

FORMAT EXPECTED

T-IOT-902

MSC-2025



**SENSOR SENSEI
IOT**



{ EPITECH }

Table of contents

1. Sensor.Community Data Format Requirements.....	3
API Endpoint.....	3
Required HTTP Headers.....	3
2. JSON Payload Format.....	3
a. Environmental Sensor (e.g., BMP280).....	3
b. Sound Level Sensor (e.g., SPH0645).....	4
3. LoRa Communication Format.....	4
4. Verification Methods.....	5
a. Serial Monitor.....	5
b. Postman or cURL.....	5
c. Sensor.Community Dashboard.....	5
d. LoRaWAN Gateway Console.....	5

1. Sensor Community Data Format Requirements

Sensor Community collects environmental data from distributed sensors. It requires data to be sent in specific JSON formats via HTTP POST requests to their API.

API Endpoint

<https://api.sensor.community/v1/push-sensor-data/>

Required HTTP Headers

Content-Type: application/json

X-Sensor: esp8266-<your-device-id>

- Replace <your-device-id> with the ESP32's unique ID (e.g., MAC address or custom name).

2. JSON Payload Format

Each sensor type must send data in a separate HTTP POST request.

a. Environmental Sensor (BMP280)

```
{  
  "software_version": "ESP32-Sensor-1.0",  
  "sensordatavalues": [  
    { "value_type": "temperature", "value": "22.4" },  
    { "value_type": "pressure", "value": "101200" }  
  ]  
}
```

The logo for Epitech, featuring the word "EPITECH" in a bold, sans-serif font, enclosed within large curly braces "{ }". The background of the entire page features a subtle, light gray geometric pattern of interconnected lines and dots, resembling a network or a molecular structure.

{ EPITECH }

- **temperature**: in Celsius (as string)
- **pressure**: in Pascals (as string)

b. Sound Level Sensor (SPH0645)

```
{
  "software_version": "ESP32-Sensor-1.0",
  "sensordatavalues": [
    { "value_type": "sound_level", "value": "67.5" }
  ]
}
```

- **sound_level**: in decibels A-weighted (dBA) (as string)

3. LoRa Communication Format

The message may need to be compact and encoded, for example in CSV format:

t:22.4,p:101200,s:67.5

Or as a compact JSON string:

```
{"t":"22.4","p":"101200","s":"67.5"}
```

This message can be parsed by the receiving gateway or device, then forwarded to Sensor.Community or other APIs if needed.

4. Verification Methods

To ensure the ESP32 sends correctly formatted data:

a. Serial Monitor

- Print JSON payloads to the serial monitor before sending to verify structure and values.

b. Postman or cURL

- Manually send test JSON payloads to the API to verify responses.

c. Sensor Community Dashboard

- Monitor real-time data updates and validate correct values are displayed.

d. LoRaWAN Gateway Console

- Use LoRa gateway's web console to inspect incoming payloads.