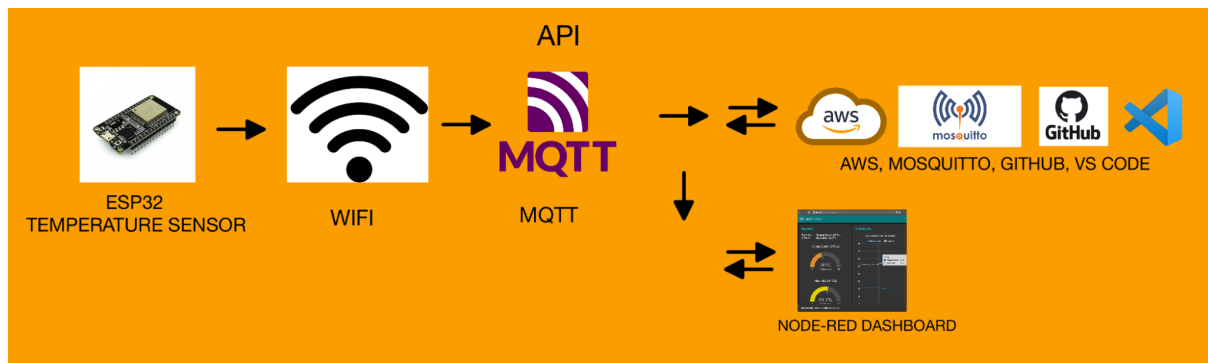


API Documentation



API (Application Programming Interface)

APIs let your product or service communicate with other products and services without having to know how they're implemented. APIs are sometimes thought of as contracts, with documentation that represents an agreement between parties: If party 1 sends a remote request structured a particular way, this is how party 2's software will respond.

MQTT Broker

It is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely fast publish/subscribe message transport. It is useful for connections to remote locations where a small code footprint is required and/or network bandwidth is at a premium.

To have a system and communication through MqTT we need an important element: The Broker. The broker is the device that will manage the reception and transmission of data to and from IOT devices. Devices that can have sensors in their configuration (which can be connected to electronic devices) and these sensors can send the data to a broker. For example, if we want a practical example, a device could be an ESP32, an arduino card, a raspberry. A broker could be a device like a raspberry pi, a pc that is installed on the MqTT broker or it could be a cloud in the cloud.

MQTT clients include publishers and subscribers, terms that refer to whether the client is publishing messages or subscribed to receive messages. These two functions can be implemented in the same MQTT client. When a device (or

client) wants to send data to a server (or broker) it is called a *publish*. When the operation is reversed, it is called a *subscribe*. Under the pub/sub model, multiple clients can connect to a broker and subscribe to topics in which they are interested.

An IBM writeup describes the pub/sub model: "Publishers send the messages, subscribers receive the messages they are interested in, and brokers pass the messages from the publishers to the subscribers. Publishers and subscribers are MQTT clients, which only communicate with an MQTT broker. MQTT clients can be any device or application (from microcontrollers like the Arduino to a full application server hosted in the Cloud) that runs an MQTT library."

An MQTT session is divided into four stages: connection, authentication, communication and termination. A client starts by creating a Transmission Control Protocol/Internet Protocol (TCP/IP) connection to the broker by using either a standard port or a custom port defined by the broker's operators. When creating the connection, it is important to recognize that the server might continue an old session if it is provided with a reused client identity. The standard ports are 1883 for non encrypted communication and 8883 for encrypted communication.

Helpful link for MQTT Broker and installation:

Explanation video by Prof. Luis Roa:

https://us02web.zoom.us/rec/share/cU5OEHyAtb-aoVWxIXAlrUHLj3exfoojMc5OZ0OG6f9n49NseALZ1K8GkhYq79T_.Kt4cevIo1FLCJDVC

API and MQTT Broker information:

<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>

<https://www.techtarget.com/iotagenda/definition/MQTT-MQ-Telemetry-Transport#:~:text=An%20MQTT%20broker%20acts%20as,be%20delivered%20to%20the%20subscriber.>

