

Python/Python 예제 코드

MQTT 개념 및 예제

webnautes 2023. 10. 21. 17:28

MQTT 개념과 예제를 다룹니다.@

다음 링크의 문서를 번역한 내용을 바탕으로 이해한 것을 보충했습니다.

<https://learn.sparkfun.com/tutorials/introduction-to-mqtt/all>

추가적으로 아래 링크를 참고했습니다.

<https://khj93.tistory.com/entry/MQTT-MQTT의-개념>

2022. 2. 23 최초작성

2023. 5. 11 추가로 다음 링크를 참고했습니다.

<https://1000sj.tistory.com/307>

<https://developer-finn.tistory.com/1>

<https://dev.to/emqx/how-to-use-mqtt-in-python-paho-3b8>

<http://www.steves-internet-guide.com/client-connections-python-mqtt/>

Mqtt를 사용하여 ESP32 두개에 각각 있는 버튼, 스위치를 제어하는 예제는 포스트에서 제거했습니다. 관련 코드는 다음 링크에서 참고하세요.

<https://learn.sparkfun.com/tutorials/introduction-to-mqtt/all>

2023. 5. 18

2023. 6. 26 윈도우에서 Mosquitto 설치 추가

MQTT 개념

MQTT는 발행/구독 메시징 프로토콜(publish/subscribe messaging protocol)입니다.

MQTT(Message Queue Telemetry Transport)는 M2M (machine to machine) 이나 IoT(internet of Things) 에서 사용됩니다. 낮은 대역폭(bandwidth), 높은 지연 시간(latency) 같은 제한된 네트워크 환경에 적합하기 때문입니다.

MQTT는 브로커(Broker), 발행자(Publisher), 구독자(Subscriber)로 구성됩니다.

브로커(Broker)에 연결하여 메시지를 보내거나 받는 장치를 클라이언트(Client)라고 합니다.

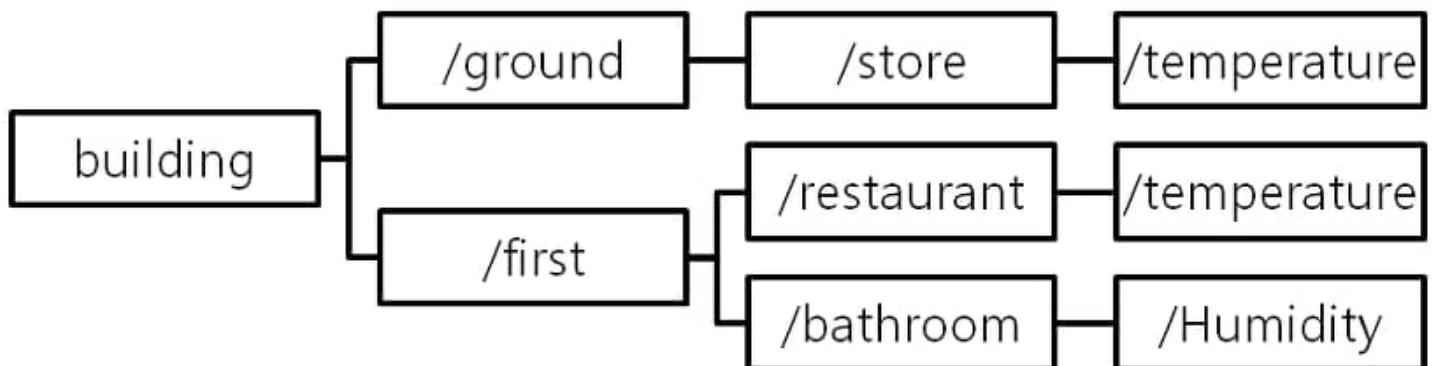
메시지를 전송하는 발행자(Publisher) 클라이언트와 메시지를 수신하는 구독자(Subscriber) 클라이언트가 있습니다.

메시지는 전송하고자 하는 데이터라고 생각하면 됩니다.

발행자 클라이언트와 구독자 클라이언트는 직접 메시지를 주고 받지 않고 브로커를 통해서 메시지를 주고 받습니다. 이때 토픽(Topic)을 사용합니다. 토픽은 슬래시(/)를 분리 문자로 사용하여 폴더 및 파일과 유사한 계층 구조를 가집니다. 토픽은 정하기 나름입니다.

토픽은 디렉토리 와 같은 구조로 상위 토픽과 하위 토픽으로 구성되어 배치됩니다. 상위 토픽 내에 여러 클라이언트가 있는 경우 토픽은 "LivingRoom" 또는 "LivingRoom/Light"일 수 있습니다.

아래 그림은 상위 토픽과 하위 토픽으로 구성된 예입니다.

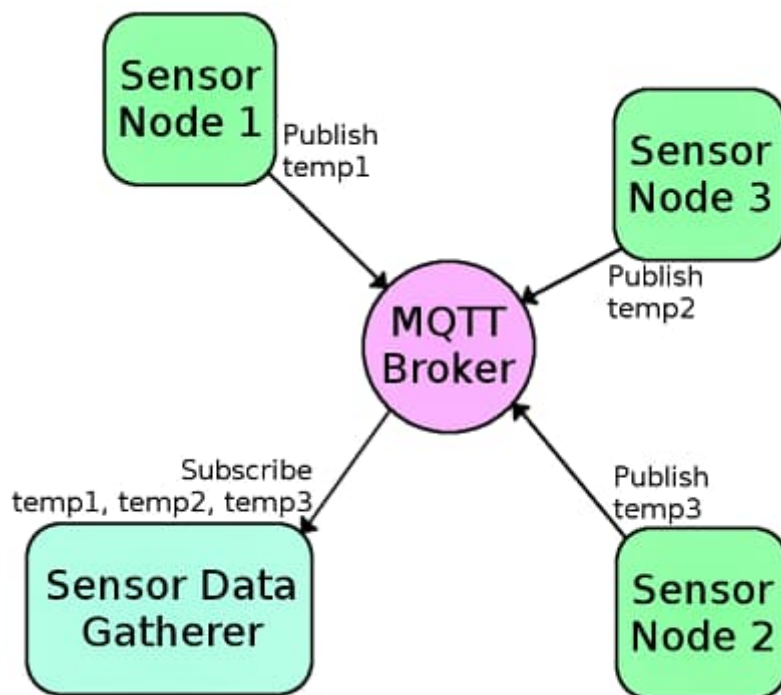


출처 - <https://khj93.tistory.com/entry/MQTT-MQTT의-개념>

구독자 클라이언트가 브로커에게 관심 있는 토픽을 구독한다고 알려주면 소켓을 생성하는 것처럼 메시지를 주고받을 수 있는 연결통로가 생깁니다. 해당 토픽에 발행자 클라이언트가 메시지를 배포하면 브로커는 배포된 해당 토픽의 모든 메시지를 해당 토픽을 구독한 모든 구독자 클라이언트에게 송신합니다.

구독자 클라이언트는 주기적으로 체크하는 폴링 방식으로 브로커에서 구독한 토픽에 해당되는 메시지를 수신하며 구독자 클라이언트는 해당 토픽에 대한 브로커로부터의 메시지 수신을 중지하기 위해 구독을 취소할 수도 있습니다.

아래 그림에서 Sensor Node 1, Sensor Node 2, Sensor Node 3은 브로커를 통해 각각 토픽 temp1, temp2, temp3에 메시지를 배포하는 발행자 클라이언트입니다. Sensor Data Gatherer는 토픽 temp1, temp2, temp3를 구독했기 때문에 브로커를 통해 해당 토픽의 메시지를 수신받습니다.



발행자 클라이언트와 구독자 클라이언트는 브로커에 연결시 QoS(서비스 품질, Quality of Service)을 지정할 수 있습니다.

QoS는 TCP 데이터 전송 처리에 영향을 주지 않고 MQTT 클라이언트 사이에서만 영향을 미칩니다.

0 으로 지정하면 최대 한번 송신하며, 정상적으로 보내졌는지는 확인하지 않습니다.

1 으로 지정하면 적어도 한번 송신하며 정상적으로 보내졌는지 확인될 때까지 여러 번 전송합니다. 같은 메시지를 중복 송신할 가능성이 있습니다

2 를 지정하면 메시지를 한 번만 송신합니다. 발신자 및 수신자 클라이언트가 2단계 핸드셰이크를 사용하기 때문에 메시지가 한번 전송되는 것을 보장하지만 다른 방식에 비해 성능이 떨어집니다.

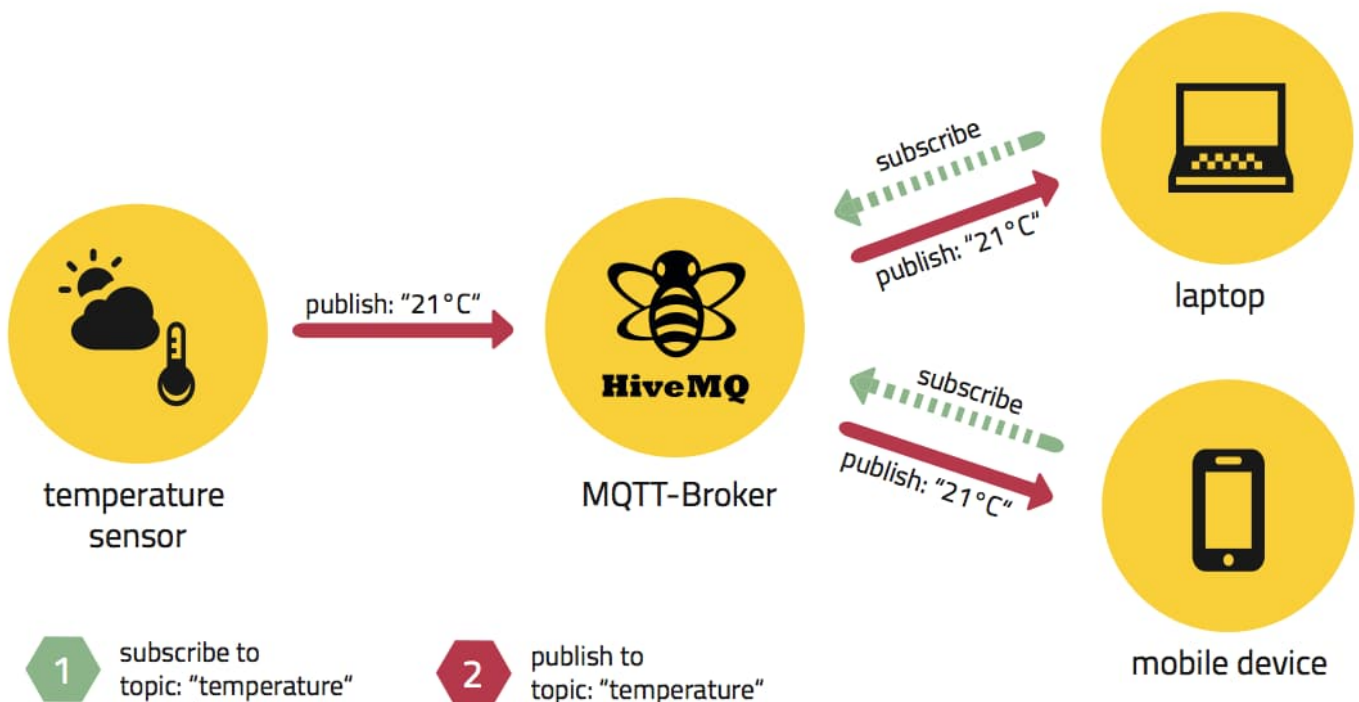
MQTT 동작

MQTT가 동작하려면 발행자/구독자 클라이언트, 브로커가 서로 통신이 가능한 네트워크에 연결되어야 합니다. 구독자 클라이언트가 토픽을 구독한 후, 발행자 클라이언트가 해당 토픽에 메시지를 배포하면 메시지가 브로커로 전송되기 시작하고 브로커는 해당 주제를 구독한 모든 구독자 클라이언트에게 해당 메시지를 전송합니다.

구독자 클라이언트는 구독한 주제에서 들어오는 메시지를 수신하고 "켜기" 또는 "끄기"와 같이 해당 토픽에 게시된 내용에 따라 전등같은 하드웨어를 제어할 수 있습니다.

하나의 클라이언트가 구독자 클라이언트 역할과 발행자 클라이언트 역할을 동시에 할 수도 있습니다. 예를 들어 클라이언트가 조명을 제어하는 "LivingRoom/Light"을 구독하여 해당 메시지 내용에 따라 조명 상태를 바꾸면서 동시에 다른 클라이언트가 해당 조명의 상태를 모니터링할 수 있도록 "LivingRoom/Light/State"와 같은 토픽에 조명의 상태가 저장된 메시지를 배포할 수 있습니다.

아래 그림에서 laptop과 mobile device가 브로커를 통해 토픽 temperature를 구독하고 온도가 포함된 메시지가 도착하기를 대기합니다. temperature sensor는 브로커를 통해 토픽 temperature에 온도를 배포합니다. 브로커는 토픽 temperature를 구독한 laptop과 mobile device에 온도를 전달합니다.



브로커 세팅

사용할 수 있는 MQTT 브로커로는 Mosquitto, Rabbit MQ 등이 있습니다. 본 문서에서는 Mosquitto를 사용합니다.

우분투와 윈도우에서 하는 방법을 각각 다룹니다.

우분투

패키지 목록을 새로 받아온 후, mosquitto 패키지를 설치합니다.

```
$ sudo apt update
```

```
$ sudo apt install mosquitto
```

설치 하고나면 브로커 역할을 하는 mosquitto가 자동으로 실행된 상태가 됩니다. 해당 프로세스를 종료시키고 다시 실행시킬 필요는 없습니다.

터미널 2개를 사용하여 한쪽에는 구독자 클라이언트를 실행하고, 다른 쪽에는 발행자 클라이언트를 실행합니다. 그러기 위해서 추가로 필요한 mosquitto-clients 패키지를 설치합니다.

```
$ sudo apt install mosquitto-clients
```

현재 브로커가 실행된 PC의 IP를 확인해야 합니다. 그러려면 추가 패키지를 설치해야 합니다.

```
$ sudo apt install net-tools
```

이제 IP를 확인합니다. ifconfig를 실행하면 보통 다음처럼 실행됩니다. lo가 아닌 곳에 있는 inet 항목이 현재 PC가 사용중인 IP입니다.

```
$ ifconfig
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

```
    inet 127.0.0.1 netmask 255.0.0.0
```

```
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
```

```
    loop txqueuelen 1000 (Local Loopback)
```

```
    RX packets 1825 bytes 183984 (183.9 KB)
```

```
    RX errors 0 dropped 0 overruns 0 frame 0
```

```
    TX packets 1825 bytes 183984 (183.9 KB)
```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.25.5 netmask 255.255.255.192 broadcast 192.168.25.63

하지만 브로커가 실행된 PC의 아이피를 사용하면 Error: Connection refused에러가 발생했습니다. 이 부분은 확인을 해봐야 겠습니다.

지금은 대신 브로커 아이피 적을 곳에 localhost를 사용합니다.

먼저 구독자 클라이언트를 실행해야 합니다. 실행할 명령의 형식은 다음과 같습니다.

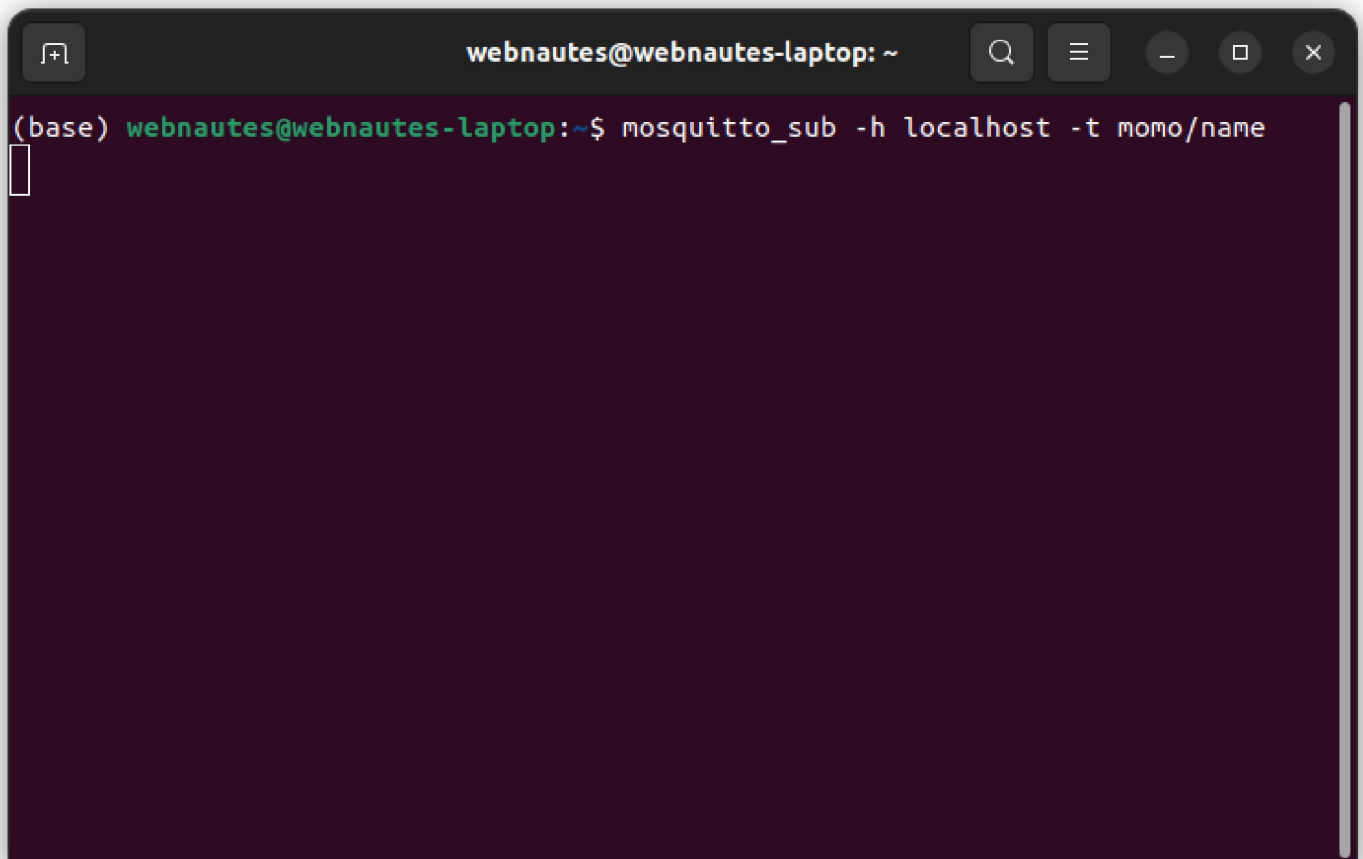
`mosquitto_sub -h <브로커의 IP> -t <TOPIC>`

Ctrl + Alt + T를 눌러 터미널을 열은 후, 해당 명령을 실행해봅니다.

구독자 클라이언트가 토픽 momo/name을 구독한다고 localhost에서 실행중인 브로커에게 알립니다.

`$ mosquitto_sub -h localhost -t momo/name`

메시지를 기다리는 대기상태가 됩니다.

A terminal window titled 'webnautes@webnautes-laptop: ~' with standard window controls. The prompt is '(base) webnautes@webnautes-laptop:~\$' and the command 'mosquitto_sub -h localhost -t momo/name' has been entered. A cursor is visible on the line following the command.

```
(base) webnautes@webnautes-laptop:~$ mosquitto_sub -h localhost -t momo/name
```

이제 발행자 클라이언트를 실행할 차례입니다. 실행할 명령의 형식은 다음과 같습니다.

구독자에서 지정한 TOPIC을 사용해야 합니다.

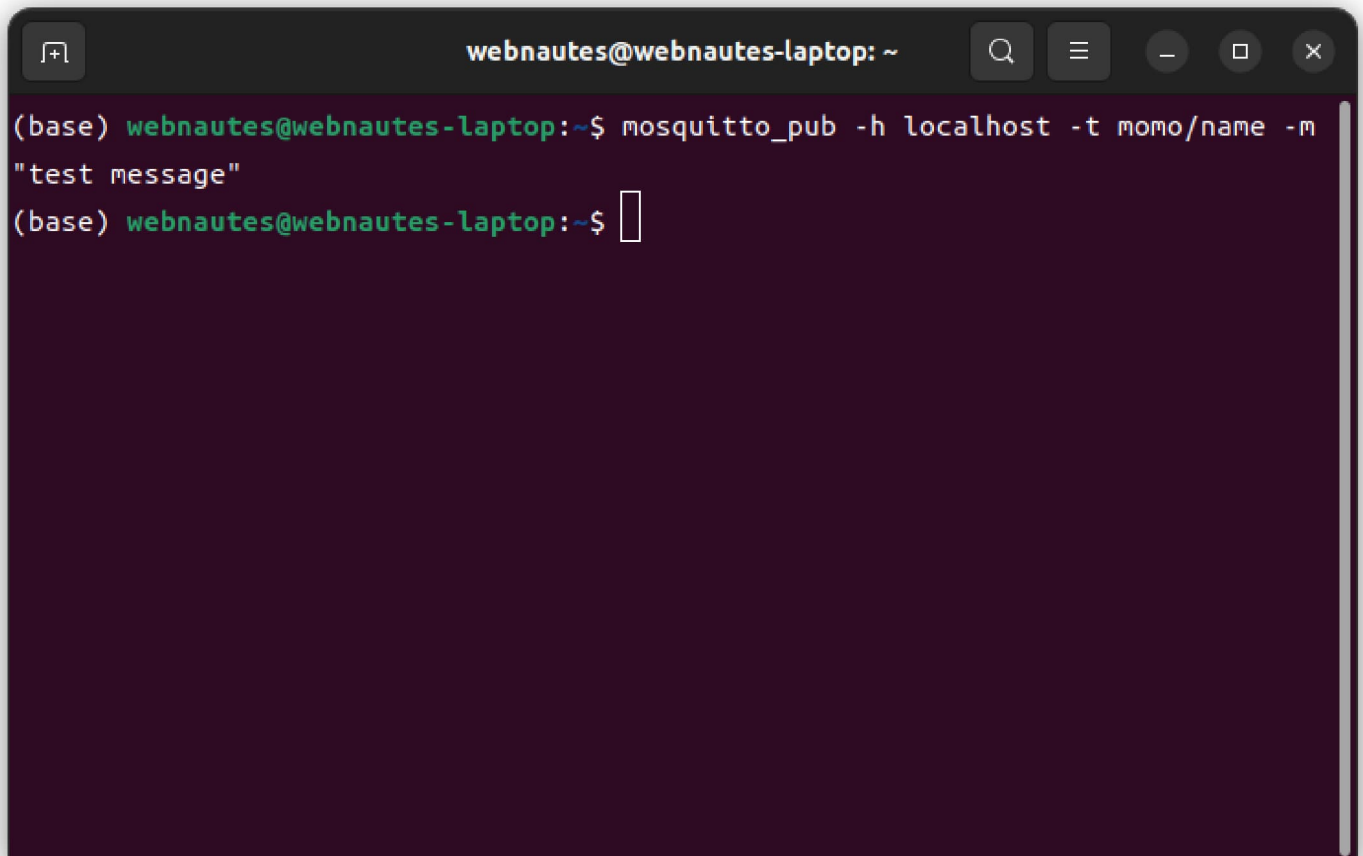
```
mosquitto_pub -h <브로커의 IP> -t <TOPIC> -m "<MESSAGE>"
```

터미널을 하나 더 열어서 다음 명령을 실행합니다.

발행자 클라이언트가 토픽 momo/name에 메시지 “test message”를 배포하겠다고 localhost에서 실행 중인 브로커에게 알립니다.

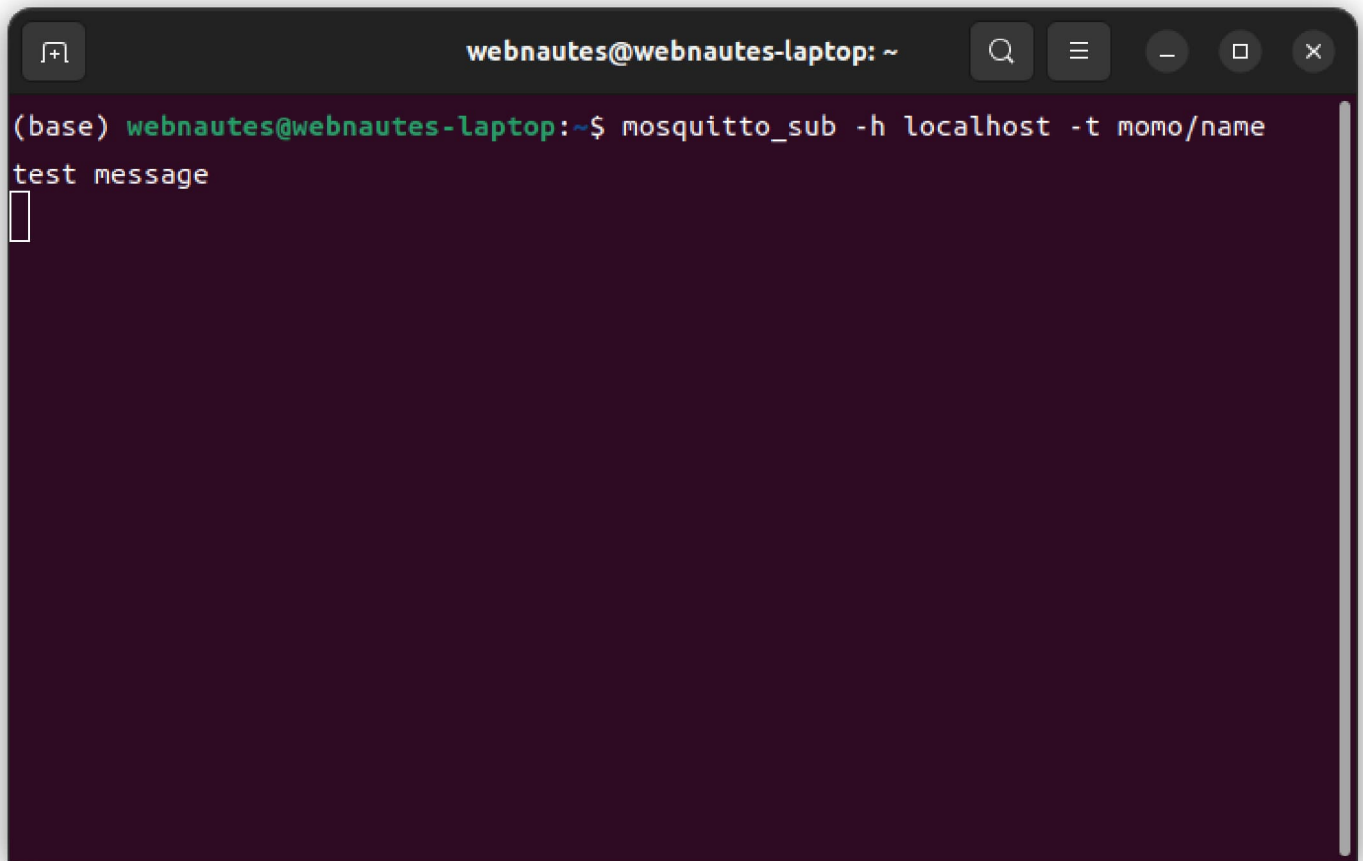
```
$ mosquitto_pub -h localhost -t momo/name -m "test message"
```

실행 후 바로 종료됩니다.

A terminal window titled 'webnautes@webnautes-laptop: ~' with search, menu, and window control icons in the title bar. The terminal shows a command prompt '(base) webnautes@webnautes-laptop:~\$' followed by the command 'mosquitto_pub -h localhost -t momo/name -m "test message"'. The command is executed, and the prompt returns to '(base) webnautes@webnautes-laptop:~\$' with a cursor, indicating the command has completed successfully.

```
(base) webnautes@webnautes-laptop:~$ mosquitto_pub -h localhost -t momo/name -m "test message"
(base) webnautes@webnautes-laptop:~$
```

구독자 클라이언트를 실행해 놓은 터미널을 보면 브로커를 통해 전달받은 메시지가 출력되어 있습니다. 다시 대기 상태가 됩니다.

A terminal window titled 'webnautes@webnautes-laptop: ~' with standard window controls. The prompt is '(base) webnautes@webnautes-laptop:~\$'. The command 'mosquitto_sub -h localhost -t momo/name' has been executed, and the output 'test message' is displayed on the next line. A cursor is visible on the line following the output.

```
(base) webnautes@webnautes-laptop:~$ mosquitto_sub -h localhost -t momo/name
test message
```

정리하면 구독자 클라이언트를 먼저 실행해놓은 상태에서 발행자 클라이언트를 실행해야 메시지가 전달됩니다.

구독자 클라이언트가 실행 안된상태에서 발행자 클라이언트가 보낸 메시지는 사라지는 것을 확인할 수 있습니다.

다음처럼 데몬으로 브로커 역할을 하는 mosquitto의 실행을 제어할 수 있습니다.

브로커 사용을 중지합니다.

```
$ sudo /etc/init.d/mosquitto stop
```

Stopping mosquitto (via systemctl): mosquitto.service.

브로커를 다시 사용합니다.

```
$ sudo /etc/init.d/mosquitto start
```

Starting mosquitto (via systemctl): mosquitto.service.

윈도우

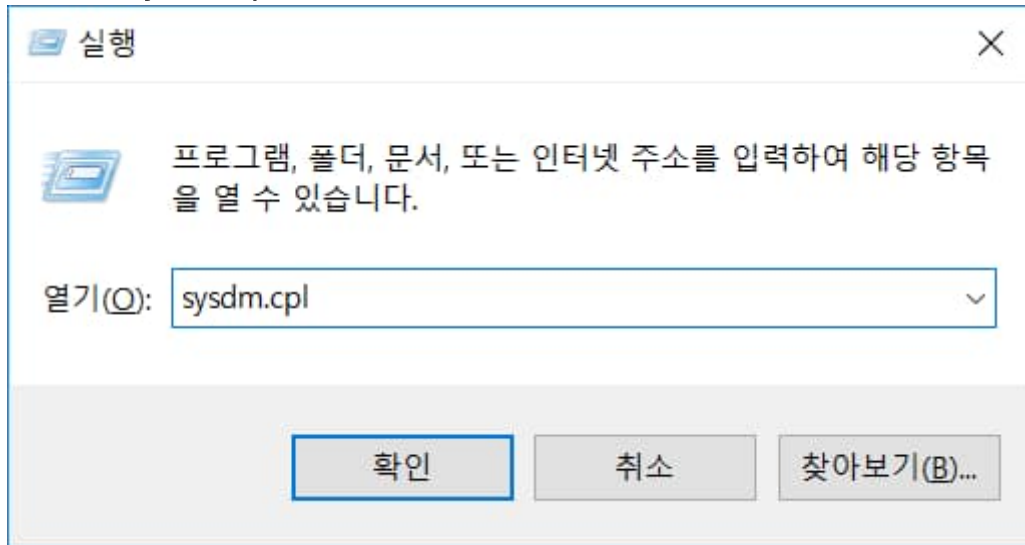
다음 링크에서 윈도우용 Mosquitto를 다운로드 받아 설치한다.

<https://mosquitto.org/download/>

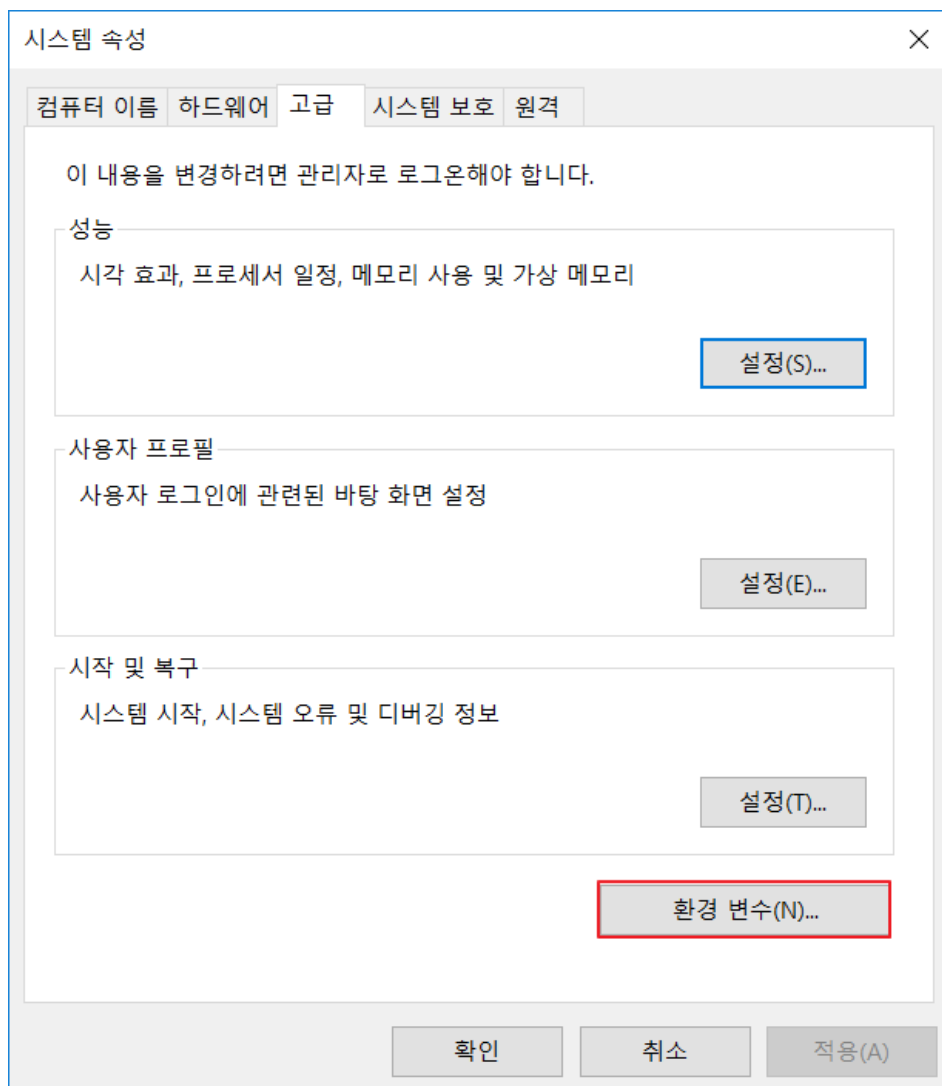
파일 이름에 x64가 있는 파일을 다운로드하면 됩니다. 포스트 작성 시점에서는 mosquitto-2.0.15-install-windows-x64.exe를 사용했습니다.

C:\Program Files\mosquitto를 환경변수 path에 등록해야합니다.

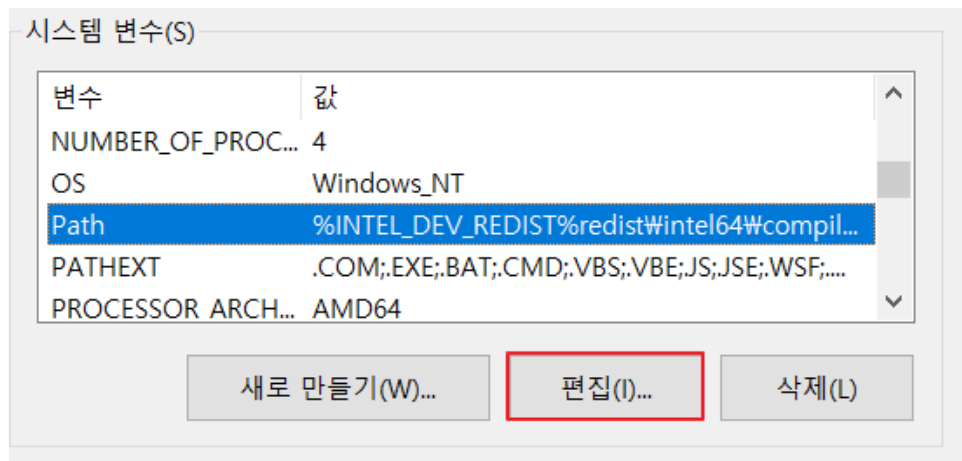
윈도우키 + R를 누른 후, sysdm.cpl을 입력하여 제어판의 시스템 속성을 실행시킵니다.



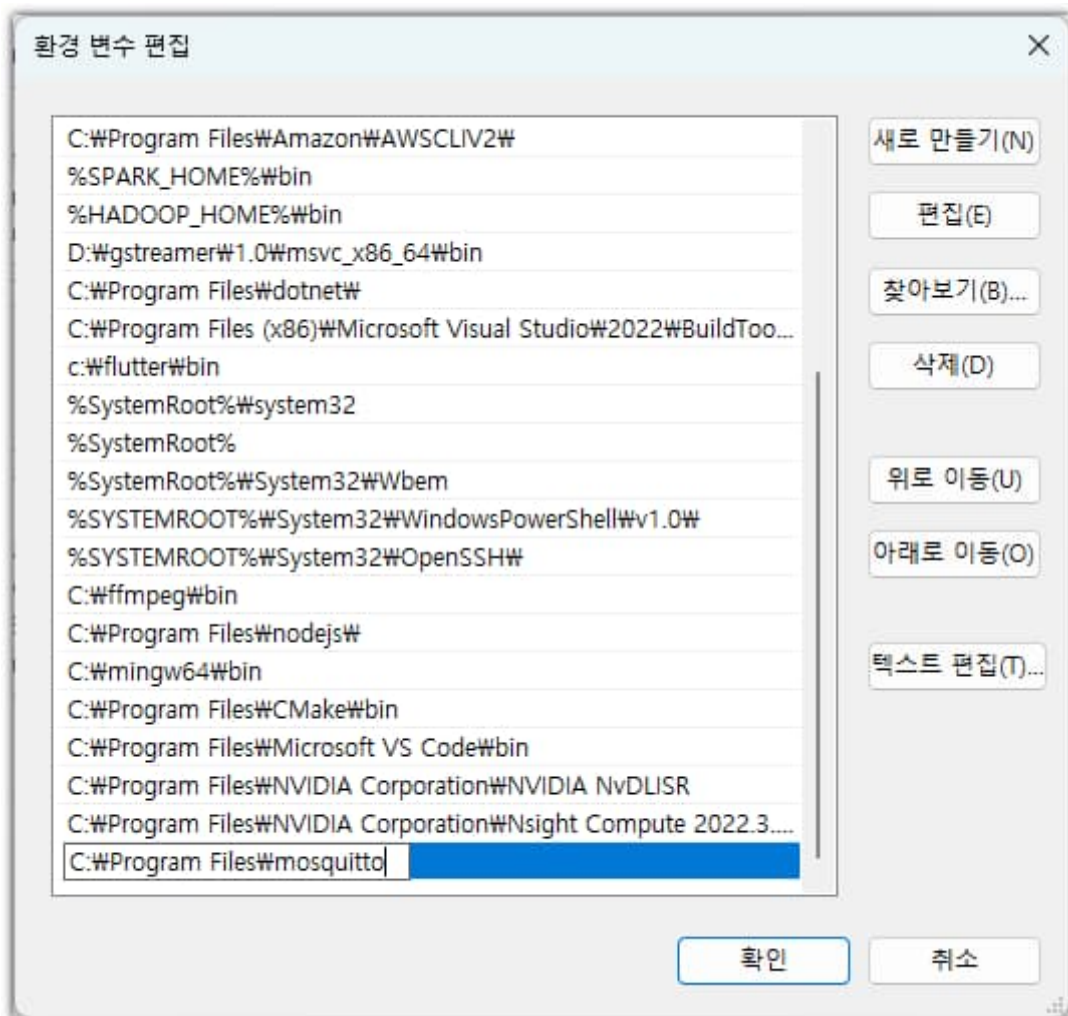
시스템 속성의 고급 탭에서 환경 변수 버튼을 클릭합니다.



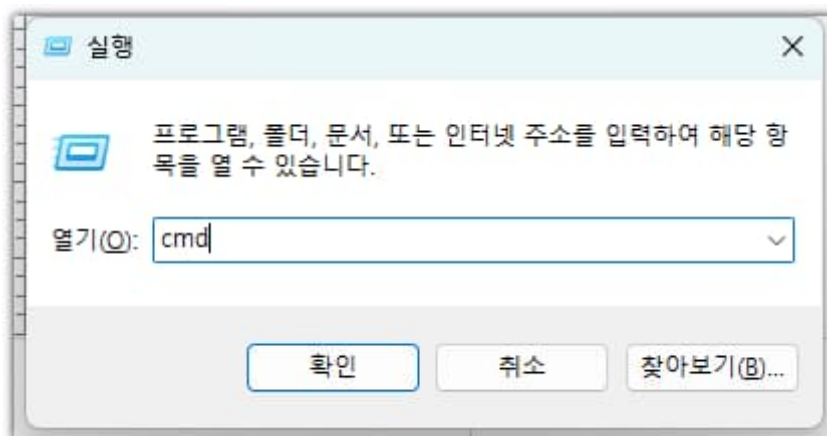
시스템 변수에서 Path를 선택하고 편집 버튼을 클릭합니다.



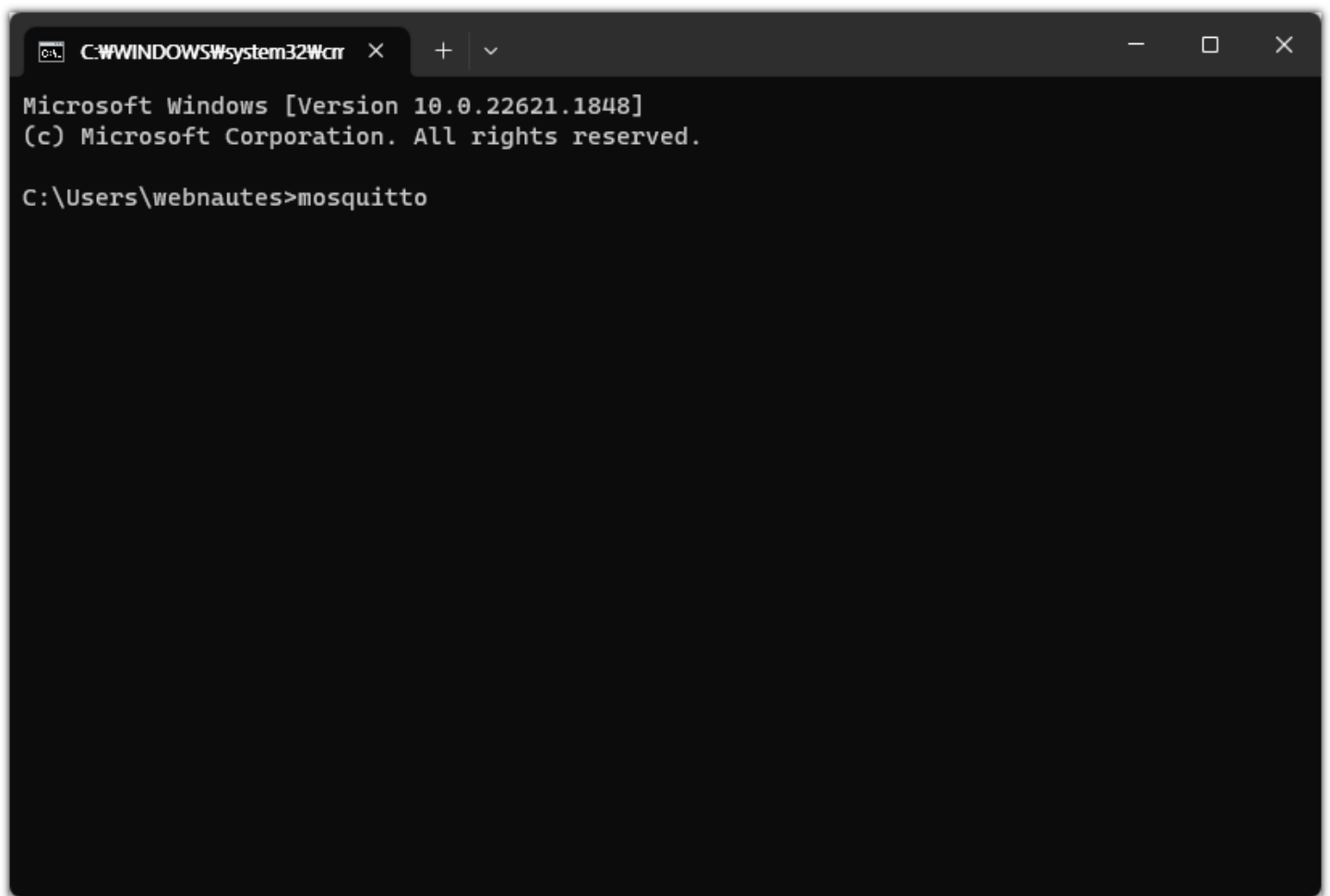
새로 만들기 버튼을 클릭 후, C:\Program Files\mosquitto를 붙여넣어 추가합니다. 이제 확인을 클릭합니다.



윈도우 키 + R을 누른 후, cmd를 입력하여 명령 프롬프트를 실행합니다.



명령 프롬프트에서 mosquitto를 실행합니다. 아무 것도 출력되지 않고 대기 상태가 됩니다.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\webnautes>mosquitto
```

윈도우 키 + R을 누른 후, cmd를 입력하여 명령 프롬프트를 실행합니다.

다음 명령을 사용하여 토픽 "momo/name"을 구독합니다.

```
mosquitto_sub -t "momo/name" -d
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\webnautes>mosquitto_sub -t "momo/name" -d
Client null sending CONNECT
Client null received CONNACK (0)
Client null sending SUBSCRIBE (Mid: 1, Topic: momo/name, QoS: 0, Options: 0x00)
Client null received SUBACK
Subscribed (mid: 1): 0
```

윈도우 키 + R을 누른 후, cmd를 입력하여 명령 프롬프트를 실행합니다.

다음 명령을 사용하여 토픽 "momo/name"을 발행합니다.

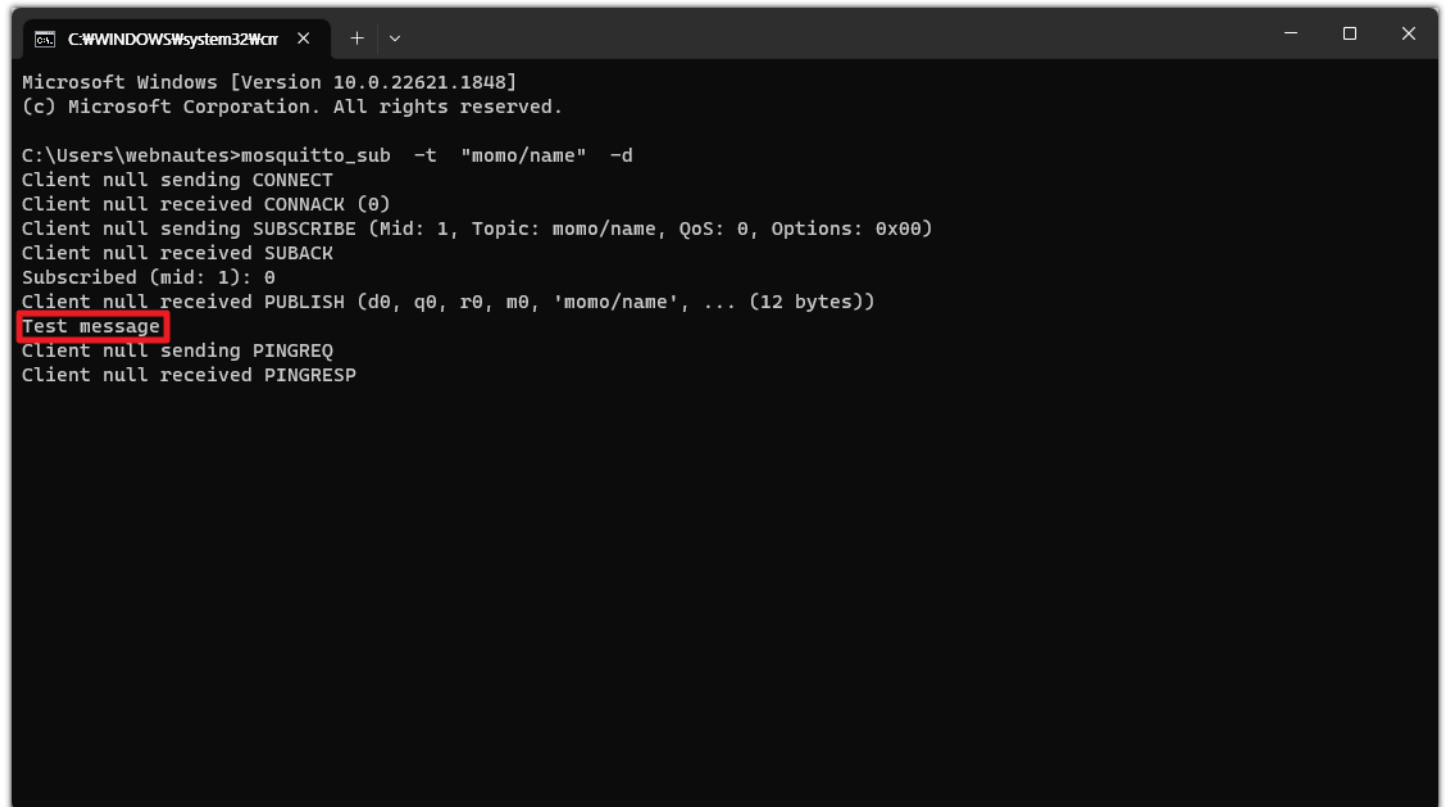
```
mosquitto_pub -i mosq_pub1 -t "momo/name" -m "Test message" -d
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\webnautes>mosquitto_pub -i mosq_pub1 -t "momo/name" -m "Test message" -d
Client mosq_pub1 sending CONNECT
Client mosq_pub1 received CONNACK (0)
Client mosq_pub1 sending PUBLISH (d0, q0, r0, m1, 'momo/name', ... (12 bytes))
Client mosq_pub1 sending DISCONNECT

C:\Users\webnautes>
```

토픽을 구독했던 명령 프롬프트를 확인해보면 메시지가 수신된 것을 볼 수 있습니다.



```
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\webnautes>mosquitto_sub -t "momo/name" -d
Client null sending CONNECT
Client null received CONNACK (0)
Client null sending SUBSCRIBE (Mid: 1, Topic: momo/name, QoS: 0, Options: 0x00)
Client null received SUBACK
Subscribed (mid: 1): 0
Client null received PUBLISH (d0, q0, r0, m0, 'momo/name', ... (12 bytes))
Test message
Client null sending PINGREQ
Client null received PINGRESP
```

Python으로 MQTT 통신

Python으로 MQTT 통신을 하려면 paho-mqtt 패키지를 설치해야 합니다.

```
$ pip install paho-mqtt
```

진행하기전 브로커가 실행된 상태여야 합니다.

파이썬코드의 실행순서도 중요합니다.

구독자 클라이언트 코드를 먼저 실행해놓은 상태에서 발행자 클라이언트 코드를 실행해야 메시지가 전달됩니다.

참고할 수 있는 샘플 코드는 아래 링크에 있습니다.

https://github.com/eclipse/paho.mqtt.python/blob/master/examples/client_sub.py

https://github.com/eclipse/paho.mqtt.python/blob/master/examples/client_pub_wait.py

테스트시 브로커의 IP 주소로는 'localhost'를 사용하세요.
또한 토픽도 일치시켜야 합니다.

1

구독하기

응원하기

천천히 하지만 꾸준히 공부한 내용을 블로그에 공유하는 것이 제 취미생활입니다.

해본 결과를 바탕으로 포스트를 작성하지만 시간이 경과하면
똑같이해도 동작안할 수 있습니다.

문제 생겼을시 댓글로 알려주시면 가능한 빨리 답변해드릴게요.

문제가 생기면 포스트와 다르게 진행한 부분을 먼저 확인해보세요.

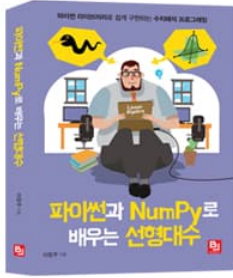
[토스 아이디어라는게 있어서 만들어봤어요.](#)

제가 쓴 책도 한번 검토해보세요 ^^



파이썬과 C++ 예제로 마스터하는 OpenCV 프로그래밍

비제이퍼블릭 | 이정주 지음 | 알짜배기 예제로 배우는 OpenCV



프로그래밍에서 점점 더 중요해지는 선형대수 이제 파이썬과 NumPy로 쉽고 재미있게 이해한다!

비제이퍼블릭 | 이정주 지음 | 파이썬과 NumPy로 배우는 선형대수

Tag

MQTT

Python

'Python/Python 예제 코드'의 다른글

이전글 파이썬 웹캠 영상 전송 TCP 서버/클라이언트 예제

현재글 : MQTT 개념 및 예제

다음글 Python Thread 예제

관련글

두개의 리스트를 같은 순서로
정렬시키는 파이썬 예제
2023.10.21

Python Thread 예제
2023.10.21

파이썬 웹캠 영상 전송 TCP 서버/
클라이언트 예제
2023.10.19

응원하기

webnautes님의
글이 좋았다면 을 보내주세요!

 응원하기