

PURPLE IOT

PROJECT DEMONSTRATION

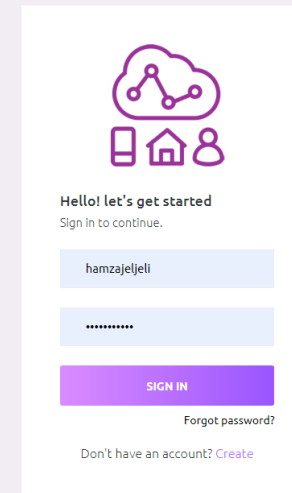
Jeljeli Hamza
jeljelihamza@gmail.com

INDEX

1. Logging In
2. Creating and managing projects
3. Adding and managing new devices
4. Sending data from the device to the broker
5. Viewing Data
6. Purple IoT Mobile Companion
7. Using the « HTTP over MQTT » feature

LOGGING IN SIGN IN

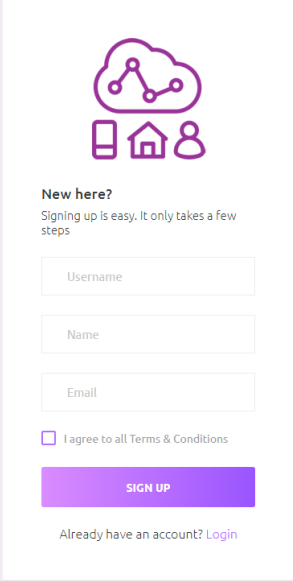
- ❖ To sign in, you just need to type your **username** and **password**.
- ❖ If you don't have an account, just click on the **Create** link.




The image shows a sign-in form centered on a light purple background. At the top is a purple logo consisting of a cloud with three nodes connected by lines, and below it are three icons: a smartphone, a house, and a person. Below the logo, the text reads "Hello! let's get started" and "Sign in to continue." There are two input fields: the first contains the text "hamzajelji" and the second contains a series of dots. Below these is a purple button with the text "SIGN IN". Under the button is a link that says "Forgot password?". At the bottom is a link that says "Don't have an account? Create".

LOGGING IN SIGN UP

❖ After filling the required fields, such as the **username** and **email** fields and clicking on the Sign up button. Your account's password will be sent to your email address.





New here?
Signing up is easy. It only takes a few steps

Username

Name

Email

☐ I agree to all Terms & Conditions

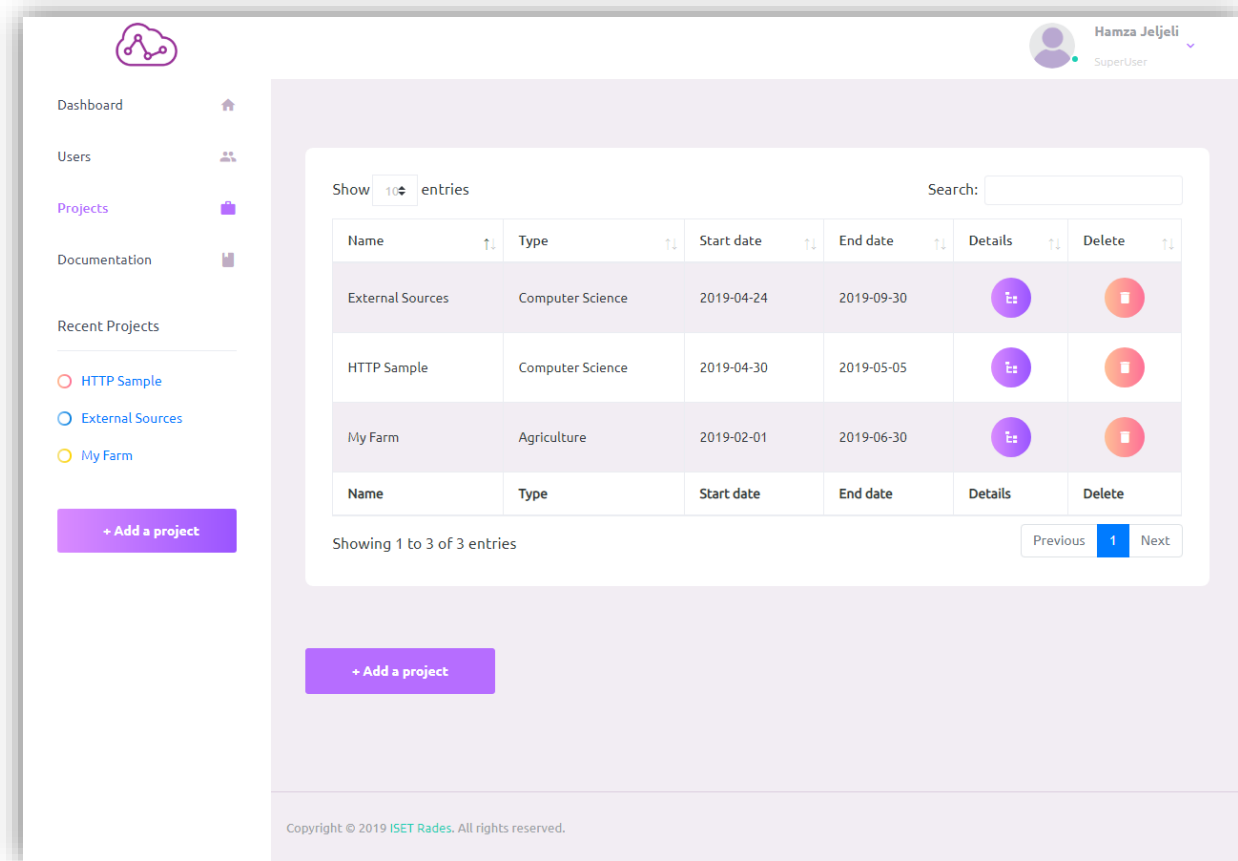
SIGN UP

Already have an account? [Login](#)

CREATING AND MANAGING PROJECTS

PROJECTS VIEW

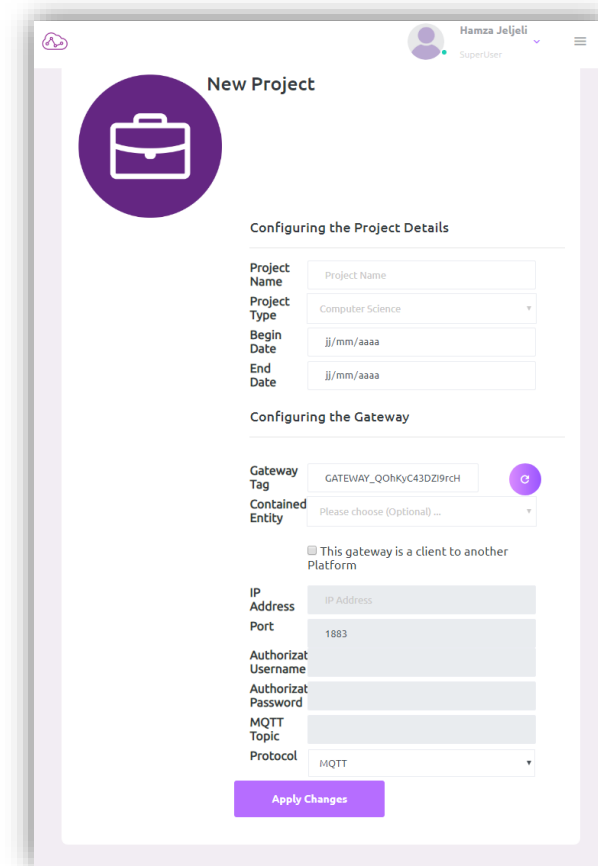
- ❖ After logging in and clicking on the **Projects** Menu. The list of your projects will appear.
- ❖ You can easily add new projects from the **Add a project** button.



CREATING AND MANAGING PROJECTS

NEW PROJECT

- ❖ To **create a project**, you just need to provide few details such as the **Project name, Type, Begin** and **end** dates.
- ❖ Configuring the project's **gateway** was made easy ! You just to specify the Gateway's **Tag** (By default, it is generated for you but still can choose your own name) and of course specify a **communication protocol**.



The screenshot shows a web application interface for creating a new project. At the top, there's a header with a user profile 'Hamza Jeljeli' and a 'SuperUser' role. The main title is 'New Project' next to a briefcase icon. The form is divided into two main sections: 'Configuring the Project Details' and 'Configuring the Gateway'. The first section includes fields for 'Project Name', 'Project Type' (set to 'Computer Science'), 'Begin Date' (format jj/mm/aaaa), and 'End Date' (format jj/mm/aaaa). The second section, 'Configuring the Gateway', includes a 'Gateway Tag' field with the value 'GATEWAY_QQhKyC43DZ9rcH', a 'Contained Entity' dropdown, a checkbox for 'This gateway is a client to another Platform', and fields for 'IP Address', 'Port' (1883), 'Authorization Username', 'Authorization Password', 'MQTT Topic', and 'Protocol' (MQTT). A purple 'Apply Changes' button is at the bottom.

CREATING AND MANAGING PROJECTS

NEW PROJECT

- ❖ If you want to use the **MQTT over HTTP** feature, you just need to switch the Protocol list box to **HTTP** and specify the topic.
- ❖ if you want that this gateway listens and retrieves data from **another MQTT Broker**, just tick the This gateway is a client to another Platform checkbox and fill the required parameters. Such as IP Address, Port, Topic, etc ...
- ❖ The **MQTTS broker** requires a certificate which can be downloaded from [here](#).

CREATING AND MANAGING PROJECTS

CHECKING A PROJECT'S DETAILS

The screenshot shows the 'My Farm' project details page. The left sidebar contains navigation links: Dashboard, Users, Projects, Documentation, and Recent Projects. The 'Recent Projects' section lists 'HTTP Sample', 'External Sources', and 'My Farm'. The main content area features a purple circular icon with a briefcase, the project name 'My Farm', and 'Edit' and 'Delete' buttons. Below this, the 'Project Details' section includes fields for Project Name (My Farm), Project Type (Agriculture), Begin Date (01/02/2019), and End Date (30/06/2019). The 'Gateway Details' section includes fields for Gateway Tag (ESP8266-CLIENT1056), Contained Entity (Please choose (Optional) ...), and a checkbox for 'This gateway is a client to another Platform'. At the bottom, there are fields for IP Address and Port.

Dashboard

Users

Projects

Documentation

Recent Projects

- HTTP Sample
- External Sources
- My Farm

+ Add a project

My Farm [Edit] [Delete]

Project Details

Project Name: My Farm

Project Type: Agriculture

Begin Date: 01/02/2019

End Date: 30/06/2019

Gateway Details

Gateway Tag: ESP8266-CLIENT1056

Contained Entity: Please choose (Optional) ...

☐ This gateway is a client to another Platform

IP Address: [Field]

Port: [Field]

The screenshot shows the 'Devices listing' page. The left sidebar is the same as the previous page. The main content area features a 'Contained Entity' dropdown menu (Please choose (Optional) ...), a checkbox for 'This gateway is a client to another Platform', and fields for IP Address, Port, Authorization Username, Authorization Password, MQTT Topic, and Protocol. Below this, the 'Devices listing' section includes a table with columns for Device ID, Device Type, and Actions. The table lists two sensors: 'SENSOR_LhDxlW248StPYdg' and 'SENSOR_xPFcA9QfQDb9lI'. Each sensor has a blue square icon and a red square icon in the Actions column.

Contained Entity: Please choose (Optional) ...

☐ This gateway is a client to another Platform

IP Address: [Field]

Port: [Field]

Authorization Username: [Field]

Authorization Password: [Field]

MQTT Topic: [Field]

Protocol: MQTT

Devices listing

Device ID	Device Type	Actions
SENSOR_LhDxlW248StPYdg	Sensor	[Blue Square] [Red Square]
SENSOR_xPFcA9QfQDb9lI	Sensor	[Blue Square] [Red Square]
Device ID	Device Type	Actions

CREATING AND MANAGING PROJECTS

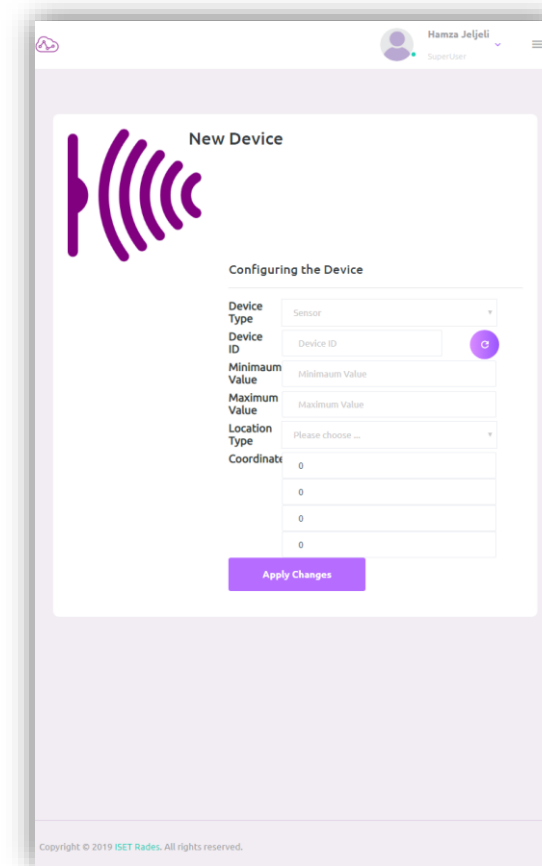
CHECKING A PROJECT'S DETAILS

- ❖ After choosing a project a project from the **Projects View** , you can easily read the details about the selected project. You can also edit the project's parameters or delete it.
- ❖ In the **Devices listing** part, in the button of the page. You will find a list of the devices associated to the gateway. To add a new **device** you just need to click on the **(+)** button !
- ❖ You can also View a **device details** or **delete** the device from the actions row in each device.
- ❖ If the Gateway listens to **another broker**, You will find instead of the Devices listing, a **chart** containing the **last 7 values** received from the distant broker.

ADDING AND MANAGING NEW DEVICES

ADDING A NEW DEVICE

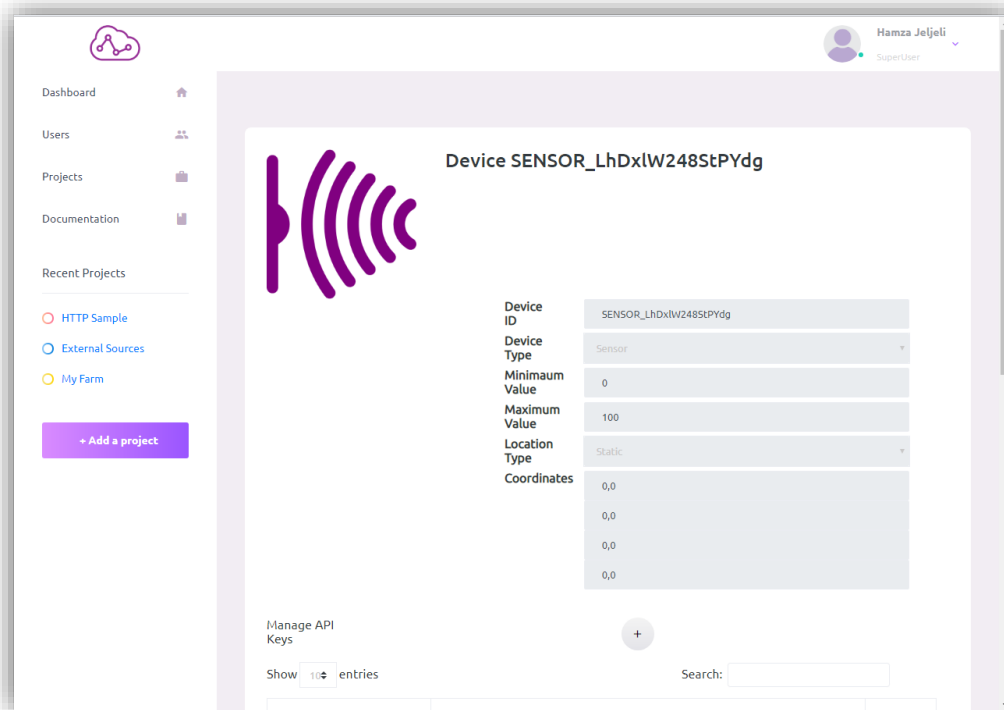
❖ To add a **new device**, you just need to associate to it a **Device ID**, set its type to a **sensor** or an **actuator**, set its **maximum** and **minimum** values and set the **location properties**.



The screenshot shows a web application interface for adding a new device. At the top, there's a header with a user profile icon and the name 'Hamza Jeljeli' with a dropdown arrow. Below the header, the main content area is titled 'New Device' and features a purple icon of a signal tower. The form is titled 'Configuring the Device' and contains several input fields: 'Device Type' (a dropdown menu with 'Sensor' selected), 'Device ID' (a text input field with a purple circular button containing a plus sign), 'Minimum Value' (a text input field), 'Maximum Value' (a text input field), 'Location Type' (a dropdown menu with 'Please choose ...' selected), and 'Coordinates' (four text input fields, each containing the number '0'). At the bottom of the form is a purple button labeled 'Apply Changes'. The footer of the application shows the copyright notice: 'Copyright © 2019 ISET Rades. All rights reserved.'

ADDING AND MANAGING NEW DEVICES

MANAGING A DEVICE



The screenshot shows the 'Manage API Keys' page for a device. The left sidebar contains navigation links: Dashboard, Users, Projects, Documentation, and Recent Projects. The main content area displays the device details for 'SENSOR_LhDxIW248StPYdg'. The details include the Device ID, Device Type (Sensor), Minimum Value (0), Maximum Value (100), Location Type (Static), and Coordinates (0,0). Below the details is a 'Manage API Keys' section with a search bar and a table showing the API keys. The user 'Hamza Jeljeli' is logged in as 'SuperUser'.

Device **SENSOR_LhDxIW248StPYdg**

Device ID: SENSOR_LhDxIW248StPYdg

Device Type: Sensor

Minimum Value: 0

Maximum Value: 100

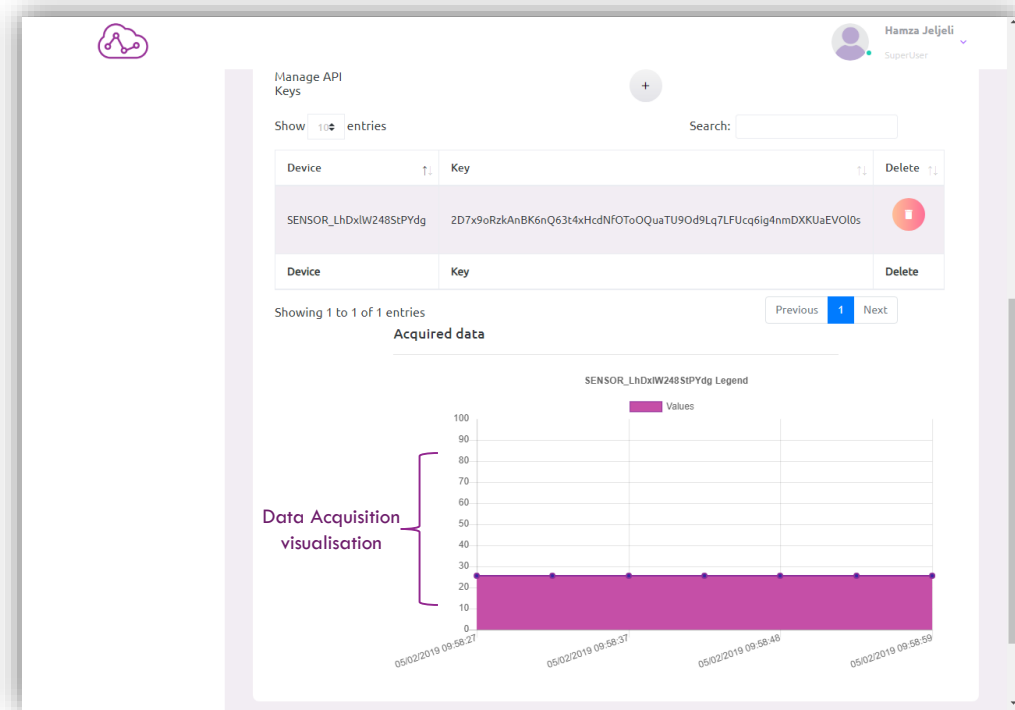
Location Type: Static

Coordinates: 0,0

Manage API Keys

Showing 1 to 1 of 1 entries

Device	Key	Delete
SENSOR_LhDxIW248StPYdg	2D7x9oRzkAnBK6nQ63t4xHcdNFOToOQuaTU9Od9Lq7LFUcq6g4nmDXKUaEVOI0s	



The screenshot shows the 'Acquired data' page for a device. The left sidebar contains navigation links: Dashboard, Users, Projects, Documentation, and Recent Projects. The main content area displays the device details for 'SENSOR_LhDxIW248StPYdg'. The details include the Device ID, Device Type (Sensor), Minimum Value (0), Maximum Value (100), Location Type (Static), and Coordinates (0,0). Below the details is a 'Manage API Keys' section with a search bar and a table showing the API keys. The user 'Hamza Jeljeli' is logged in as 'SuperUser'.

Device **SENSOR_LhDxIW248StPYdg**

Device ID: SENSOR_LhDxIW248StPYdg

Device Type: Sensor

Minimum Value: 0

Maximum Value: 100

Location Type: Static

Coordinates: 0,0

Manage API Keys

Showing 1 to 1 of 1 entries

Device	Key	Delete
SENSOR_LhDxIW248StPYdg	2D7x9oRzkAnBK6nQ63t4xHcdNFOToOQuaTU9Od9Lq7LFUcq6g4nmDXKUaEVOI0s	

Acquired data

SENSOR_LhDxIW248StPYdg Legend

Values

Data Acquisition visualisation

05/02/2019 09:58:27

05/02/2019 09:58:37

05/02/2019 09:58:48

05/02/2019 09:58:59

ADDING AND MANAGING NEW DEVICES

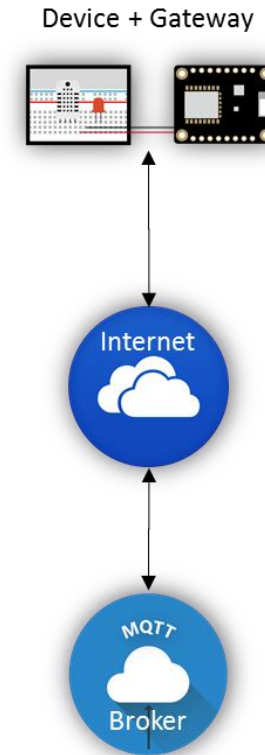
MANAGING A DEVICE

❖ From the **device management view**, you can check the **devices informations**, associate **API keys** to view received it's transmitted values and also view a **Chart** containing the variation of the **last 7 transmitted values** or to delete the device.

SENDING DATA FROM THE DEVICE TO THE BROKER

CONCEPTS

- ❖ The **device** sends through the **gateway** its data to the **MQTT Broker**.
- ❖ To send data to the broker, you must be **authenticated** to the broker using the **administration dashboard credentials**.



SENDING DATA FROM THE DEVICE TO THE BROKER

CONCEPTS

- ❖ The Data sent through the gateway must be in JSON format.
- ❖ **GID** : GatewayID (the broker can add this field if it wasn't found in the message).
- ❖ **SID** : The Device ID (required)
- ❖ **TD**: TimeDate (the broker can add this field if it wasn't found in the message).
- ❖ **VAL** : The value sent by the Device (required)

```
{  
  "GID": "ESP8266-CLIENT1056",  
  "VAL": 25.5,  
  "SID": "SENSOR_LhDx1W248StPYdg"  
}
```

Note : Keys must be written in UPPERCASE.

SENDING DATA FROM THE DEVICE TO THE BROKER

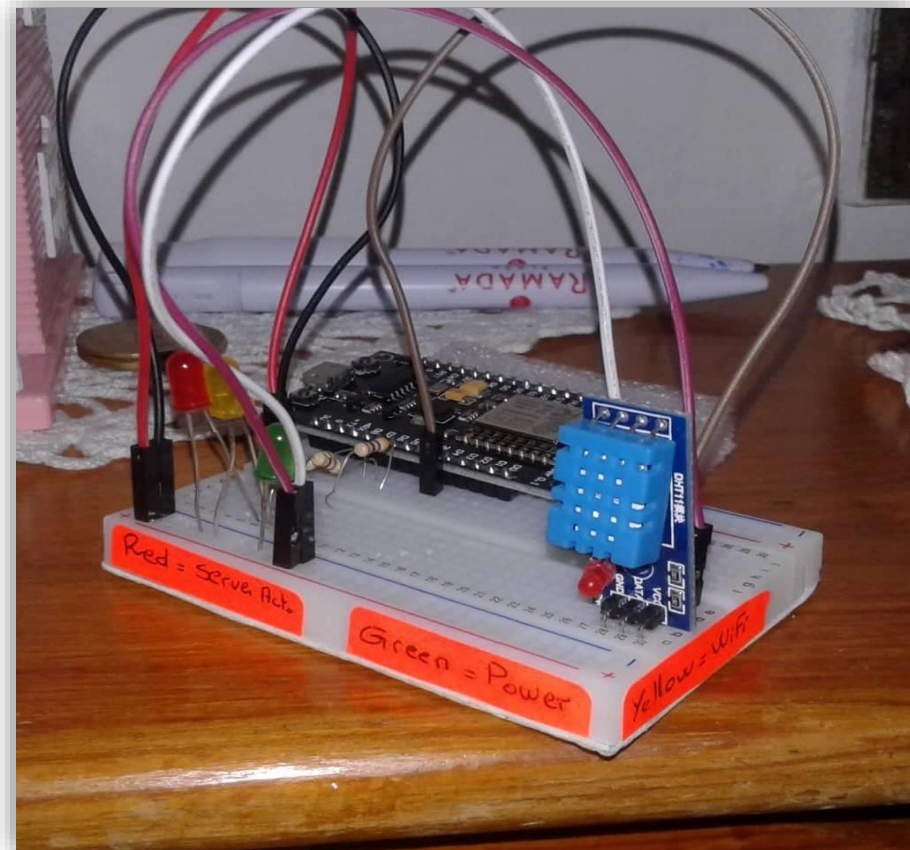
EXAMPLE 1 : USING A PYTHON MQTT CLIENT

- ❖ The purpose of this experiment was to send random values from the **Python based MQTT Client** to the **MQTT Broker** to check the compatibility with the existing MQTT Clients.
- ❖ The source code of this experiment can be found [here](#).

SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 2 : USING A NODEMCU ESP8266 V3 (ARDUINO BASED)

- ❖ The purpose of this experiment was to send **temperature** and **humidity** values using a **DHT11 sensor** to the MQTT Broker.
- ❖ The source code of this experiment can be found [here](#).



SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 3 : SIMULATING AN MQTT CLIENT

- ❖ For this experiment we used **MQTT.FX**, which is downloadable from [here](#).
- ❖ MQTT.fx is a **MQTT Client** written in Java based on [Eclipse Paho](#).
- ❖ MQTT.fx is published under [Apache License, Version 2.0](#).



SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 3 : SIMULATING AN MQTT CLIENT

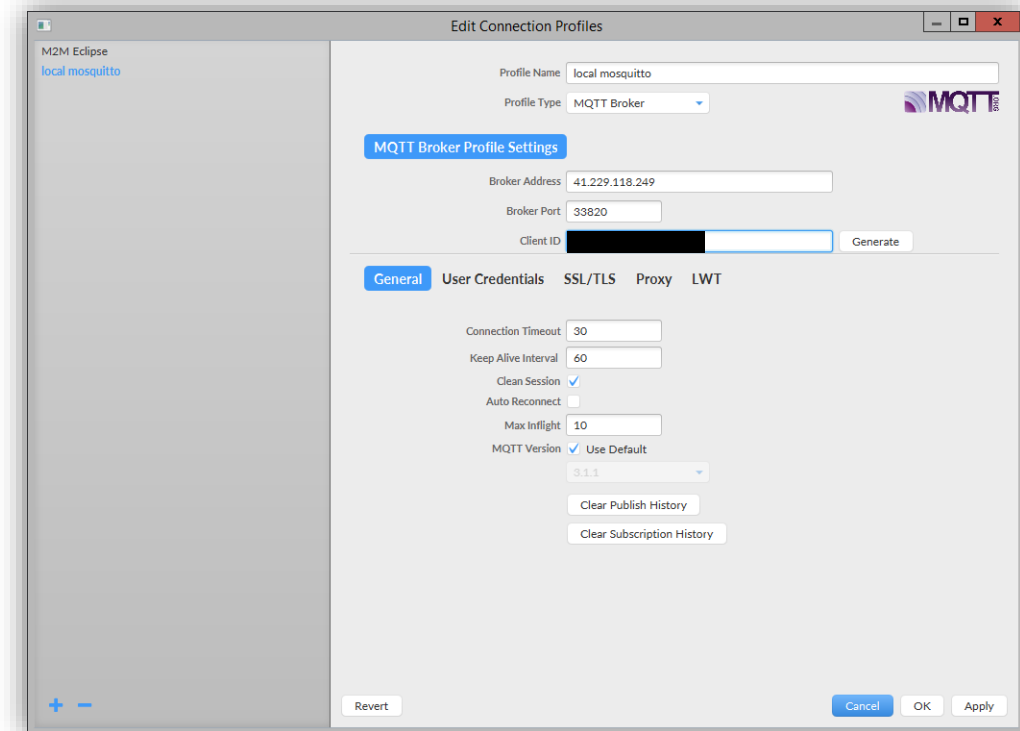
❖ After installing MQTT.FX, we need to configure it by clicking on the **blue gear** near the connect button.

❖ Connection properties are :

❖ IP Address : 41.229.118.249

❖ Port : 33820 (for MQTT) or 33830 (for MQTTS)

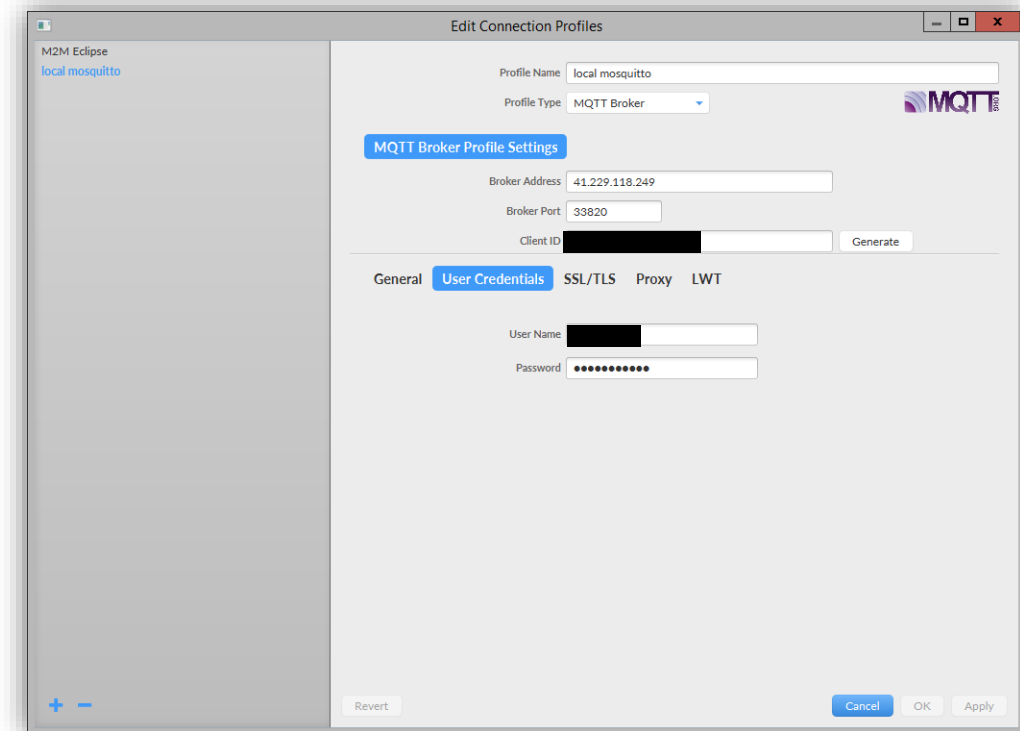
❖ Client ID : Your project's gateway tag



SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 3 : SIMULATING AN MQTT CLIENT

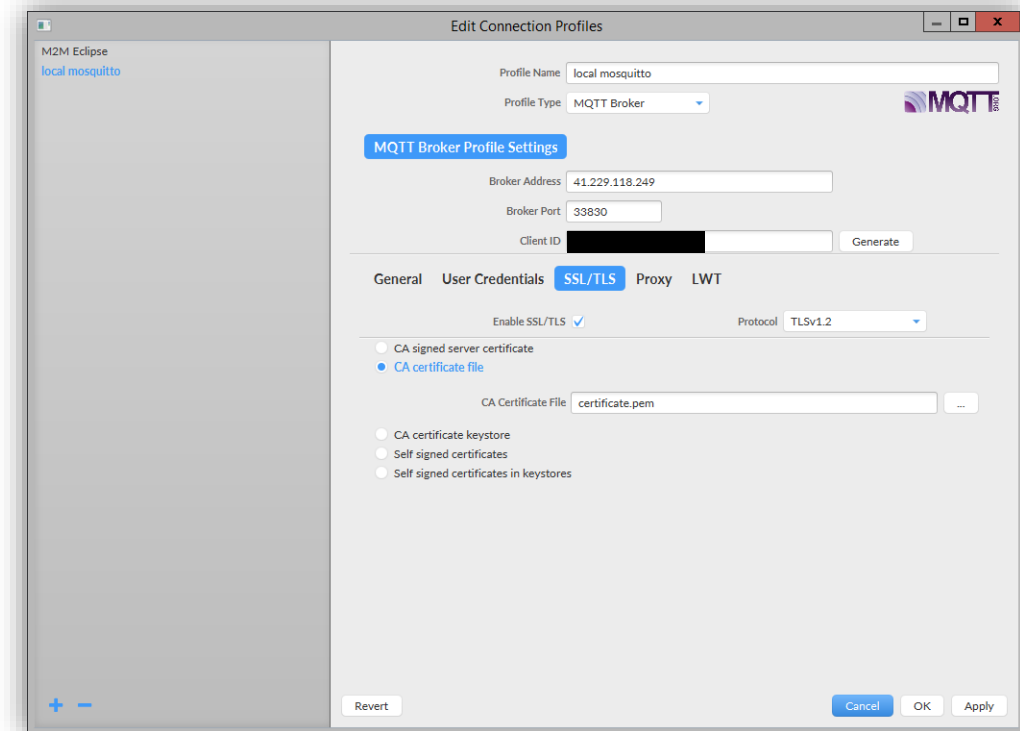
❖ You also need to type your **credentials** by selecting the User Credentials tab and writing your administration dashboard's credentials.



SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 3 : SIMULATING AN MQTT CLIENT

❖ If you are using MQTTS, you need to click on the SSL/TLS tab, check the Enable SSL/TLS checkbox and click on CA certificate file which is downloadable from [here](#).



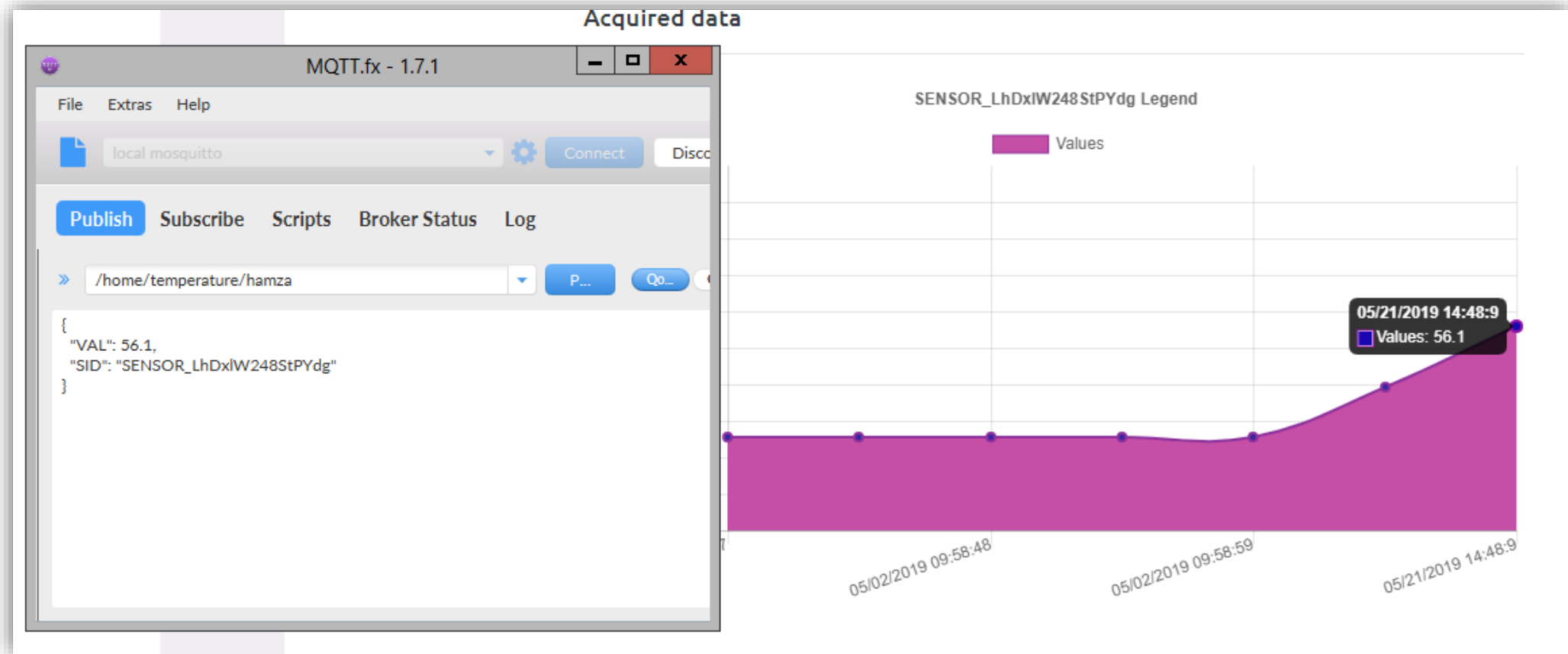
SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 3 : SIMULATING AN MQTT CLIENT

- ❖ After publishing the values to the broker, you will notice that the chart will be updated immediately. The broker will return **OK** if publishing was successfully done.
- ❖ The broker will also return other messages, such as :
 - ❖ NOT_OK : if there was a problem during the publishing.
 - ❖ PROJ_EXPIRED: if the project associated with the gateway has expired.
 - ❖ MISSING_VALUES: if SID or VAL keys are not found in the JSON message.
 - ❖ VALUES_MISMATCH: if the provided informations in the JSON message are incoherent.

SENDING DATA FROM THE DEVICE TO THE BROKER

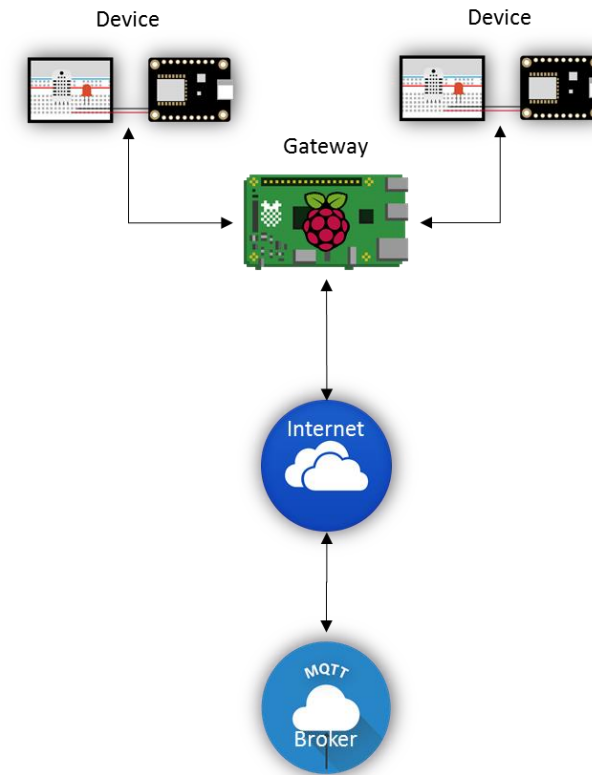
EXAMPLE 3 : SIMULATING AN MQTT CLIENT



SENDING DATA FROM THE DEVICE TO THE BROKER

EXAMPLE 4 : TRANSMITTING DATA FROM DEVICES BROKER THROUGH A RASPBERRY PI GATEWAY

❖ In this experiment, We installed into a Raspberry PI an MQTT Broker which could accept and handle requests from devices connected to him. This MQTT broker will also publish received messages to the Purple IoT Broker to enable the user to monitor and use the transmitted data.



VIEWING DATA

USING BASIC AUTHENTICATION

❖ Viewing the Data* acquired from the devices or gateways is guaranteed using Webservices :

Link	Method	Description
/WS/SensorsData/publish/	POST	Publish message as MQTT from HTTP.
/WS/SensorsData/get/{device}/	GET	Get the last value received from a device .
/WS/SensorsData/get/ {device}/all/	GET	Dumps all the values received from a device .
/WS/SensorsData/get/ {device}/{count}/	GET	Get a certain number of values received from a device .
/WS/GatewayData/get/{gatewayTag}/	GET	Get the last value received from a gateway** .
/WS/GatewayData/get/ {gatewayTag}/all/	GET	Dumps all the values received from a gateway** .
/WS/GatewayData/get/ {gatewayTag}/{count}	GET	Get a certain number of values received from a gateway** .

* : Requires authentication using the administration dashboard credentials.

** : Gateway that listen to an External Broker.

VIEWING DATA

USING BASIC AUTHENTICATION

❖ URL Examples :

- ❖ http://41.229.118.249:33810/WS/SensorsData/get/SENSOR_LhDxIW248StPYdg/
- ❖ http://41.229.118.249:33810/WS/SensorsData/get/SENSOR_LhDxIW248StPYdg/7/
- ❖ http://41.229.118.249:33810/WS/SensorsData/get/SENSOR_LhDxIW248StPYdg/all/

VIEWING DATA USING API KEYS

❖ Viewing the Data using generated API keys is also made possible through the Webservices :

Link	Method	Description
/WS/api/SensorsData/get/{device}/{apikey}/	GET	Get the last value received from a device .
/WS/api/SensorsData/get/{device}/all/{apikey}/	GET	Dumps all the values received from a device .
/WS/api/SensorsData/get/{device}/{count}/{apikey}/	GET	Get a certain number of values received from a device .
/WS/api/GatewayData/get/{gatewayTag}/{apikey}/	GET	Get the last value received from a gateway* .
/WS/api/GatewayData/get/{gatewayTag}/all/{apikey}/	GET	Dumps all the values received from a gateway* .
/WS/api/GatewayData/get/{gatewayTag}/{count}/{apikey}/	GET	Get a certain number of values received from a gateway* .

* : Gateway that listen to an External Broker.

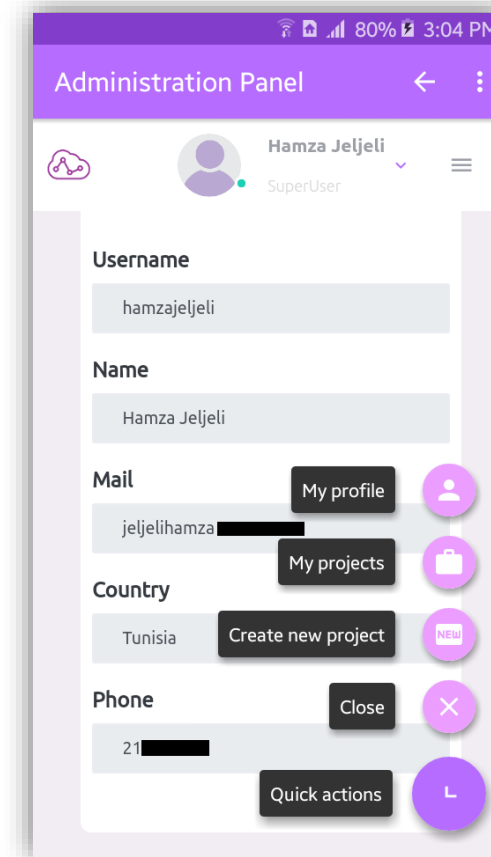
VIEWING DATA USING API KEYS

❖ URL Examples :

❖ <http://41.229.118.249:33810/WS/api/SensorsData/get/2D7x9oRzkAnBK6nQ63t4xHcdNfOToOQuaTU9Od9Lq7LFUcq6ig4nmDXKUaEVOI0s/>

PURPLE IOT MOBILE COMPANION

❖ All of the functionalities mentioned above may be also done through the Purple IoT Mobile Companion which is downloadable from [here](#).



USING THE « HTTP OVER MQTT » FEATURE

CONCEPTS

- ❖ In some cases, a device is not able to publish data to the broker due the lack of MQTT support for example. The “HTTP over MQTT” feature will enable any device to publish it’s data to the Purple IoT broker using a “POST” HTTP request containing it’s Device ID and Value to be published in JSON format.
- ❖ Depending on the response from the broker. An HTTP response will be sent back to the device.

USING THE « HTTP OVER MQTT » FEATURE

EXAMPLE : PUBLISHING USING « HTTP OVER MQTT »

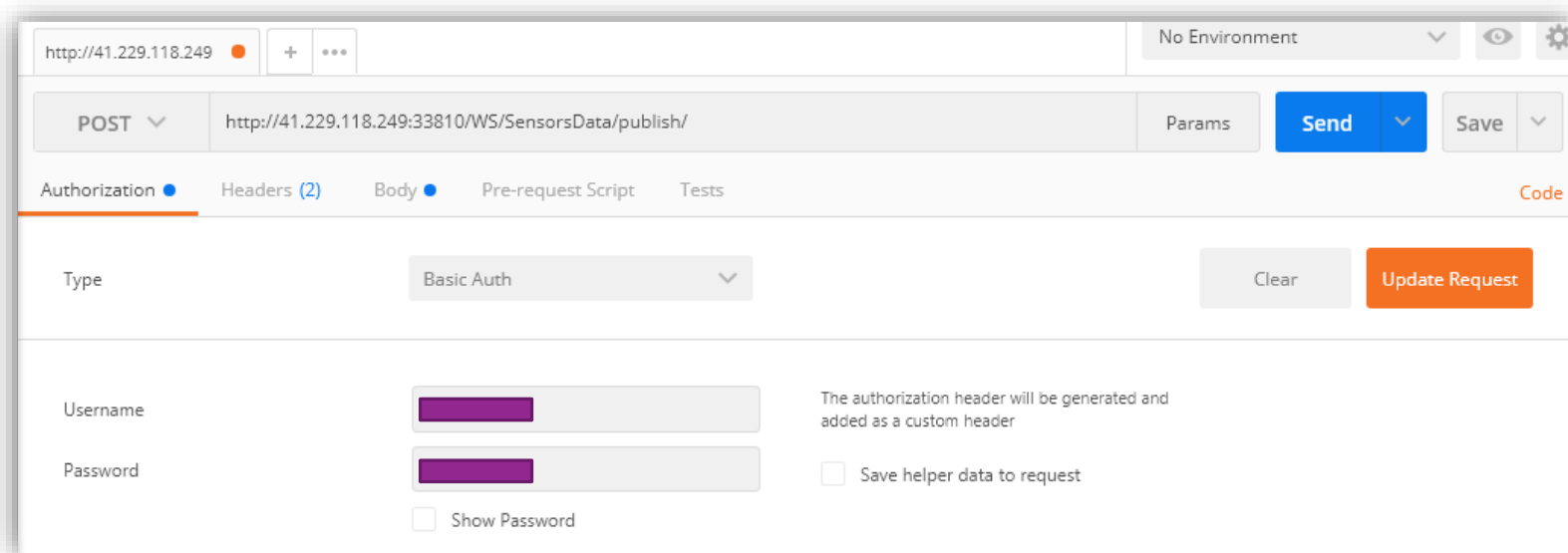
- ❖ For this experiment we used **Postman**, which is downloadable from [here](#).
- ❖ **Postman** is a tool used to send requests and receive responses through our REST API.



USING THE « HTTP OVER MQTT » FEATURE

EXAMPLE : PUBLISHING USING « HTTP OVER MQTT »

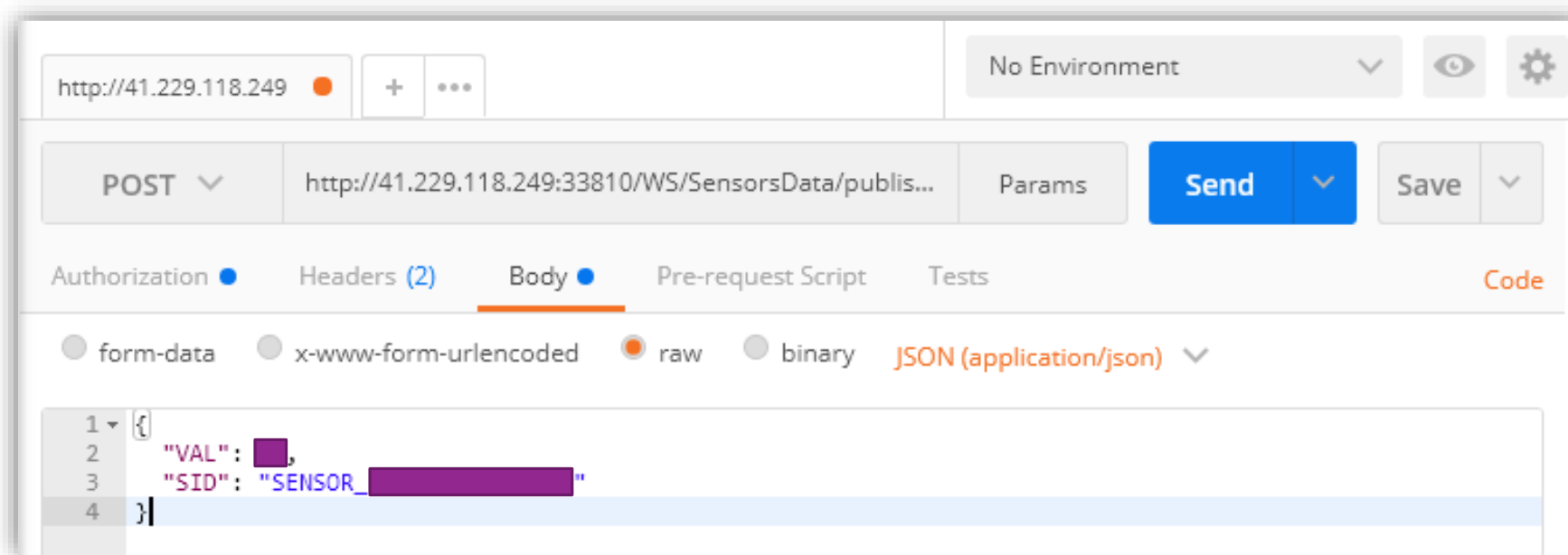
❖ After launching Postman and writing your credentials in the **Authorization** Tab, you must use the following URI to publish data to the MQTT broker form HTTP in POST method : **http://41.229.118.249:33810/WS/SensorsData/publish/**



USING THE « HTTP OVER MQTT » FEATURE

EXAMPLE : PUBLISHING USING « HTTP OVER MQTT »

❖ In the **Body** tab, we must send a JSON message and specify the SID and VAL values. For example :

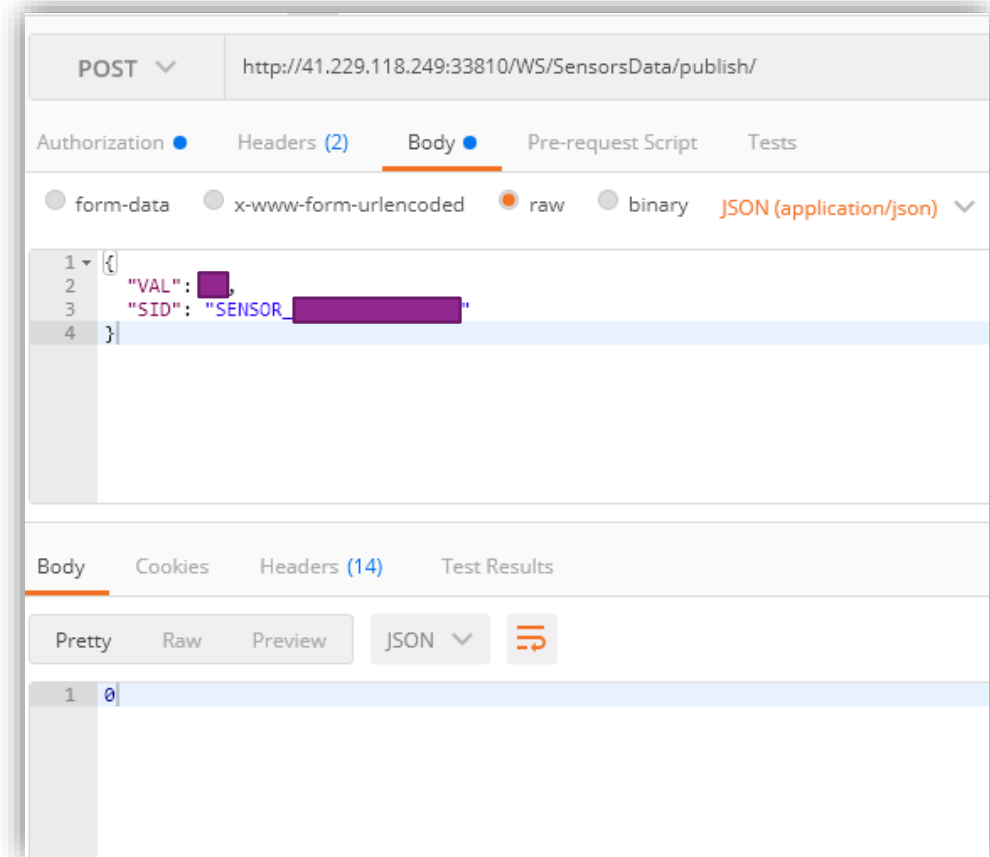


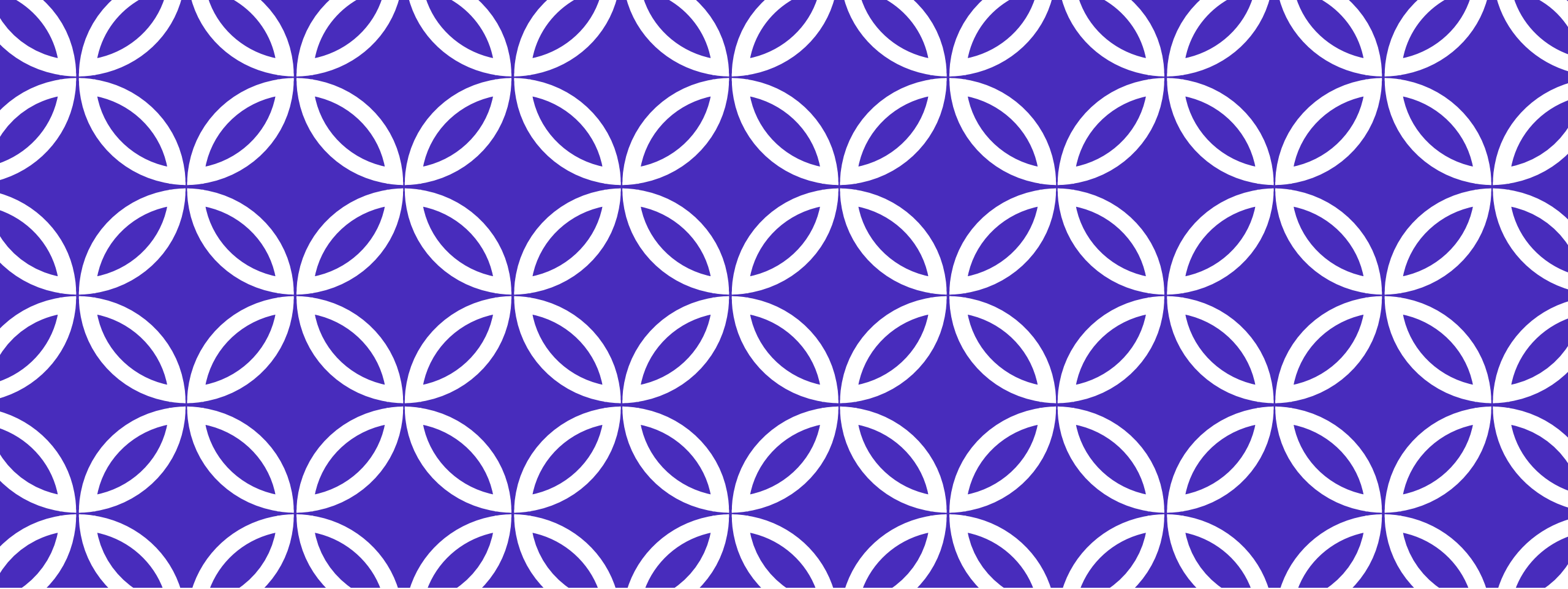
USING THE « HTTP OVER MQTT » FEATURE

EXAMPLE : PUBLISHING USING « HTTP OVER MQTT »

❖ After clicking on **Send** button, a response will be shown in the bottom. The response is an integer containing the publish response :

- ❖ SUCCESS = 0
- ❖ USERNAME_PASSWORD_MISMATCH = 1
- ❖ MISSING_REQUIRED_KEYS = 2
- ❖ SID_NOT_BELONGS_TO_USER = 3
- ❖ ERROR_HANDLING_THE_MESSAGE = 4
- ❖ MQTT_CONNECTION_FAILED = 5
- ❖ MESSAGE_PERSIST_ERROR = 6
- ❖ UNKNOWN_ERROR = 19





THANK YOU FOR YOUR ATTENTION !