# Introduction

The MQTT protocol provides message sending services to its clients by using a central distributor, a **broker**.

The MQTT protocol works with HTTP request like hierarchical endpoints. Its name is **topic**. That means, a topic can have several sub-topics.

Examples for topics:

* indoor/kitchen/coffeemachine
* indoor/kitchen/fridge
* indoor/kitchen/light
* indoor/bedroom/light
* outdoor/gate

We have 2 main “procedures”:

* subscribe
* publish

Those clients who subscribe for a specific topic receive a notification for every single new message comes from that “channel”.

It is possible to subscribe for multiple topics at the same time by using some universal characters. The “+” mark means only one sub-topic, the “#” mark means several sub-topics, but can be only at the end of the path.

Examples:

* indoor/+/light
  + E.g. matches: indoor/kitchen/light, indoor/bedroom/light
* indoor/#
  + E.g. matches: indoor/kitchen/coffeemachine,indoor/kitchen/fridge, indoor/kitchen/light, indoor/bedroom/light

# Install

## Linux

Preparation

* sudo apt-get update
* sudo apt-get upgrade

MQTT installation:

* sudo apt-get install mosquitto
* sudo apt-get install mosquitto-clients

## Windows

Navigate to <https://mosquitto.org/download/> and download the installer. Run it.

## Mac OS

Install brew package manager (https://brew.sh)

* /bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install.sh>)"

MQTT installation

* brew install mosquitto

## Additional tools

MQTT Lens chrome plugin: <https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlgmdigohjobjm?hl=hu>

MQTT python package installation:

* sudo pip3 install paho-mqtt

# Start the broker

If you would like to start a broker without any authentication (so everyone can subscribe/publish without username + password), open a terminal and run the following command:

mosquitto -p <PORTNUMBER> -v

It is possible to setting up authentication for your broker. Follow the instructions below:

1. Create a simple text file and enter the username and passwords, one for each line, with the username and password separated by a colon:

Example file (passwd.txt):

user1:passwd1

user2:passwd2

user3:passwd3

2.Now you need to convert the password file which encrypts the passwords, Go to a command line and type:

mosquitto\_passwd -U passwordfile

3.Create an acl (Access Control List) file, where you can define the rules for the users. For get more information about the syntax, read: http://www.steves-internet-guide.com/topic-restriction-mosquitto-configuration/

For an example file, see the attached *acl* file.

4.Create a configuration file like the attached *moquitto.config* file.

5.Start a broker with the following command:

mosquitto -c <CONFIGFILE> -p <PORTNUMBER>

# Sample codes

## Bash

Run these bash scripts in the given order in different terminals:

* bash subscribe\_client.sh "test/#"
  + Subscribes on every message for every topics under “test
* bash subscribe\_client.sh "test/+/a"
  + Subscribes on the “a” subtopic of all direct subtopic of “test”
* bash subscribe\_client.sh "test/b/+"
  + Subscribes on every direct subtopic of “test/b”
* bash subscribe\_client.sh "test/x/y/a"
  + Subscribes on “test/x/y/a” topic
* bash post\_client.sh
  + Publishes the given message

Output

* post\_client.sh

This script will post messages in the local MQTT broker.

TOPIC: test MESSAGE: DATA\_test

TOPIC: test/a/a MESSAGE: DATA\_test-a-a

TOPIC: test/a/b MESSAGE: DATA\_test-a-b

TOPIC: test/b/a MESSAGE: DATA\_test-b-a

TOPIC: test/b/b MESSAGE: DATA\_test-b-b

TOPIC: test/x/y/a MESSAGE: DATA\_test-x-y-a

* test/#

This client get messages. (TOPIC: test/#)

Waiting for messages... (Cancel: Ctrl+C)

DATA\_test

DATA\_test-a-a

DATA\_test-a-b

DATA\_test-b-a

DATA\_test-b-b

DATA\_test-x-y-a

* test/+/a

This client get messages. (TOPIC: test/+/a)

Waiting for messages... (Cancel: Ctrl+C)

DATA\_test-a-a

DATA\_test-b-a

* test/b/+

This client get messages. (TOPIC: test/b/+)

Waiting for messages... (Cancel: Ctrl+C)

DATA\_test-b-a

DATA\_test-b-b

* test/x/y/a

This client get messages. (TOPIC: test/x/y/a)

Waiting for messages... (Cancel: Ctrl+C)

DATA\_test-x-y-a