Data

Clusters

Billing

Billing & Payment

Getting started with Arduino ESP8266

Prerequisites

Install Arduino IDE

To write code for an Arduino you first need to install the proprietary Arduino IDE. After installation, enable support for the ESP8266 chip, which provides the WiFi functionality. To do this, start the Arduino IDE and open the *Preferences* window.

Enter `https://arduino.esp8266.com/stable/package_esp8266com_index.json` into the `File>Preferences>Additional Boards Manager URLs` field of the Arduino IDE. You can add multiple URLs, by separating them with commas. Afterwards, open `Tools>Board>Boards Manager` and install the `esp8266` platform by searching for it. Then you can select your ESP8266 board in `Tools>Board` . In testing of this example, the Wemos D1 mini was used, but any board with an ESP8266 should work.

PubSubClient library

To use HiveMQ Cloud an MQTT library has to be added first. In this example the PubSubClient is used but there are a lot of other alternatives that provide similar functionalities. Download the PubSubClient from GitHub as a `.zip` file. Then go to the Arduino IDE, open `Sketch>Include Library>Add .ZIP Library` and select the downloaded `.zip` file.

NTPClient library

The next library that is required is the NTPClient. Install this library by opening `Tools>Manage Libraries...` and searching for `NTPClient` . This library provides functionality for date and time.

Help

Documentation

Feedback

Logout

LittleFS Filesystem Uploader

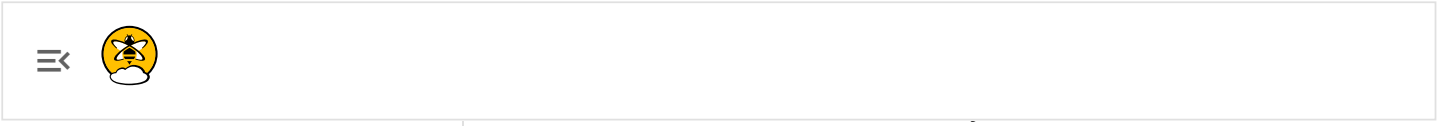
The LittleFS Filesystem Uploader is needed to upload certificates to your board. This enables you to connect securely with HiveMQ Cloud. Download the `ESP8266LittleFS-X.zip` . Go to the Arduino IDE directory and open the `tools` folder. Unzip the downloaded `.zip` folder to the `tools` folder. The result will be a folder called `ESP8266LittleFS-2.6.0` or it can have a different version. This folder contains the `ESP8266LittleFS` folder. Cut the `ESP8266LittleFS` folder and paste it into the `tools` folder. Afterwards, you can delete the `ESP8266LittleFS-2.6.0` folder. The resulting folder structure should look like this:

```
<home_dir>/Arduino-<version>/tools/ESP8266LittleFS/tool/esp826
```

On OS X create the `tools` directory in `~/Documents/Arduino/` and unpack the files there. Now restart Arduino IDE and `ESP8266 LittleFS Data Upload` will appear when opening the `tools` menu.

Upload certificates to the board

The next step is to get the certificates that are needed for an encrypted connection to HiveMQ Cloud. For this purpose use this script by downloading the repository as a `.zip` folder. Only the file `certs-from-mozilla.py` is needed, you can delete the rest. Open this script in a Python IDE of your choice, for example PyCharm or Visual Studio Code. Execute the script in your IDE. A file will be generated, that is called `certs.ar` . Move this file into the directory of your Arduino IDE sketch. To find it you can go to `Sketch>Show Sketch Folder` in Arduino IDE. The folder where your sketch is saved should open. Inside this folder, create a new folder called `data` . Put your generated `certs.ar` into this `data` folder. In the Arduino IDE, in the `Tools` menu, you may want to select a bigger flash size, this depends on the size of your files. Then open `Tools>ESP8266 LittleFS Data Upload` . This will upload the files located in `data` to your board's flash memory. After a few seconds, you should get the message `LittleFS Image Uploaded` .



Data

Clusters

Billing

Billing & Payment

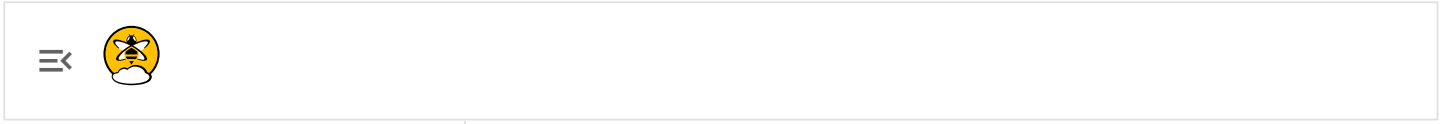
First add your WiFi information, so that the ESP8266 can connect to the internet. The `host name` of your cluster is already inserted as `mqtt_server`. Your `username` is also inserted automatically. To fully verify your credentials, replace the variable `"your_password"` with the value you entered when creating your credentials. As HiveMQ Cloud does not support insecure connections, TLS is required. A secure connection will be established by using the certificates we downloaded earlier. The default port used for secure MQTT connections is `8883`.

Help

Documentation

Feedback

Logout



Data

Clusters

Billing

Billing & Payment

What's New 🎉

Introducing our new HiveMQ Cloud Starter plan

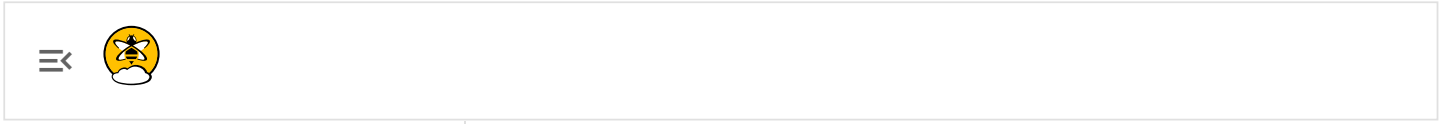
LEARN MORE

Help

Documentation

Feedback

Logout



Data

Clusters

Billing

Billing & Payment

What's New 🎉

Introducing our new HiveMQ Cloud Starter plan

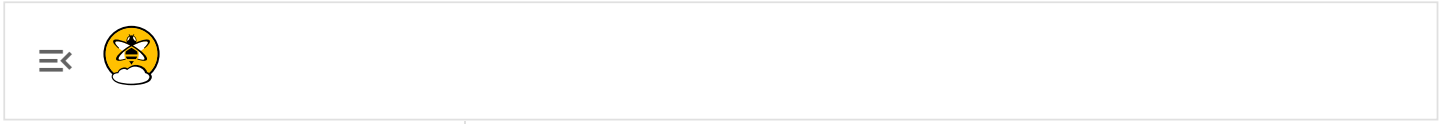
LEARN MORE

Help

Documentation

Feedback

Logout



Data

Clusters

Billing

Billing & Payment

What's New 🎉

Introducing our new HiveMQ Cloud Starter plan

LEARN MORE

Help

Documentation

Feedback

Logout

```
    }
}
```

Run this example by connecting your board to your PC via a USB-cable and uploading the sketch with the button in the Arduino IDE.

Publish and Subscribe with your MQTT client

The code shown above creates an MQTT client that is publishing and subscribing to a topic on your HiveMQ Cloud cluster. First it sets up the WiFi by using the SSID and password you set. Then the internal clock of the board is updated by getting the present time from an NTP server. This is necessary to validate the certificates that were uploaded to the board with the LittleFS Filesystem Uploader. For `callbacks`, `Message arrived` is registered. The LED of the board will also flash when a non-empty message arrives. After `setup_wifi()` and `setDateTime()`, the certificate store (`certStore`) is being initialized and made to store the certificates of `certs.ar` . Then the WiFi client will be integrated with the `certStore` . It sets up the MQTT client by using the earlier set host name and port. Afterwards it connects to the cluster using your username and password in the `reconnect()` function, that gets executed in a loop. It subscribes to the topic `"testTopic"` to which it also publishes. Then it publishes a message with the payload `"hello world"` and incrementing numbers to this topic.

Next Steps

Get familiar with the Arduino IDE API and build your first application. Further information on the PubSubClient for Arduino can be found in our MQTT Client Library Encyclopedia. Learn more about MQTT by visiting the MQTT Essentials guide, that explains the core of MQTT concepts, its features and other essential information.