, information about the detected **Device**. Regarding these devices, we can distinguish Access Points and Clients .In the case of each client, Meshlium gets which Access Point the device is connected to (if any). Also, the signal strength (**RSSI**) of the device along with a **timestamp** which identifies when the scan was performed. The timestamp is always stored in UTC to avoid inconsistencies (regardless of the time zone selected in Meshlium). It is important to set the correct time in the System before starting with the storage of the data. See the Time Synchronization in the System section.

As extra information, the System also identifies the **Vendor** of the WiFi devices using its MAC address and if the information is synchronized to the external database (**Sync**).

Example of information scanned:

B ID	Sync	Timestamp	MAC	Device	RSSI	Vendor
53483	0	2012-04-24 07:56:25	C4:2C:03:96:0E:4A	(not detected)	69	Apple
53482	1	2012-04-24 09:11:26	D8:2A:7E:10:1E:63	libelium_wsn1	60	Nokia Corporation

Wifi scanner configuration menu is located in:

Tools → Wifi Scan

In this section we can select the Scanning Time from a drop-down list. This time specifies how many seconds the scanner will spend searching. After each scanning process, the system performs a pause of one second before starting again.

The Scanning Time must be trimmed in order to avoid that a temperature of 70 oC is reached in the Meshlium's microprocessor. See chapter "Internal temperature sensor" to know how to monitor the microprocessor's temperature.

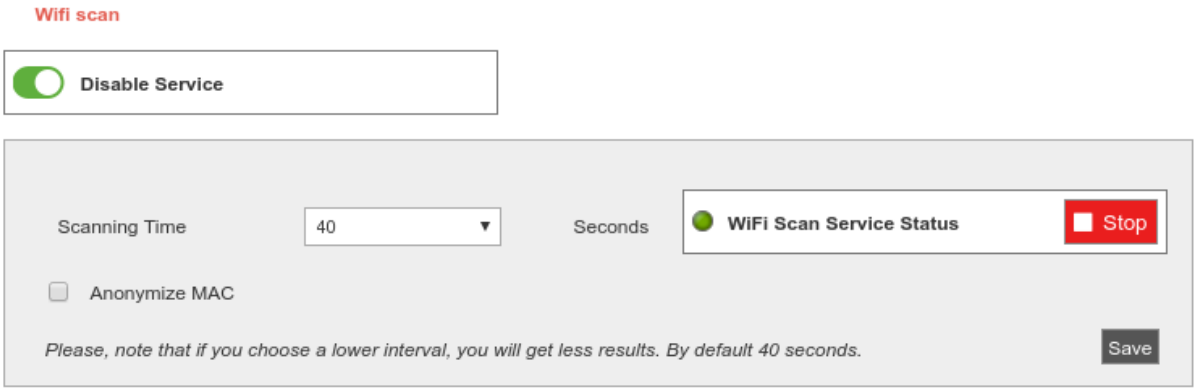


Figure : Configuring WiFi Scanner

We can also activate the anonymization of the MAC addresses. This option will store the MAC address encoded with an MD5 hash. The hash will be consistent in the same day, but will change from one day to another. This system allows to follow a particular user in the same day, but keeps the privacy of the user, not storing the real MAC of the device and not allowing to track a user more than one day.

From this section the user can start and stop the service from the button next to the status indicator.

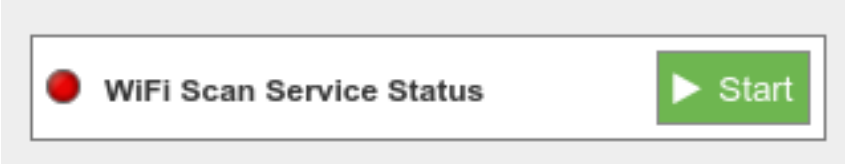


Figure : WiFi Scanner was stopped

If the user manually stops the service, it will be automatically relaunched upon reboot. In order to completely disable service, the user have to click on the slider "Disable Service". This will stop the service and avoid it to run upon reboot. Setup cannot be changed when disabled, but already stored data is available to be shown.

To enable the service again, click on the slider "Enable Service".

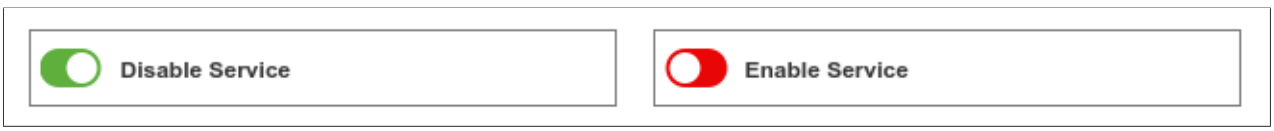


Figure : Enable and Disable controls

It is possible to perform two different storage options with the data captured:

- **Local database**: This is always used.
- **External database**: The data is synchronized to an external database from the local database.

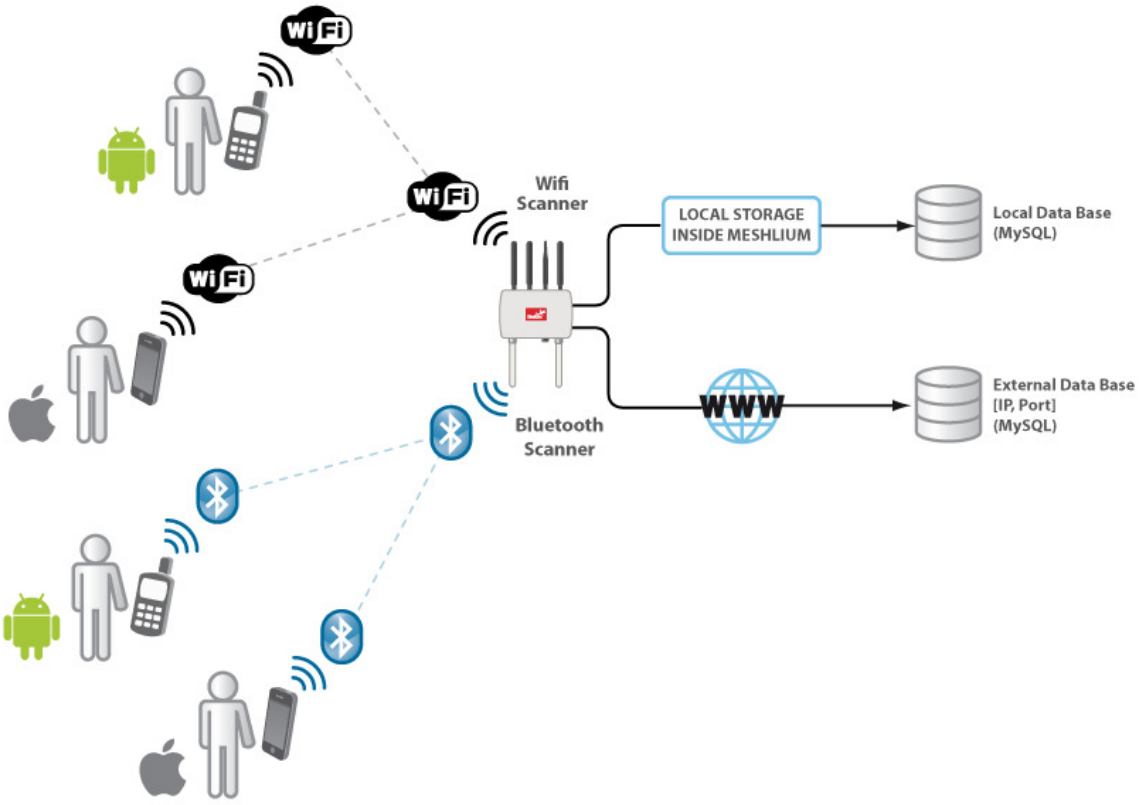


Figure : WiFi Scanner storing options

14.2.2. Local database

Meshlium has a MySQL data base up and running which is used to store locally the information captured. In the "Local Data Base" tab you can see the connection parameters.

- **Database**: MeshliumDB
- **Table**: wifiScan
- **IP**: localhost
- **Port**: 3306
- **User**: root
- **Password**: libelium2007

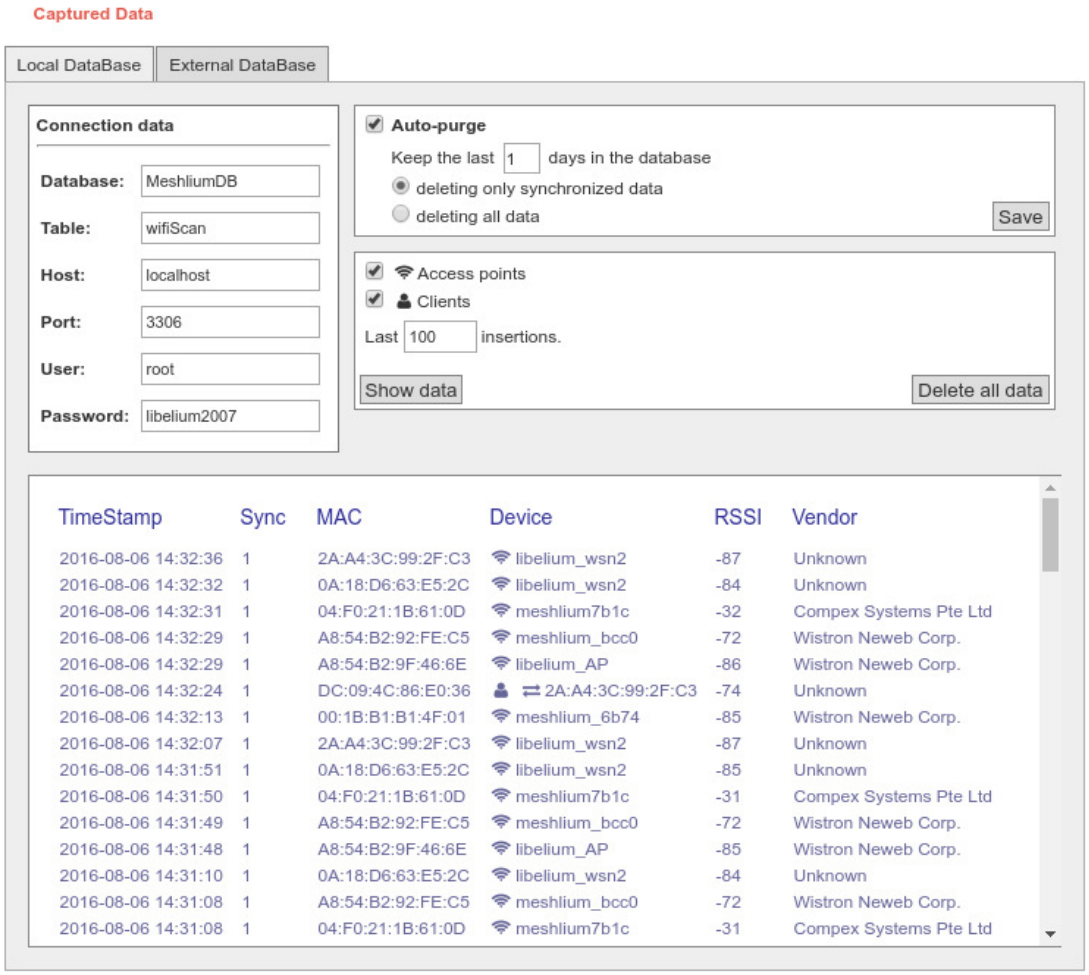


Figure : Local database for WiFi Scanner

At any time you can see the last "x" records stored, filtered by access points or clients. Just set how many and what kind of insertions you want to see and press the "Show data" button. The maximum number of data to display is 500.

The data from the database can be deleted pressing the button "Delete all data". Be careful, as this option deletes all the information of WiFi scans in the local database.

There is an option to program an automatic purge in the database every day, keeping the information in the database the days you specify. Furthermore, if you intend to configure the external database, you can choose if you want to delete only synchronized data or everything, taking care of the days established before.

14.2.3. External database

Meshlium can synchronize all the WiFi devices information stored in the local database to an external MySQL database managed by the user.

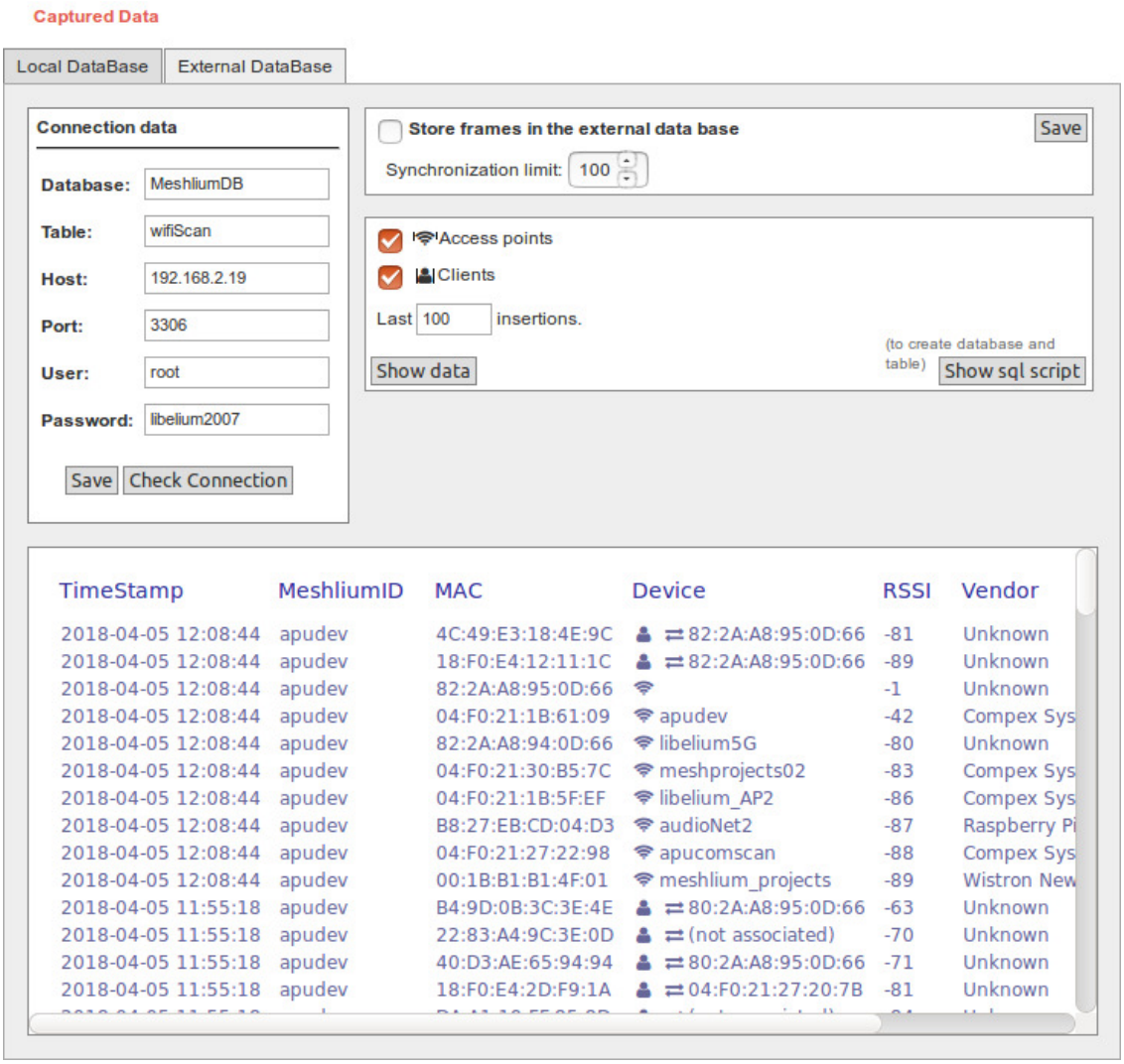


Figure : External database tab

In this tab the user can:

- Setup the parameters of the external database and check the connection.
- Enable or disable the synchronization.
- Select the number of fields sent per synchronization iteration.
- Show last data inserted in the external database (up to 500 data).
- Show the SQL script used to create the database and table needed for the synchronization.
- Mark all data in the local database as synchronized so it will not be sent to the external database.

The steps to setup the synchronization are:

- Before configuring anything, make sure you have a MySQL database working under your control. Make sure the database listen to connections in an external IP.
- Press the “Show sql script” button, copy the SQL code. You can modify user, password, database name and table, as long as you change the setup of the connection to match.

Just copy and paste:

```
CREATE database MeshliumDB;


CREATE TABLE `wifiScan` (
  `ID_frame` int(11) NOT NULL AUTO_INCREMENT,
  `TimeStamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `MAC` varchar(17) COLLATE utf8_unicode_ci NOT NULL,
  `SSID` varchar(32) COLLATE utf8_unicode_ci NOT NULL,
  `RSSI` varchar(4) COLLATE utf8_unicode_ci NOT NULL,
  `Vendor` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `Type` varchar(45) COLLATE utf8_unicode_ci NOT NULL,
  `AP` varchar(17) COLLATE utf8_unicode_ci NOT NULL,
  `MeshliumID` varchar(150) COLLATE utf8_unicode_ci NOT NULL default 'meshlium
  PRIMARY KEY (`ID_frame`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1


GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'passw';
```

Figure : SQL script

- Enter the connection settings and press “Save” button. You can check the connection now to ensure the settings are correct.
- Enable the service with the checkbox and save.

The synchronization service runs every 60 seconds and synchronizes up to 100 data every loop. The service synchronizes first newer data, as it is more relevant for decision making. This could make data in external database to be out of order. As every data has a timestamp, this should not be a problem for using the data in any external application.

14.3. Bluetooth Scanner

14.3.1. Concepts

This Bluetooth radio integrated in Meshlium Scanner allows to scan Bluetooth devices in a range of action up to 200m depending on the line of sight conditions.

The idea is to search for Bluetooth devices in a defined interval which can be configured. Meshlium will get the **MAC address**, the **Bluetooth ID** and the **RSSI** of the devices along with a **timestamp** which identifies when the scan was performed. The timestamp is always stored in UTC to avoid inconsistencies (regardless of the time zone selected in Meshlium). It is important to set the correct time in the System before starting with the storage of the data. See the Time Synchronization in the System section.

Other interesting parameters the system also detects are the **Class of Device (CoD)** which allows us to differentiate the type of device (smartphone, hands-free, laptop, LAN/Network AP) and the **Vendor** of the Bluetooth devices using its MAC address.

With these parameters we can differentiate among pedestrians and vehicles.

B ID	Timestamp	MAC	ID	RSSI	CoD	Vendor
45400	2012-05-16 16:18:12 07:56:25	00:26:7e:5f:3c:18	myCar	-72	Handsfree	PARROT SA
78005	2012-04-20 12:59:27 09:11:26	D8:2A:7E:0E:C3:10	Tropic	-85	Smartphone	Nokia Corporation

Wifi scanner configuration menu is located in:

Tools → Bluetooth Scan

In this section we can configure the Scanning Type which specifies the use of our Bluetooth Scanner:

- Indoor type is recommended to scan static devices or devices with slow movement (offices, malls, etc). This option retrieves device names after about 15 seconds scanning.
- Outdoor type focus on devices which stay a brief period of time in our Bluetooth action range (roads, highways,...). This option does not ask the device name and the scanning period is about 45 seconds.

In both types, there is a second between two consecutive scans.

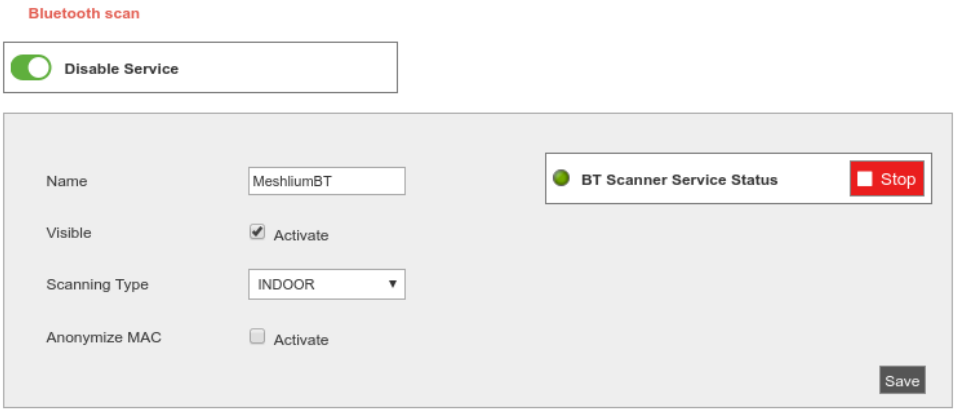


Figure : Configuring Bluetooth Scanner

We can also activate the anonymization of the MAC addresses. This option will store the MAC address encoded with an MD5 hash. The hash will be consistent in the same day, but will change from one day to another. This system allows to follow a particular user in the same day, but keeps the privacy of the user, not storing the real MAC of the device and not allowing to track a user more than one day.

From this section the user can start and stop the service from the button next to the status indicator.

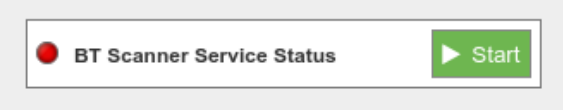


Figure : Bluetooth Scanner was stopped

If the user manually stops the service, it will be automatically relaunched upon reboot. In order to completely disable service, the user have to click on the slider "Disable Service". This will stop the service and avoid it to run upon reboot. Setup cannot be changed when disabled, but already stored data is available to be shown.

To enable the service again, click on the slider "Enable Service".

Figure : Enable and Disable controls

***Note**: Last versions of Android and iOS devices may need the Bluetooth Setup Screen be activated to be detected.*

We have two different storage options for the data captured:

- **Local database**: This is always active.
- **External database**: This synchronizes local database data to an external MySQL database.

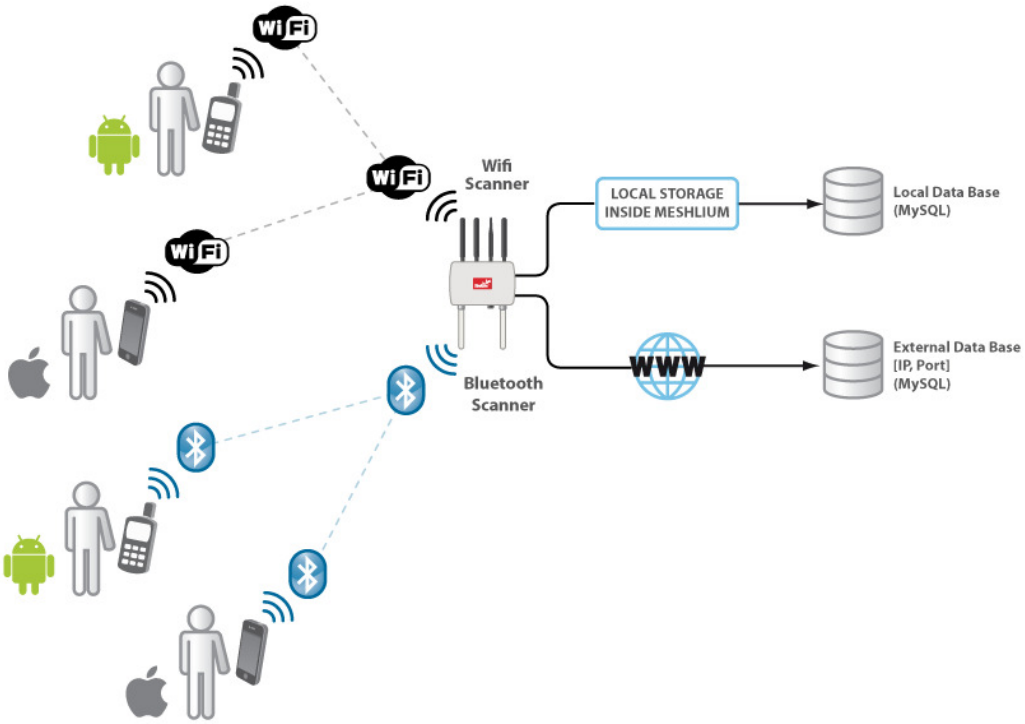


Figure : Bluetooth Scanner storing options

14.3.2. Local database

Meshlium has a MySQL database up and running which is used to store locally the information captured. In the "Local Data Base" tab you can see the connection parameters.

- **Database**: MeshliumDB
- **Table**: bluetoothData
- **IP**: localhost
- **Port**: 3306
- **User**: root
- **Password**: libelium2007

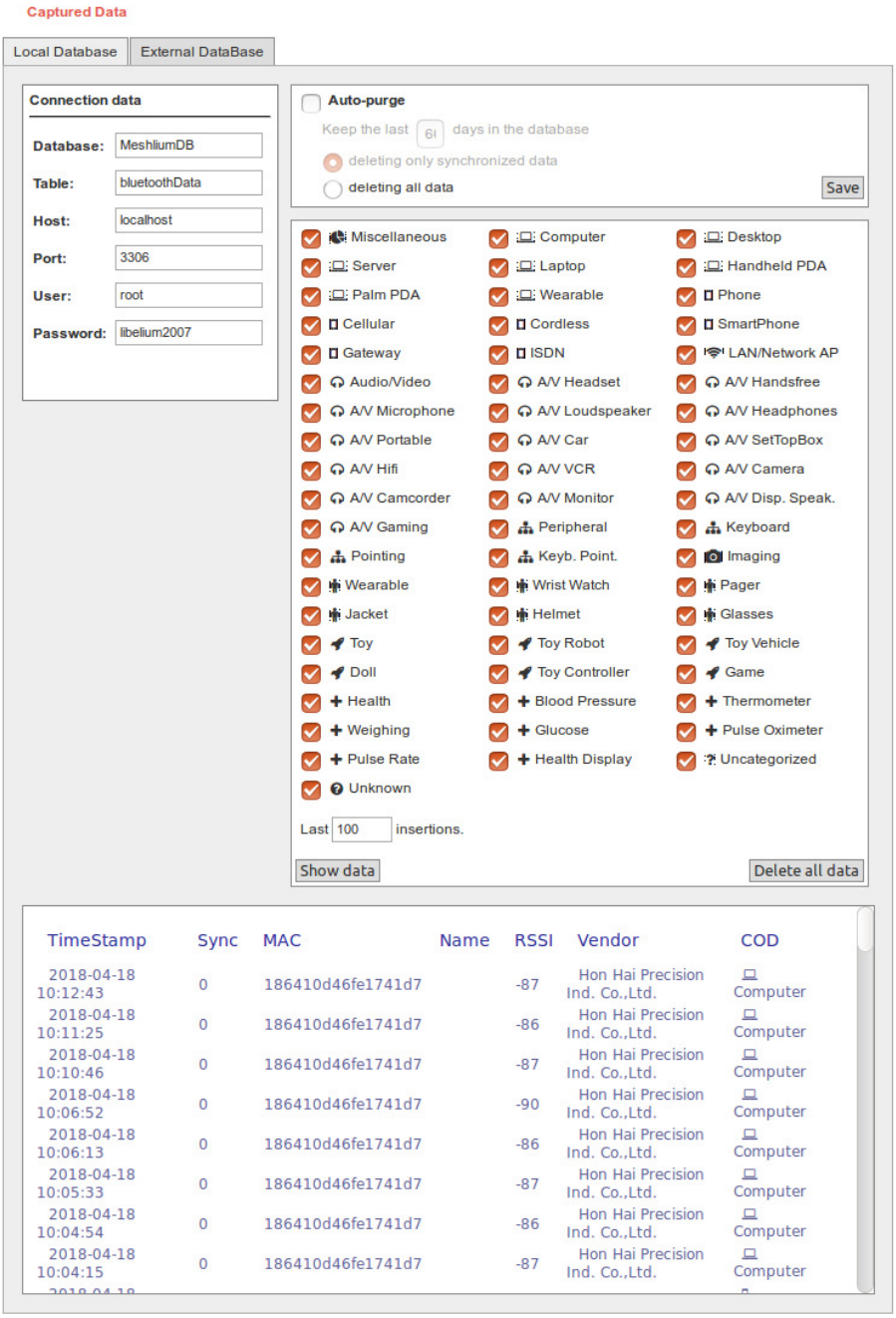


Figure : Local database for Bluetooth Scanner

At any time you can see the last "x" records stored, filtered by access points or clients. Just set how many and what

kind of insertions you want to see and press the “Show data” button. The maximum number of data to display is 500.

The data from the database can be deleted pressing the button “Delete all data”. Be careful, as this option deletes all the information of Bluetooth scans in the local database.

There is an option to program an automatic purge in the database every day, keeping the information in the database the days you specify. Furthermore, if you intend to configure the external database, you can choose if you want to delete only synchronized data or everything, taking care of the days established before.

14.3.3. External database

Meshlium can synchronize all the WiFi devices information stored in the local database to an external MySQL database managed by the user.

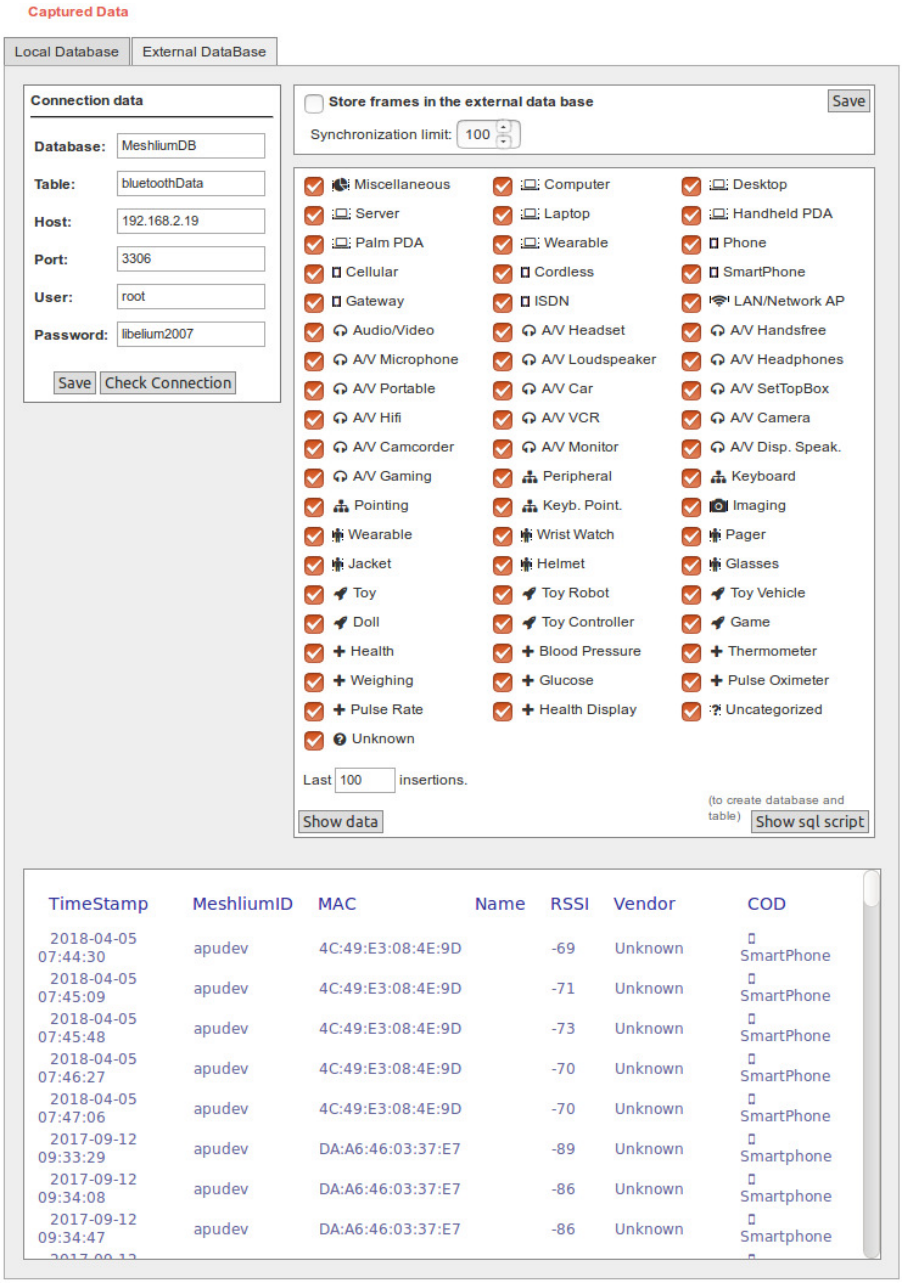


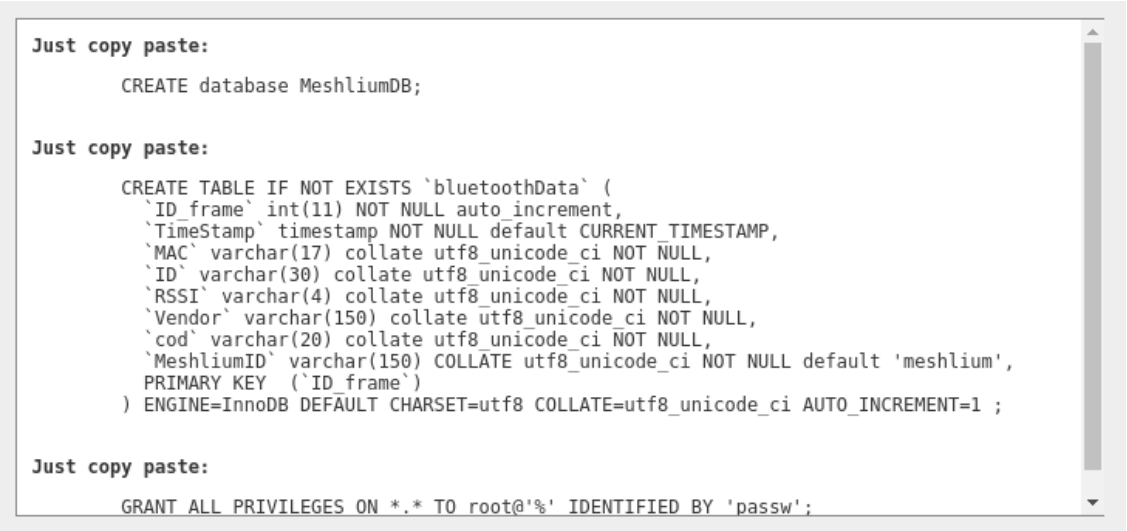
Figure : External database tab

In this tab the user can:

- Setup the parameters of the external database and check the connection.
- Enable or disable the synchronization.
- Select the number of fields sent per synchronization iteration.
- Show last data inserted in the external database (up to 500 data).
- Show the SQL script used to create the database and table needed for the synchronization.
- Mark all data in the local database as synchronized so it will not be sent to the external database.

The steps to setup the synchronization are:

- Before configuring anything, make sure you have a MySQL database working under your control. Make sure the database listen to connections in an external IP.
- Press the “Show SQL script” button, copy the SQL code. You can modify user, password, database name and table, as long as you change the setup of the connection to match.



```
Just copy paste:

        CREATE database MeshliumDB;


Just copy paste:

        CREATE TABLE IF NOT EXISTS `bluetoothData` (
          `ID_frame` int(11) NOT NULL auto_increment,
          `TimeStamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
          `MAC` varchar(17) collate utf8_unicode_ci NOT NULL,
          `ID` varchar(30) collate utf8_unicode_ci NOT NULL,
          `RSSI` varchar(4) collate utf8_unicode_ci NOT NULL,
          `Vendor` varchar(150) collate utf8_unicode_ci NOT NULL,
          `cod` varchar(20) collate utf8_unicode_ci NOT NULL,
          `MeshliumID` varchar(150) COLLATE utf8_unicode_ci NOT NULL default 'meshlium',
          PRIMARY KEY  (`ID_frame`)
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;


Just copy paste:

        GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'passw';
```

Figure : SQL script

- Enter the connection settings and press “Save” button. You can check the connection now to ensure the settings are correct.
- Enable the service with the checkbox and save.

The synchronization service runs every 60 seconds and synchronizes up to 100 data every loop. The service synchronizes first newer data, as it is more relevant for decision making. This could make data in external database to be out of order. As every data has a timestamp, this should not be a problem for using the data in any external application.

15. Tools

15.1. Fresnel calculator

The Fresnel Zone is the space which should be empty of objects in a wireless transmission between two points to get the maximum throughput and transmission quality. Here you can find a tool in order to calculate when choosing the right points for your nodes.

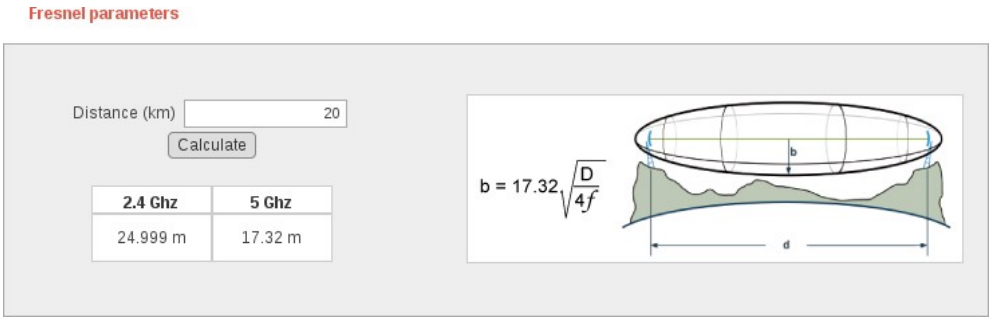


Figure : Fresnel calculator plugin

15.2. Iperf

This tool lets you know the real bandwidth between Meshlium and an iperf server. This plugin uses the correct interface in local networks and uses the default gateway for external networks. The default gateway is 4G/LTE if connected or Ethernet otherwise.

To use the tool, enter the IP address or the host of the iperf server. Iperf v3 is used so ensure the server is compatible with that version.

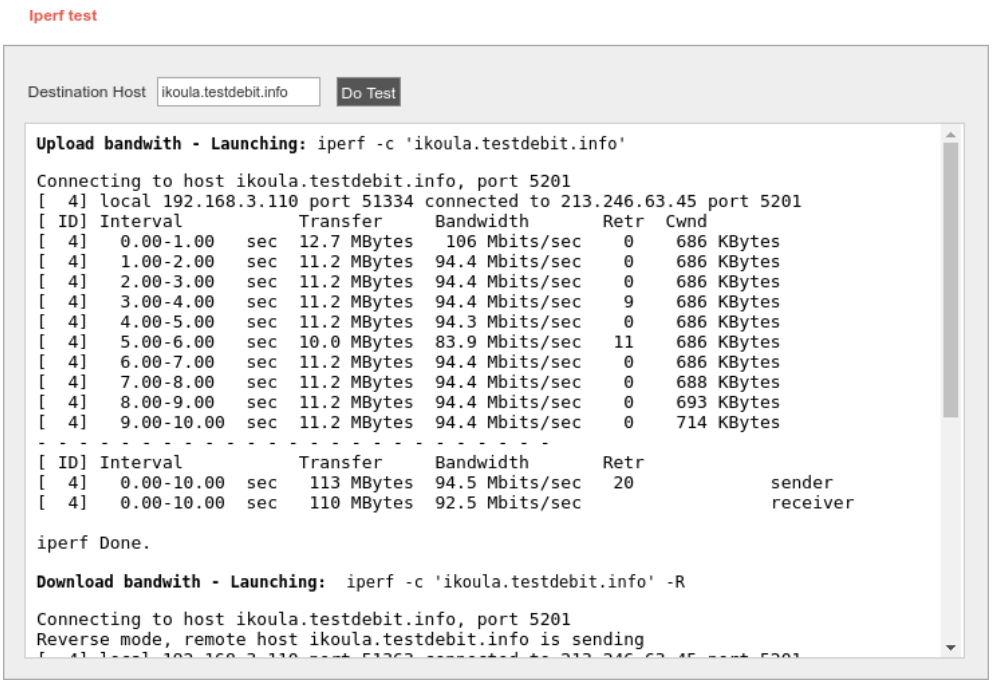


Figure : Iperf plugin

15.3. Ping

It lets you test if you can reach a certain IP or Hostname through a specific network Interface: Ethernet (IPv4), Ethernet (IPv6), WiFi AP, and 4G/LTE.

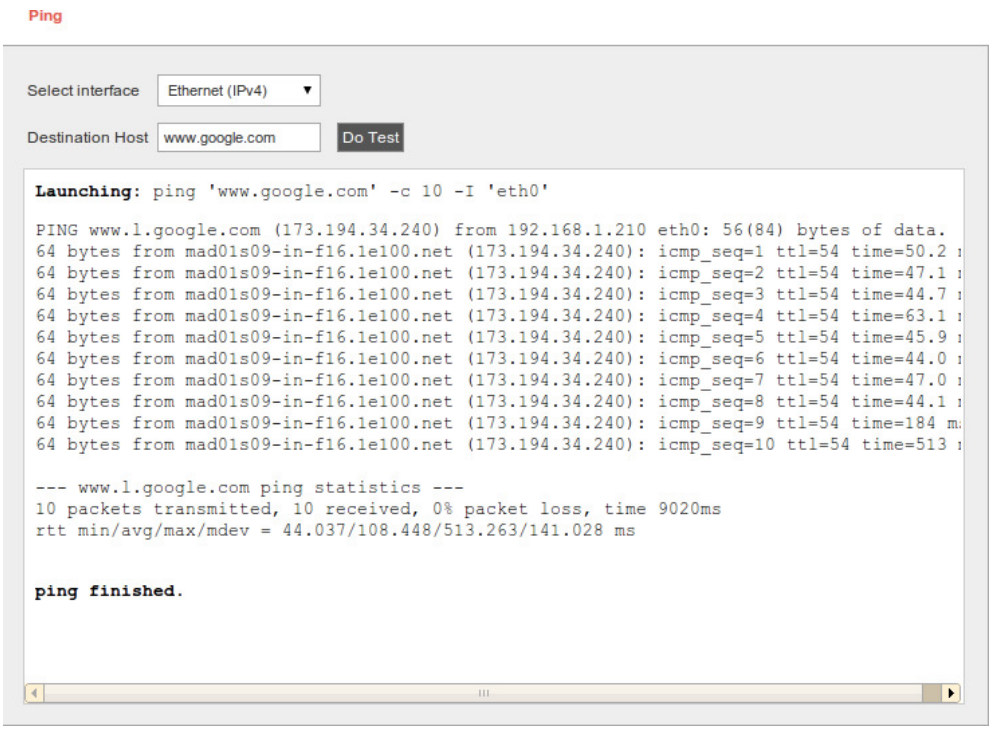


Figure : Ping plugin

Meshlium can also perform this test over IPv6 on Ethernet interface.

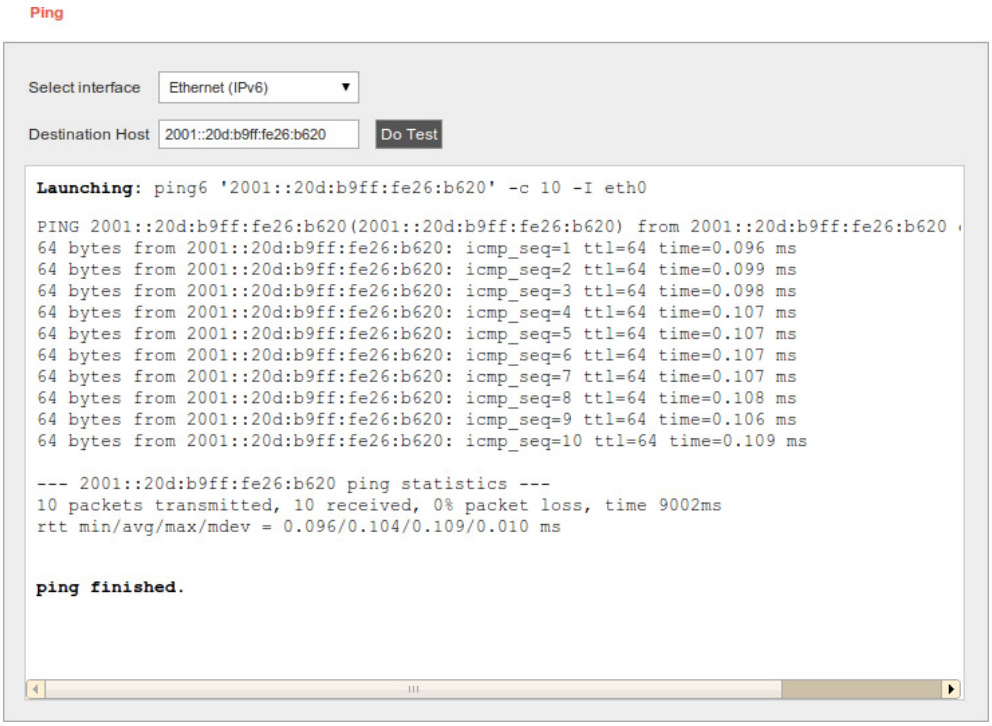


Figure : IPv6 ping

15.4. Traceroute

Another interesting tool to discover the path of the communication between Meshlium and the selected host.

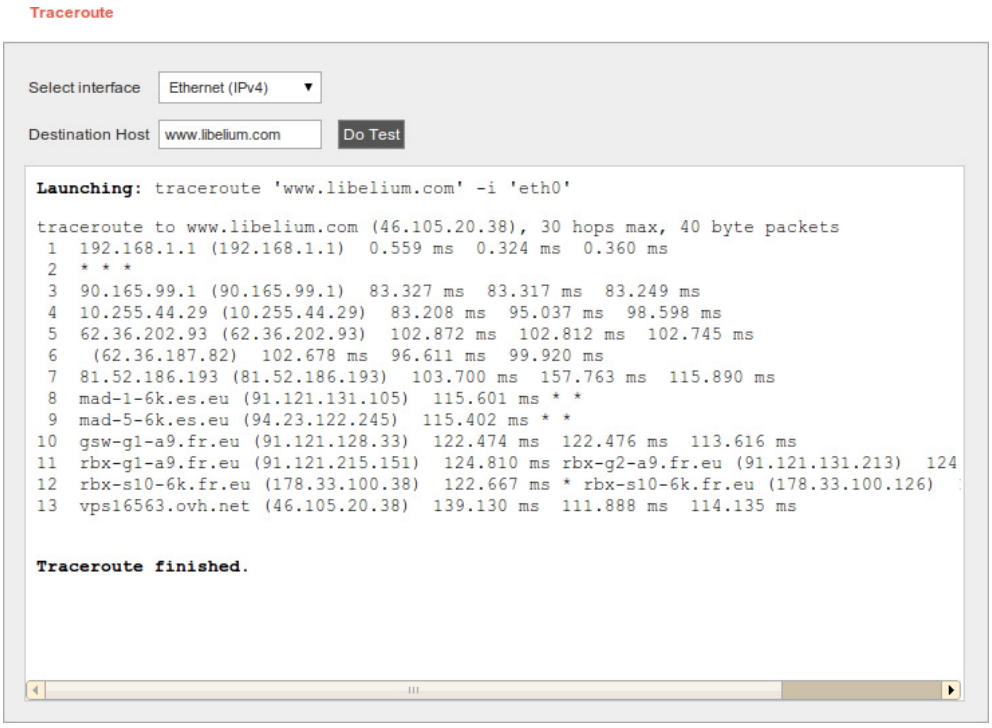


Figure : Traceroute plugin

Meshlium can also perform this test over the Ethernet (IPv6) interface.

15.5. Netstat

Discover which connections IPv4-Port (tcp), and IPv6-Port (tcp6) are active.

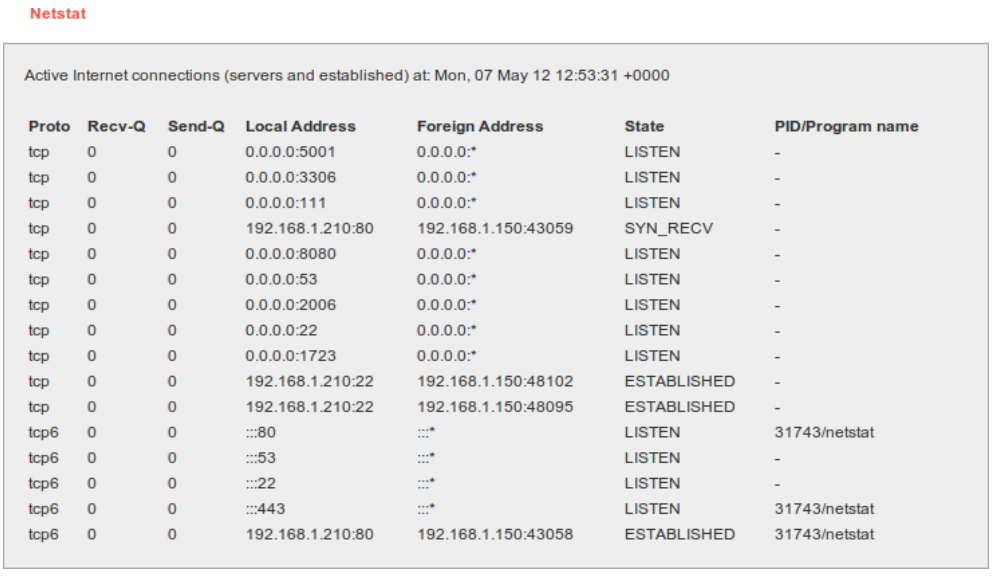


Figure : Netstat information

15.6. GPS

15.6.1. Concepts

Meshlium can integrate a GPS receiver which allows to know the exact location of the router any time. It is specially interesting for mobile and vehicular applications and when setting long range links as the GPS position also gives information about the height of each point so the Fresnel Zone can be accurately known.

The GPS module gives us information about:

- latitude.
- longitude.
- height (meters).
- speed (km/h).
- date/time.

Data captured form GPS is stored in the local database. In addition, the data can be synchronized to an external MySQL database to be used in other systems. The GPS data is stored with timestamps always in UTC to avoid inconsistencies (regardless of the time zone selected in Meshlium).

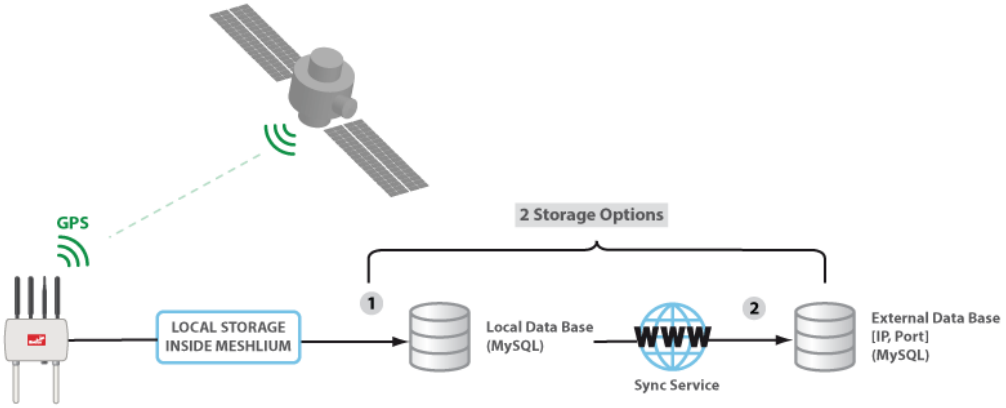


Figure : GPS storage options

15.6.2. Configuring GPS service

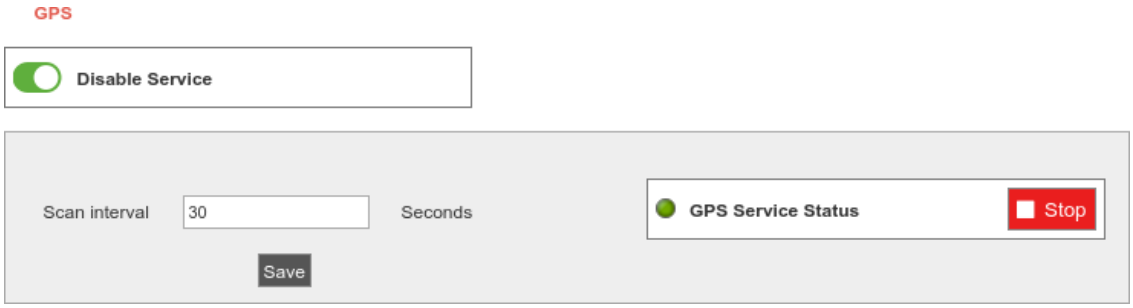


Figure : GPS plugin setup

The GPS service can be enabled or disabled. If the user disables the service the service will be stopped and will not be launched when Meshlium is powered on. In addition, setup will be blocked and cannot be changed. When the service is disabled no GPS information will be read or stored.

Figure : Enable control

Figure : Disable control

The user can set the time interval between data acquisition.

Figure : GPS data read interval

The service can be manually started and stopped.

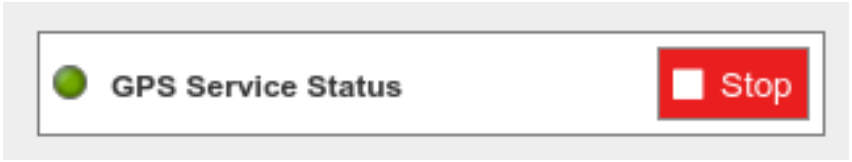


Figure : Service stop button

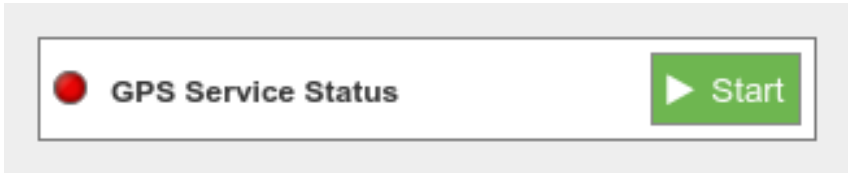


Figure : Service start button

15.6.3. Local database

Meshlium has a MySQL database up and running which is used to store locally the information captured. In the "Local Data Base" tab you can see the connection parameters.

A service in Meshlium will read periodically the GPS to gather location and will store it in the local database.

- **Database**: MeshliumDB
- **Table**: gpsData
- **IP**: localhost
- **Port**: 3306
- **User**: root
- **Password**: libelium2007

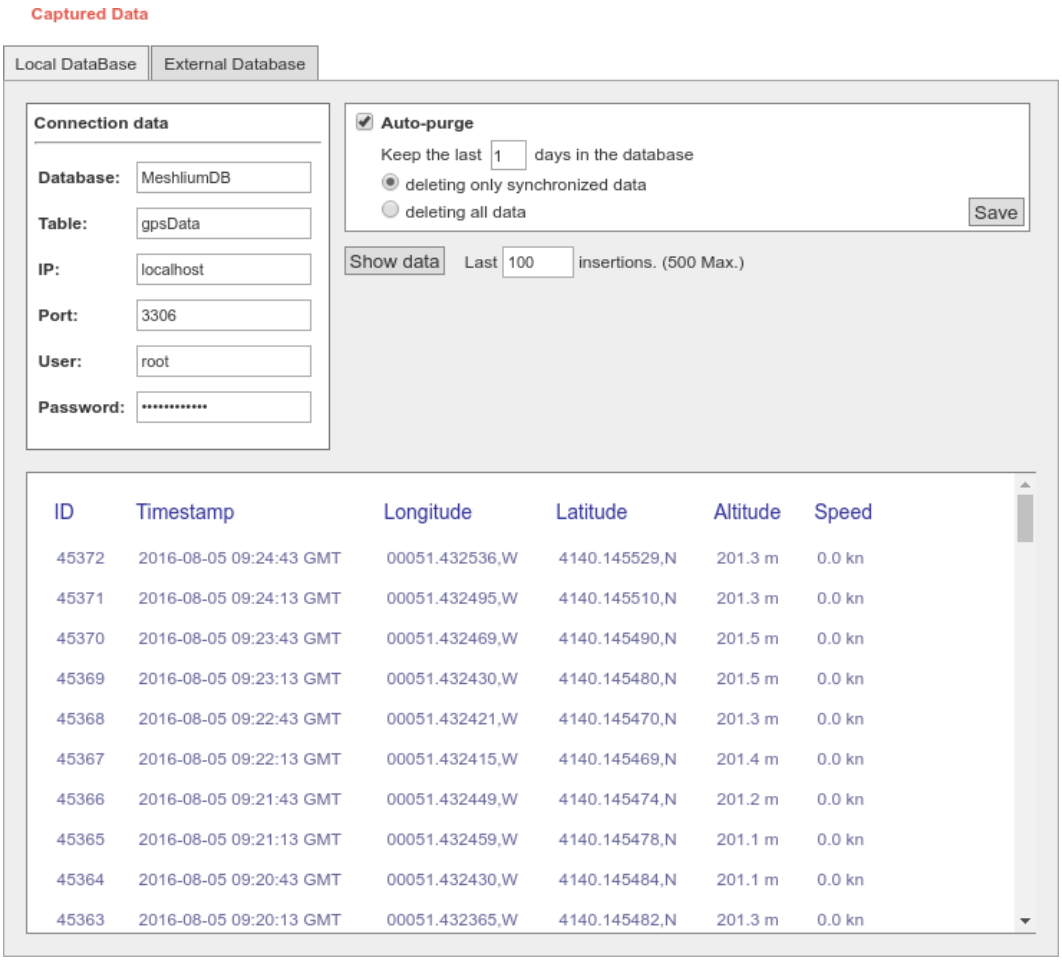


Figure : Local database for GPS data

At any time you can see the last "x" records stored, filtered by access points or clients. Just set how many and what kind of insertions you want to see and press the "Show data" button. The maximum number of data to display is 500.

The data from the database can be deleted pressing the button "Delete all data". Be careful, as this option deletes all the information of Bluetooth scans in the local database.

There is an option to program an automatic purge in the database every day, keeping the information in the database the days you specify. Furthermore, if you intend to configure the external database, you can choose if you want to delete only synchronized data or everything, taking care of the days established before.

15.6.4. External database

Meshlium can synchronize all the WiFi devices information stored in the local database to an external MySQL database managed by the user.

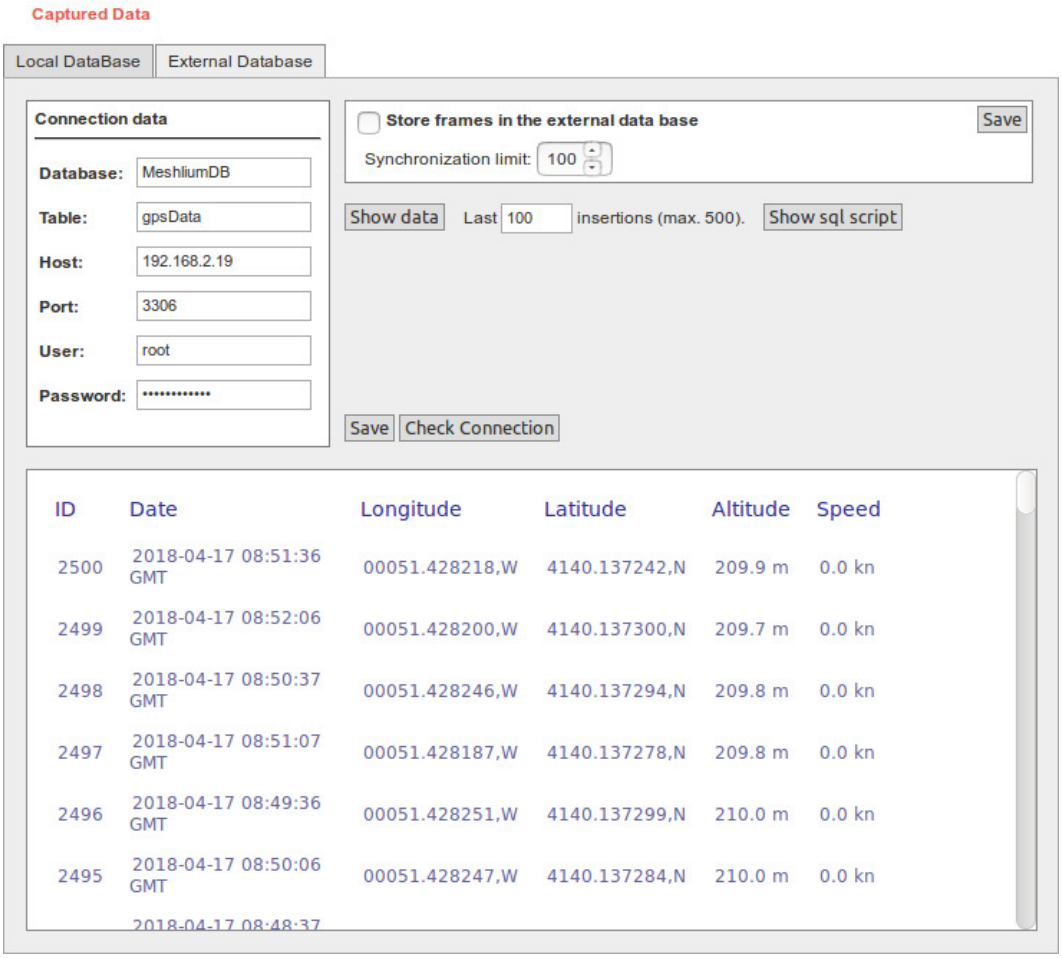


Figure : External database tab

In this tab the user can:

- Setup the parameters of the external database and check the connection.
- Enable or disable the synchronization.
- Select the number of fields sent per synchronization iteration.
- Show last data inserted in the external database (up to 500 data).
- Show the SQL script used to create the database and table needed for the synchronization.
- Mark all data in the local database as synchronized so it will not be sent to the external database.

The steps to setup the synchronization are:

- Before configuring anything, make sure you have a MySQL database working under your control. Make sure the database listen to connections in an external IP.
- Press the “Show sql script” button, copy the SQL code. You can modify user, password, database name and table, as long as you change the setup of the connection to match.

```
Just copy and paste:

CREATE database MeshliumDB;

CREATE TABLE IF NOT EXISTS `gpsData` (
  `ID_frame` int(11) NOT NULL auto_increment,
  `TimeStamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `date` DATETIME NOT NULL,
  `longitude` text collate utf8_unicode_ci NOT NULL,
  `latitude` text collate utf8_unicode_ci NOT NULL,
  `altitude` text collate utf8_unicode_ci NOT NULL,
  `satellites` int(11) NOT NULL,
  `speed` text collate utf8_unicode_ci NOT NULL,
  `MeshliumID` varchar(150) COLLATE utf8_unicode_ci NOT NULL default 'meshlium',
  PRIMARY KEY  (`ID_frame`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;


GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'passw';
```

Figure : SQL script

- Enter the connection settings and press “Save” button. You can check the connection now to ensure the settings are correct.
- Enable the service with the checkbox and save.

The synchronization service runs every 60 seconds and synchronizes up to 100 data every loop. The service synchronizes first newer data, as it is more relevant for decision making. This could make data in external database to be out of order. As every data has a timestamp, this should not be a problem for using the data in any external application.

15.7. Beep

When configuring several Meshlium at the same time in the laboratory, it can be difficult to distinguish between them (specially when the IP addresses are given by a external DHCP router). For this reason we have added a Beep button in the Tools section which will make the current Meshlium emit a short sound (“beep!”).

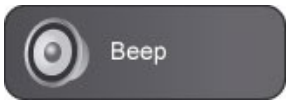


Figure : Beep plugin

16. Database management

16.1. Direct access

In order to access to the Meshlium Database from an external application you have to use the following parameters:

- **IP**:
  - **WiFi**: 10.10.10.1
  - **Ethernet**: Depending on your DHCP server (you can specify a static IP in the Interfaces section).
- **Database**: MeshliumDB.
- **Table**: Depending on the data to be extracted. Some options are: sensorParser, bluetoothData, gpsData, wifiScan. You can list the tables of the database.
- **Port**: 3306
- **User**: sslroot
- **Password**: The default password is "libelium2007". It is important that the user changes all default passwords immediately. For changing the password go to "Download HTTPS certificate" section.
- **Certificates**: Download the certificates. For more information go to "Download MySQL certificates" section.

Using the command line, you can connect using this command:

```
mysql -h 10.10.10.1 -u sslroot -plibelium2007 \
   --ssl-ca=ca-cert.pem \
   --ssl-cert=client-cert.pem \
   --ssl-key=client-key.pem
```

You can use any management MySQL application like *MySQL Workbench* or *SQLyog* to access the database in Meshlium and perform any maintenance operation.

16.2. PhpMyAdmin

Meshlium has a built-in phpMyAdmin instance that allows local database management. Go to:

Tools → phpMyAdmin

Click on "Open in a new window" to open the phpMyAdmin panel.

You can directly access phpMyAdmin panel in the URL:

https://[Meshlium_IP]/phpmyadmin

Where **[Meshlium_IP]** has to be replaced for the IP used in Meshlium. It can be WiFi AP, Ethernet or 4G IP.

To login, use the credential shown in the section "Direct access".

Figure : phpMyAdmin login page

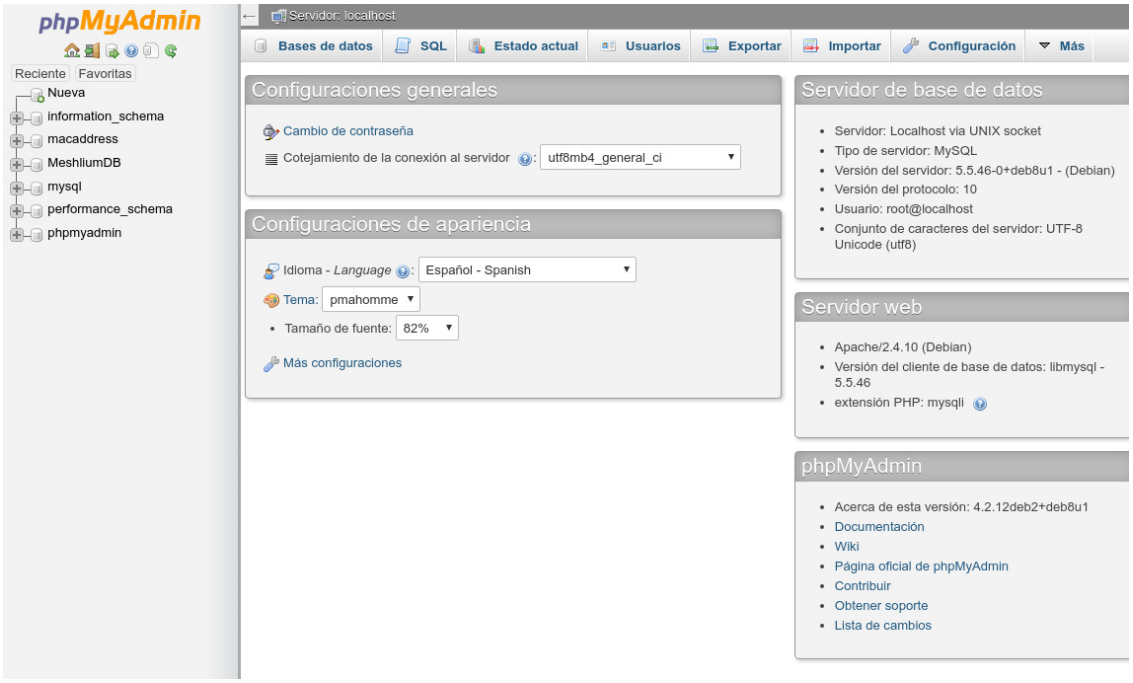


Figure : phpMyAdmin panel

17. System Information

17.1. Hostname

This plugin allow the user to change the hostname of the gateway. By default the name is “meshliumXXXX” where XXXX are the last four digits of Ethernet MAC address.

To change the hostname, enter the desired value and press “Save and Apply” button.

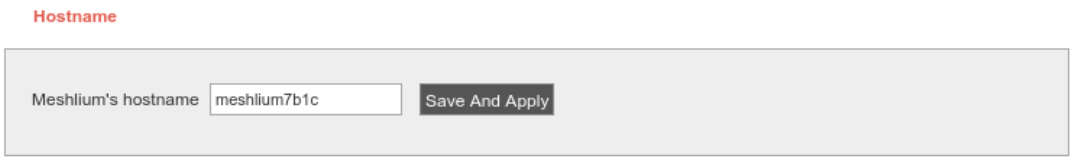


Figure : Hostname change form

17.2. User Manager

The “User Manager” configuration menu is located in:

System → Users Manager

In this section you can perform these actions:

- **Change password**: change the passwords needed to access to the different Meshlium services.
- **Download certificates**: download the certificates to communicate securely with Meshlium.

Figure : User Manager screen

Important:
Libelium strongly advises the user to change all default passwords immediately after receiving a new Meshlium unit.

17.2.1. Change passwords

You can change the password for these users:

- **admin** (Manager System): password for accesing to the Manager System interface.
- **log** (FTP): user for connecting to the FTP server and extract logs.
- **ota** (FTP): user for installing new binaries in remote Waspmotes.
- **sslroot** (MySQL): user for connecting securely to the MySQL database.

To change a password, press the "Change Password" button, introduce and confirm new password and press "Ok". You can abort the operation pressing "Cancel".

Figure : Password change form

17.2.2. Download certificates

Download HTTPS certificate

To download the certificate for sending secure frames to Meshlium, use the button "Download Certificate" of the Manager System "admin" user.

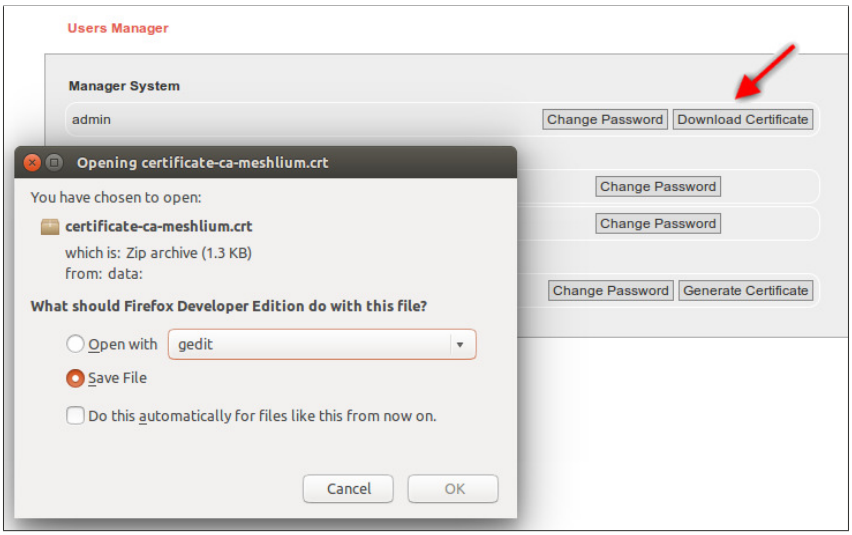


Figure : Download the certificate

For more information about how to use the certificate for secure communications in Libelium Ecosystem, check the pertinent guide:

- **Programming Cloud Service (PCS) guide**, section "How to download the Meshlium certificate for HTTPS connections":

  https://www.libelium.com/downloads/documentation/programming_cloud_service_guide.pdf.

- **Waspmote 4G Networking Guide**, section "Sending Waspmote frames to Meshlium via HTTPS":

  https://www.libelium.com/downloads/documentation/waspmote_4g_networking_guide.pdf.

- **Waspmote WiFi Networking Guide**, section "Send Waspmote frames to Meshlium via HTTP or HTTPS":

  http://www.libelium.com/downloads/documentation/wifi_networking_guide.pdf.

Download MySQL certificates

To connect to the Meshlium MySQL database securely (SSL), you have to generate and download MySQL client certificates. Use the button "Download Certificates" of the MySQL "sslroot" user. Introduce the number of days that the certificate will be valid and press "Ok". Wait until the certificate is generated for downloading. A dialog to download the cerficate in a .zip file will be displayed. You can abort the generation of the certificate pressing "Cancel".

Figure : Download MySQL certificates form

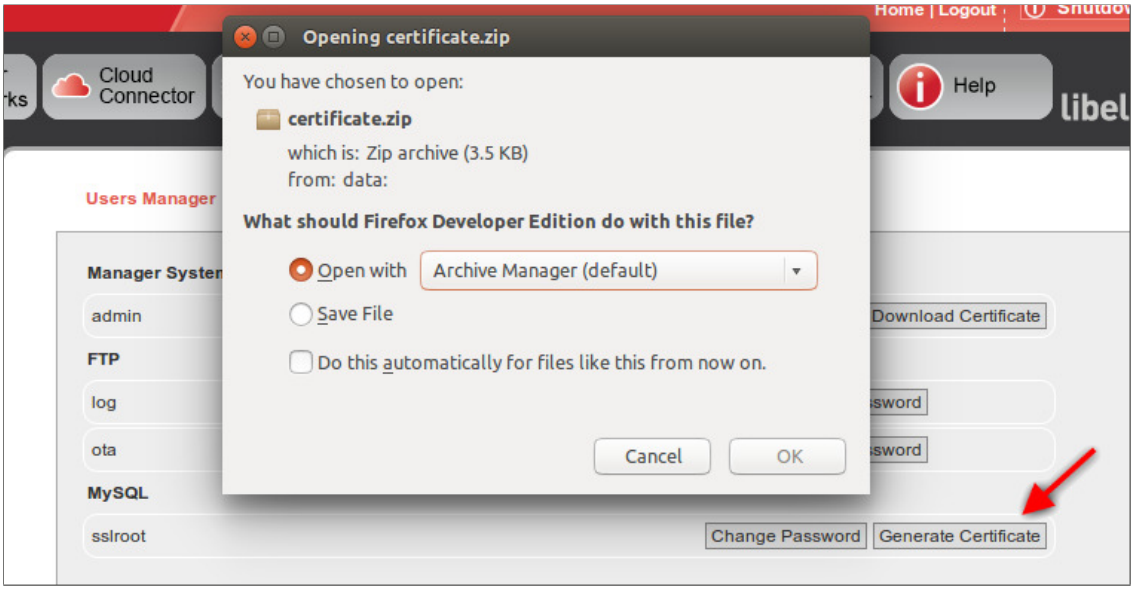


Figure : zip file with the compressed certificates (MySQL)

The zip contains 3 files:

- **ca-cert.pem**: file that contains the certificate of the Meshlium MySQL DB.
- **client-cert.pem**: file that contains X509 certificate generated for the client.
- **client-key.pem**: file that contains X509 key generated for the client.

For more information about how to use these files, go to "Direct access" section.

17.3. Security

This plugin offer the option to enable or disable Meshlium **HTTP** connections. By default, the system only allows *HTTPS* connections for security reasons. For enabling HTTP connections follow these steps:

1. Select “HTTP Service”.
2. Click on the tick.
3. Press “Save”.

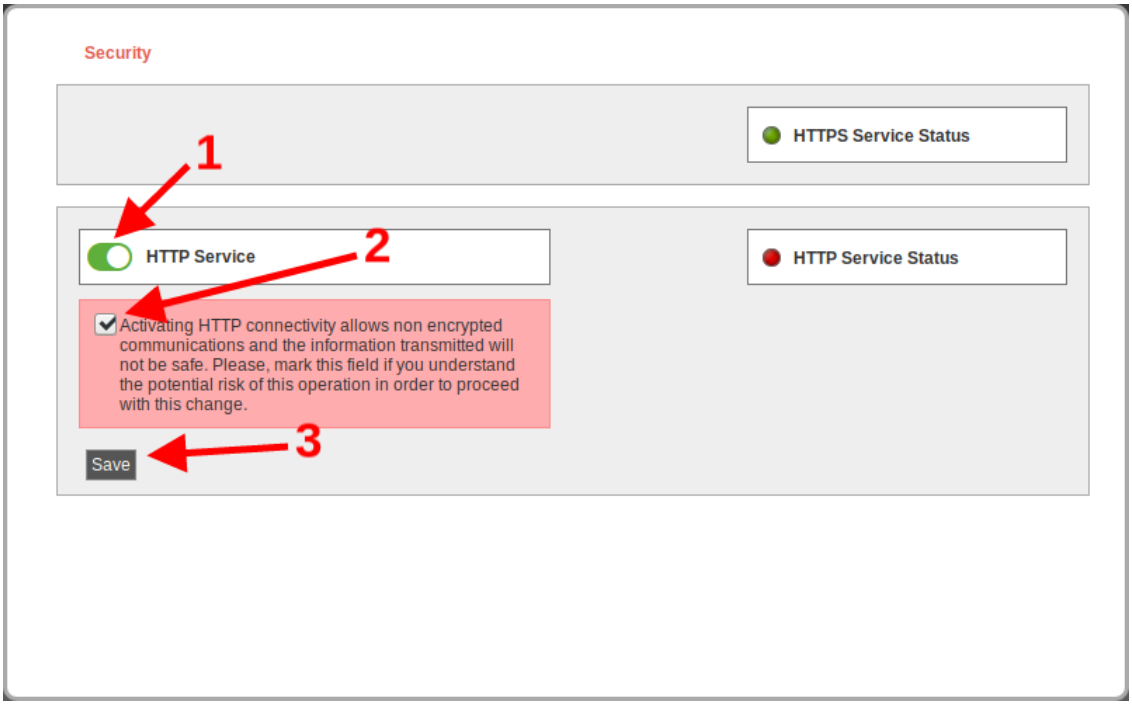


Figure : Enable HTTP service

Important:
Activating HTTP connectivity allows non-encrypted communications and the information transmitted will not be secure.

17.4. Activity Monitor

This plugin offers a graphical board to check important metrics of the system. The metrics displayed are: Uptime, Disk usage, Memory usage, Network usage and Proc usage.

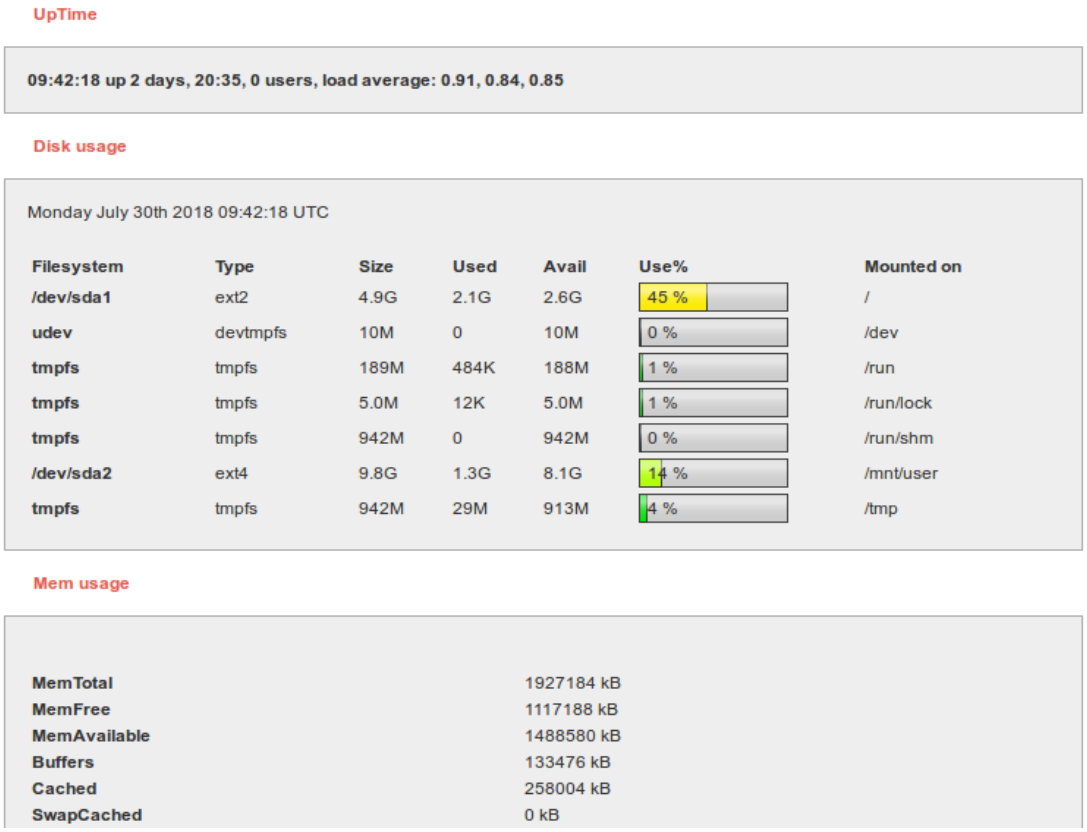


Figure : Activity Monitor plugin

17.5. Internal temperature sensor

In this plugin the user can see in real time the processor temperature in Celsius degree.

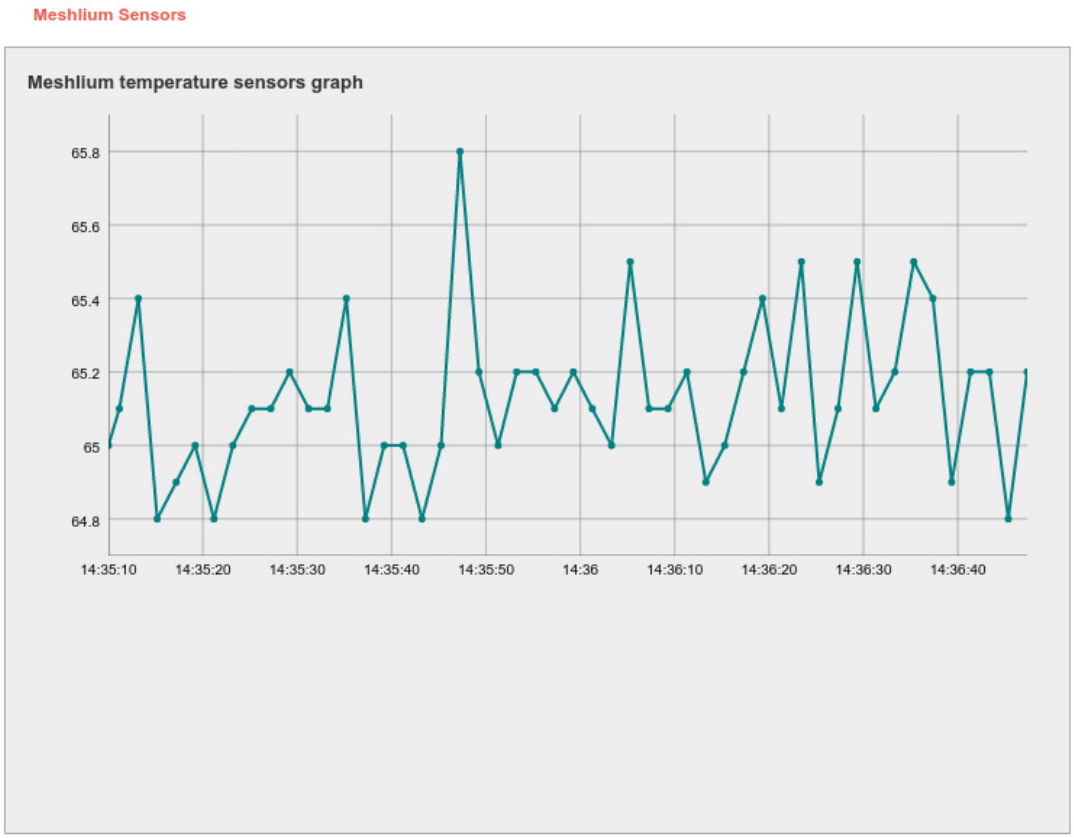


Figure : Temperature Sensors plugin

A temperature above 70°C is considered high and a temperature above 100°C could be dangerous for the device.

17.6. Time synchronization

In order to store correctly in the file system and in the local database the data captured from the RF modules and scanner is important to have the time correctly synchronized. Meshlium can be automatically synchronized using the NTP protocol. Activate the service to perform time synchronization: Meshlium will access an NTP server periodically, download the correct time and date and overwrite its internal clock. This feature has a small impact in terms of internet traffic via Ethernet or 4G connections.

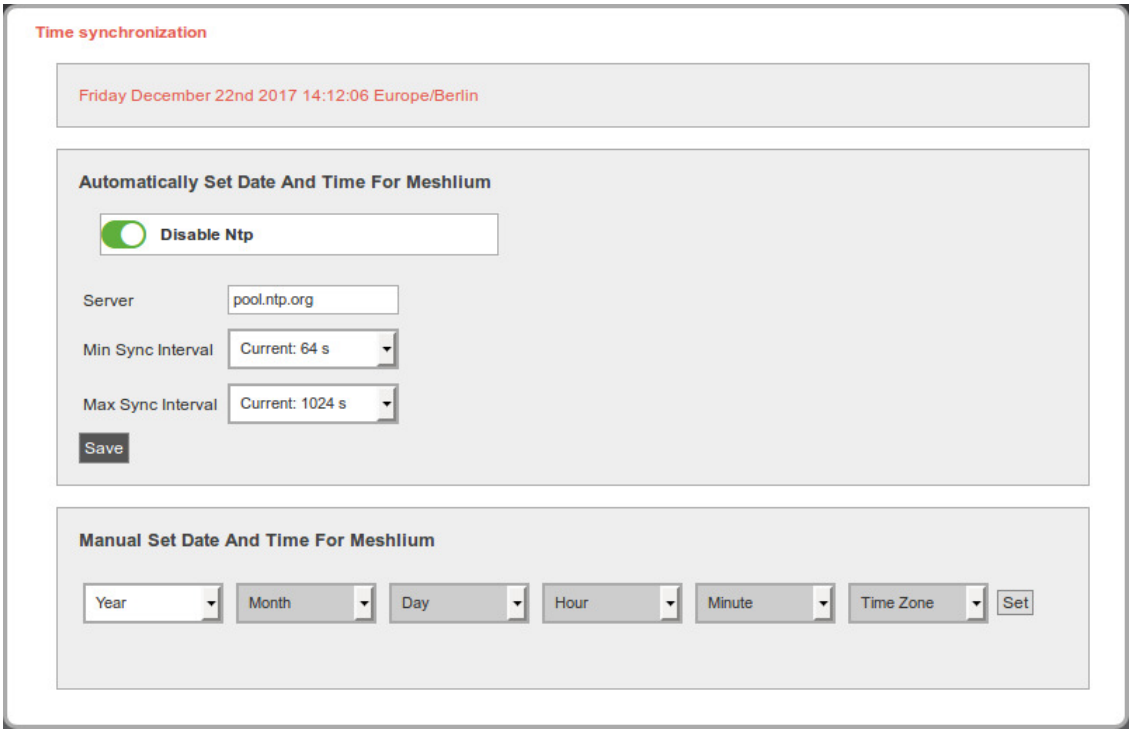


Figure : Time Synchronization plugin

The plugin also allows the user to set the time manually selecting the time from the selectors and setting the timezone. Press the “Set” button to apply the time selected.

Figure : Manually selecting time and timezone

18. Upgrading Meshlium

In the updates plugin, the user can install new Manager System versions. To get to the plugin go to:

Update Manager → Install Updates

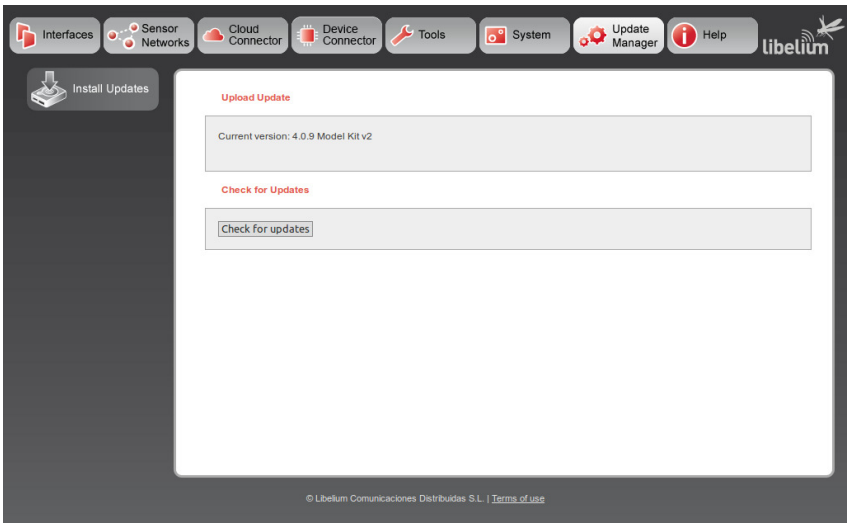


Figure : Install Updates plugin

In order to use this feature Meshlium needs to be connected to the Internet. Meshlium will **securely contact the Libelium servers** for performing all the checks and updates.

Press the "Check for updates" button to let Meshlium access the Libelium repository and search for new versions.

If there are new versions, the plugin will display the changelog of every version and will show an "Update" button. This button will download and install ALL the pending updates in one step, always leaving Meshlium with the latest version available.

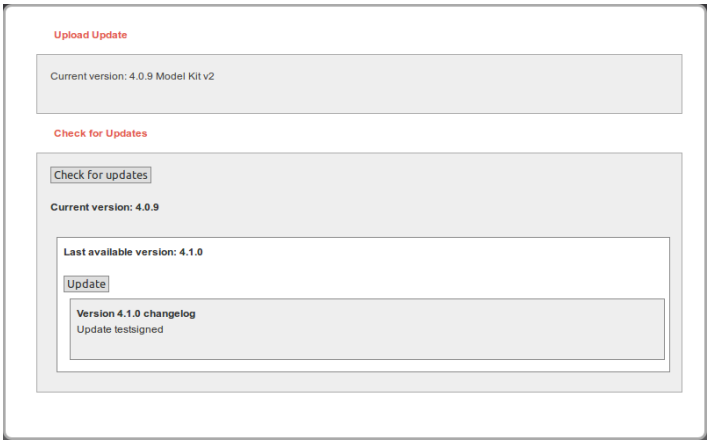


Figure : Updates found

Meshlium will reboot after installing the updates.

Important:
Make sure you do not disconnect Meshlium while the upgrading process happens. If the power supply is suddenly cut, Meshlium could be damaged.

19. Rescue System

The Rescue System allows the user to return Meshlium to factory defaults. After applying the rescue, the device will be formatted and the disk will be left as it was brand new.

The rescue process is recommended when:

- The operating system is corrupt or malfunctioning
- The device has been wrongly setup and it is unreachable
- The device needs to be recovered after an unexpected error

The rescue process is not covered by warranty as described in the section “Important: read me before using”. and must be considered an emergency process. This process is intended to help the user to recover the device of software issues without having to send Meshlium to the technical support service.

Important:
Executing the rescue process will delete all the user information stored in Meshlium including sensor information stored in the internal database.
The rescue process can potentially damage the device and left it unusable.
The rescue process may not work if the file system is severely damaged.

19.1. Rescue steps

You will need a USB pen drive of at least 8 GB formated in FAT32.

The first thing to do is download from the website of Libelium the image file “meshliumrescue.iso” needed to restore Meshlium to factory defaults.

***Note**: You need to contact first our Technical Service Department in order to get the user and password and URL to download the image.*

Go to https://www.libelium.com/contact/#RMA.

Once the file has downloaded, you must burn it in a USB pen drive. To perform this operation we recommend the use of **unetbootin**.

Unetbootin is a freeware and multiplatform application that allows to create bootable USB pendrives. It can be installed easily in Windows, Linux and MacOS machines.

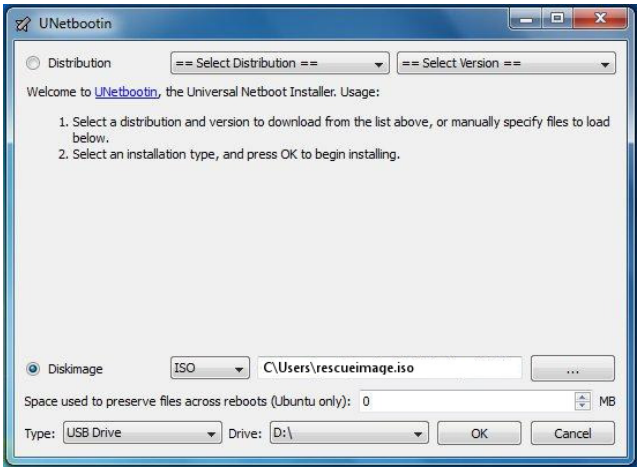


Figure : unetbootin interface

To create the rescue you have to:

- Start **unetbootin** and select the option “Diskimage”.
- Select the iso of the rescue image in the file selector.
- Select the USB unit where the image will be written.
- Click “OK” button to start the process.

In some minutes the image will be written and the USB will be ready to use.

To apply the rescue in Meshlium:

- Power off the device.
- Plug the USB pendrive in the micro-USB connector trough a USB-OTG cable.

Figure : USB pendrive connected to Meshlium

- Power on the device.
- When the rescue starts, Meshlium will emit a beep. The process can take a few minutes, so be patient.
- When the rescue process finishes, Meshlium will emit several beeps to notify.
- Power off Meshlium and unplug the USB pendrive.
- Your Meshlium should be now in a default state. It can be powered on now. The user can access Manager System with default credentials and start the setup fresh and use Meshlium normally.

20. Manager System changelog

Version 4.1.2

- Added new Libelium Cloud Hive service cloud connector.
- Updated Amazon cloud connector.
- Updated ThingWorx cloud connector.

Version 4.1.1

- Updated sensor list with the new Smart Agriculture Xtreme sensors.
- Updated Microsoft Azure Hub cloud connector.
- Updated ThingPlus cloud connector.
- Updated ThingSpeak cloud connector: fix problems with identical timestamps.
- Updated Axis device connector.

Version 4.1.0

- Added Apache HTTP/HTTPS service tool.
- Added new cloud connector: ThingSpeak.
- Updated cloud connectors: ThingWorx, Telefónica, Symphoni, MQTT.
- Fixed 4G reconnection after a restart.
- Updated sensor list with the new Smart Agriculture Xtreme sensors.

Version 4.0.9

- Enhanced Security:
  - HTTPS protocol for all Manager System connections.
  - MySQL over SSL/TLS new "sslroot" user.
  - GnuPG to encrypt and sign update files.
- Added new cloud connector: Alibaba Cloud.
- Updated cloud connectors: Amazon Web Services IoT, Cumulocity, Plasmacomp (C2M), Symphoni (Sparkster), Telit.
- Stored field MeshliumID with serialNumber in database tables.
- Updated sensor list.
- Accepted non alfanumeric characters in fields Host, User and Password of external databases.

Version 4.0.8

- Patched escape args exploit.

Version 4.0.7

- Accepted IP address in External Database of sensor capturer.
- Added option for accepting only cyphered frames in HTTP sensor parser.
- Added new cloud connector Aveva (Wonderware).
- Added new cloud connector Scriptr.io.
- Corrected GPS service activation/deactivation.

- Added feature to detect 5 GHz devices (Meshlium Scanner, WiFi mode).
- Fixed synchronization bit in WiFi/Bluetooth external database.
- New Premium classification clouds 2018: Arrow, SensorInsight, Telit, Ericsson, ThingWorx.
- New Advanced classification clouds 2018: CymbIoT, Ubicamovil, Microsoft, Infiswift.
- Set timezone to UTC in WiFi/Bluetooth Scanner.
- Stored/Showed minor devices Class of Device (CoD) in Bluetooth Scanner.
- Updated cloud connectors: Azure IoT Hub, NEC Connexive, ElementBlue SensorInsight, Ericsson DDM and Symphoni (now Sparkster).

Version 4.0.6

- Updated Azure IoT cloud connector: correct status detection.
- Several bug fixes and improvements.

Version 4.0.5

- Supported the use of 256-bit AES cryptographic keys in the Sensor Parser.
- Increased sync fields in tables wifiScan and gpsData.
- Patched Australian 4G connection failure.
- Added NTP service.
- Corrected “Show me know” function.
- Corrected last_update field value in Waspmote list.
- Solved bug ping and traceroute.
- Added cloud connectors: Biz4intellia, Connexive, Fujitsu, RoboMQ, Sparkcompass, Ubicamovil.
- Updated Amazon cloud connector: add a QoS parameter, correct bugs.
- Updated Arrow cloud connector: improve performance.
- Updated Azure IoT cloud connector: add synchronization interval, correct bugs.
- Updated Cumulocity cloud connector: add interval fields and timestamps.
- Updated ESRI cloud connector: use Meshlium GPS positions.
- Updated Symphoni cloud connector: improve performance.

Version 4.0.4

- Added new menu “Devices”.
- Added Axis device connector.
- Added new menu “Activity Monitor” for replacing “Disk Usage”.
- Added option to change log level in Sensor Parser.
- Allowed more characters for parameters in Bluetooth and WiFi Scanner.
- Added SSL support in the ThingWorx cloud connector.
- Now Azure cloud connector shows logs.
- Corrected bug on list of sensors: use wrong sensor list.
- Corrected bug on ThingPlus cloud connector: error at start time.
- Removed Devicify cloud connector.

Version 4.0.3

- Dynamic DNS configuration added with NoIP2 agent.
- Added new cloud connectors: Ericsson, infiswift, PlasmaComp, Redd System.

- Updated cloud connectors: Azure Hub, Bluemix.
- Augmented default log level in all the cloud connectors.
- Corrected 4G connection bugs.
- Corrected WiFiScan bug detection empty.

Version 4.0.2

- Added support for new protocol Device to Cloud.
- New cloud distribution: premium, advanced, basic.
- Added new cloud connectors: Arrow, Haibu, Qmic.
- Updated cloud connectors: Amazon, ElementBlue.
- Removed deprecated cloud connector Microsoft Azure Service Bus.
- Added a check process on the file system after system reboot.
- Show Wifi scan results on visualizations.
- Updated sensor list for v15.
- Several minor stability and interface updates incorporated.

Version 4.0.1

- Added new cloud connectors:
  - Cumulocity.
  - Kii.
  - Nexmachina.
  - RedTone.
  - SmartPlants.
  - TechEdge(SAP).

Version 4.0.0

- Network setup reviewed.
- Added “disable/enable” control to main services.
- Added compatibility with several radio modules.
- Added Auto-purge to sensor, Bluetooth scanner, WiFi scanner and GPS data.
- Cellular connection mechanism improved.
- Added Azure IoT Hub cloud connector.
- Update process improved. No reboot between updates needed..
- Added option to synchronize with.

21. Documentation changelog

From v8.0 to v8.1

- Added Libelium Cloud Hive service cloud connector.

From v7.9 to v8.0

- Updated information and figures with the new enclosure.

From v7.8 to v7.9

- Added instructions to make browsers trust the Meshlium Manager System self-signed certificate.
- Added the “Security” section.

From v7.7 to v7.8

- Added ThingSpeak cloud connector.

From v7.6 to v7.7

- Added Alibaba Cloud cloud connector.
- Added instructions to access by HTTPS.
- Added instructions to connect to Meshlium MySQL database over SSL.
- Changed the “User Manager” section.

From v7.5 to v7.6

- Added Aveva (Wonderware) cloud connector.
- Added scriptr.io cloud connector.
- Updated NEC Connexive cloud connector.
- Updated Ericsson cloud connector (also renamed to Ericsson DDM).
- Changed cloud connector name from “Symphony” to “Sparkster”.
- Changed cloud connector name from “RoboMQ Connector” to “RoboMQ”.
- Changed cloud connector name from “Connexive” to “NEC connexive”.
- Added references to the new Meshlium Scanner feature detecting devices on the 5 GHz band.
- Added field “Synchronization limit” in External Database of Wireless Scanner (WiFi and Bluetooth) and GPS.
- Added explanation the WiFi and Bluetooth data being stored in UTC time.
- Added description for the new “Accept only encrypted frames” option.
- Corrected synchronization values from 200 to 100 fields.
- Updated cloud connectors classification.

From v7.4 to v7.5

- Changed description of the new External SIM/USB Socket version, now nano-SIM compliant.

From v7.3 to v7.4

- Added new cloud connectors: Biz4Intellia, Connexive, Fujitsu, RoboMQ, Sparkcompass, Symphoni and Ubicamovil.

- Cloud connectors are reorganized in 3 groups: “Premium”, “Advanced” and “Basic”.
- Added NTP time synchronization description.
- Added Activity Monitor description.
- Added FTP access to logs explanation.
- Updated specs of the Scanner’s directional antennas.
- The upgrading process description was simplified.
- Added advice about the power supply via PoE connector.
- Removed WEP support.
- Removed iQmenic and Devicify cloud connectors.

From v7.2 to v7.3

- Added the new chapter “Device connectors”.
- Added new device connector for Axis.

From v7.1 to v7.2

- Added new cloud connectors: Arrow (Arrow Connect), Ericsson (Ericsson AppIoT), Haibu (HaibuSmart), infiswift, PlasmaComp (C2M), iQmenic-NexMachina (Labeeb), Redd System (Redd).
- Updated cloud connectors classification.
- Added notes for NoIP config in the “Interfaces” menu.

From v7.0 to v7.1

- Added new cloud connectors: BaseN, Cumulocity, Ensura, Kii, Orchestra, Microsoft Azure IoT Hub, Nexmachina, RedTone, SensorUP, SmartCityPlatform, SmartPlants, TechEdge SAP HANA, Telit.
- Updated cloud connectors: Amazon IoT, Microsoft Azure Event Hubs, MQTT, Sentilo.
- Updated antenna position in section “Contents of the box”.
- Added Bluetooth radio specifications in section “Specifications”.
- Removed SolvView cloud connector.

22. Certifications

22.1. General overview

Products	Europe	US	Canada	Australia	Brazil
Meshlium 4G 802.15.4 AP 868 EU	CE	-	-	-	-
Meshlium 4G 802.15.4 AP 900 US	-	FCC / PTCRB / AT&T	IC	-	-
Meshlium 4G 802.15.4 AP 900 AU	-	-	-	RCM	-
Meshlium 4G 802.15.4 AP 900 BR	-	-	-	-	ANATEL

22.2. CE (Europe)

Compliance with regulations:

- Electromagnetic Compatibility: EN 301 489-1 (1.9.2) / -17 (2.2.1) / -24 (1.5.1), EN 55022 (2010)
- Electrical Security: EN 60950-1 (2006) + A11 (2009) + A1 (2010) + A12 (2011) + Ac (2011) + A2 (2013) (except appendix Zx)

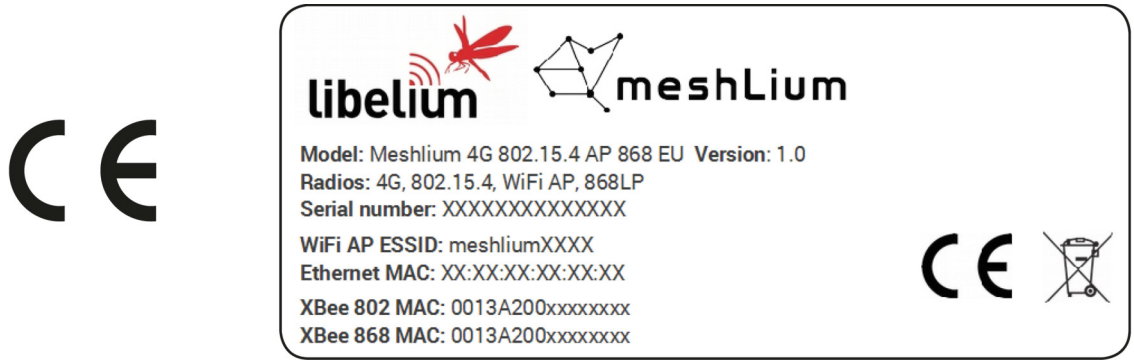


Figure : Back sticker for Meshlium 4G 802.15.4 AP 868 EU

22.3. FCC (US)

This document applies to the following Meshlium model:

Model	FCC ID
Meshlium 4G 802.15.4 AP 900 US	XKM-MESHLIUM-V1

Compliance with regulations:

- Electromagnetic Compatibility: FCC Part 15B ed.10.1.13
- Radiofrequency (radiated spurious): FCC Part 15.247 (2013) + CFR 47 Part 15.247 (2013) + FCC Part 22 (2014) + FCC Part 24 (2014) + FCC Part 27 (2014)

PTCRB compliance:

- Radiated Spurious Emissions: 3GPP TS 51.010-1 (s.12.2.x) + 3GPP TS 36.124 (s.8.2)
  - Bands: LTE FDD2, FDD4, FDD5, FDD17, 2G 900/1800
- OTA: CTIA Test Plan for Mobile Station OTA Performance v3.3.2 + AT&T document 13340, version 5.6 - Device Requirements.
  - Bands: LTE FDD2, FDD4, FDD5, FDD17, 2G 900/1800
  - Measures: TRP / TIS / ICS
- SIM electrical ETSI TS 102 230 (s. 5.x)

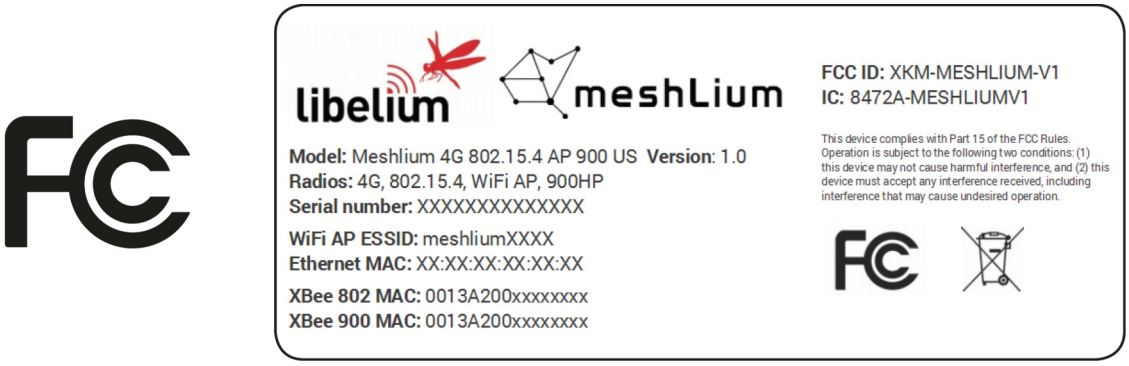


Figure : Back sticker for Meshlium 4G 802.15.4 AP 900 US

22.4. IC (Canada)

This document applies to the following Meshlium model:

Model	IC ID
Meshlium 4G 802.15.4 AP 900 US	8472A-MESHLIUMV1

IC

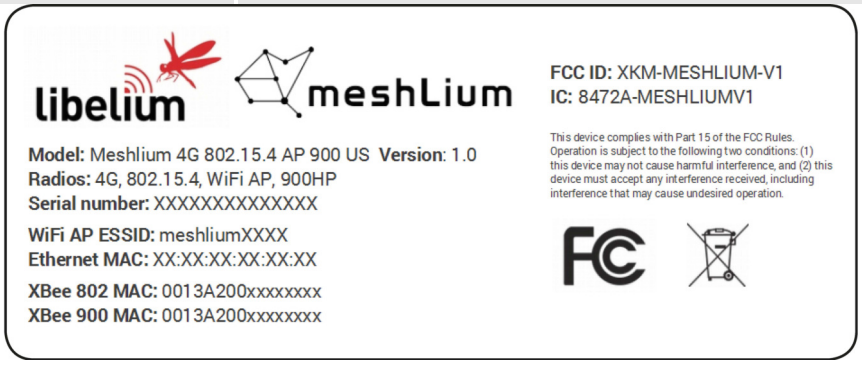


Figure : Back sticker for Meshlium 4G 802.15.4 AP 900 US

22.5. ANATEL (Brazil)

ANATEL

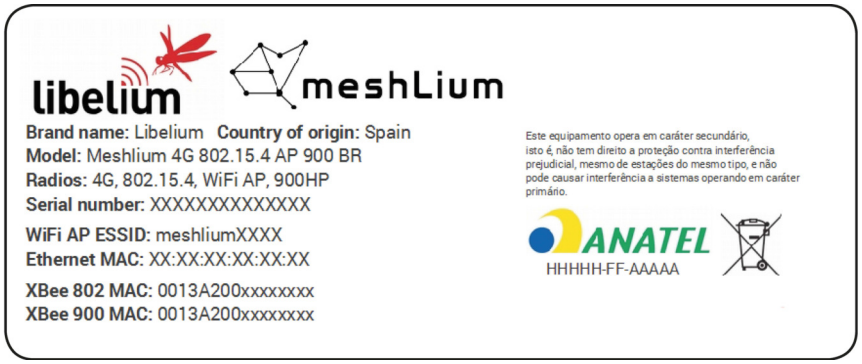


Figure : Back sticker for Meshlium 4G 802.15.4 AP 900 BR

22.6. RCM (Australia)

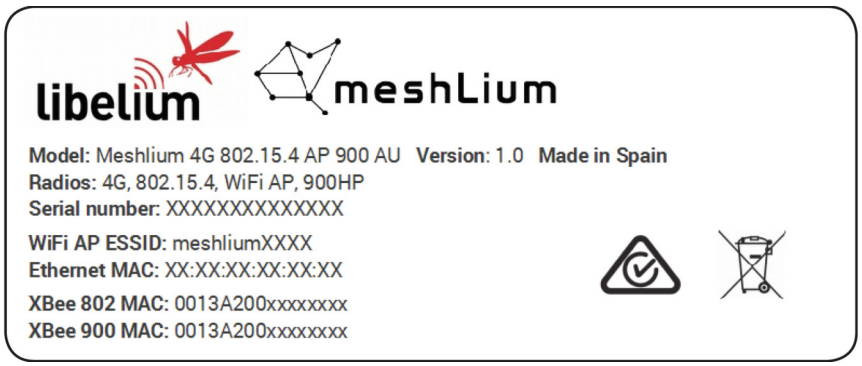


Figure : Back sticker for Meshlium 4G 802.15.4 AP 900 AU

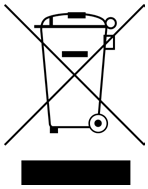
23. Maintenance

- Although Meshlium is a highly resistant product (IP67), please handle with care in order to enjoy a longer life of the product.
- Handle Meshlium with care, do not allow it to drop or move roughly.
- Avoid placing the devices in areas reaching very high temperatures that could damage the electronic components.
- The antennas screw on gently to the connector, do not force them while installing or you could damage the connectors.
- Do not use any type of paint on the device, it could affect the operation of connections and closing mechanisms.
- Power accessories must only be used indoors.
- Do not store Meshlium in places exposed to dirt and dust in order to avoid damage to electronic components.
- Never open the casing, the guarantee will not cover products that have been opened.
- For cleaning, use a damp cloth, do not use aggressive chemical products.

24. Disposal and recycling

When Meshlium reaches the end of its useful life it must be taken to a recycling point for electronic equipment.

- The equipment should be disposed of separately from solid urban waste, please dispose of correctly.
- Your distributor will advise you on the most appropriate and environmentally-friendly way of disposing of the product and its packing.



List of Tables