

(IoT Network) Practice -9-

MQTT client in Node.js



Index

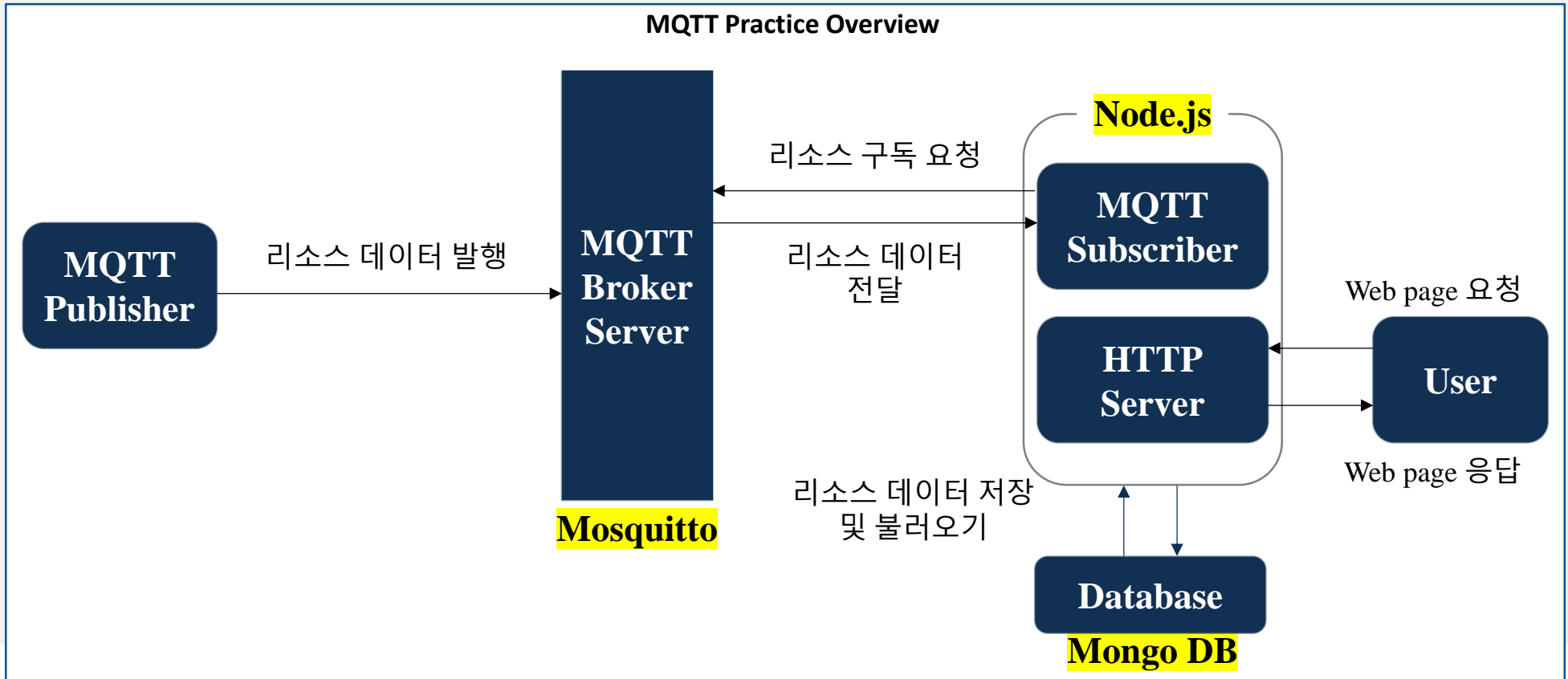
- I. Practice Overview
- II. Express Project Components & Run
- III. MQTT Publisher using Mosquitto
- IV. MQTT Client in Node.js



Practice Overview

• 실습목표

- Express Project의 구성 및 동작 이해
- Mosquitto를 사용한 MQTT Publisher 구현
- Node.js에서 MQTT Client (Subscriber) 구현
- 단, MQTT Client를 통해 수신한 데이터를 Web 페이지에 띄우는 것은 다음 실습에서 진행



Express Project Review & Run

• Express Project Components

```

▼ IoT_practice
  ▼ bin
    ≡ www
  ▼ public
    > images
    > javascripts
    > stylesheets
  ▼ routes
    JS index.js
    JS users.js
  ▼ views
    🐼 error.jade
    🐼 index.jade
    🐼 layout.jade
  JS app.js
  {} package.json

```

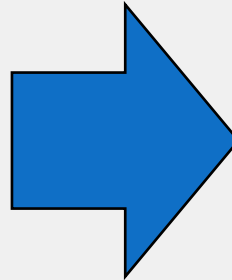
- **www**: start up script (최초로 Web 서버를 실행하기 위한 Script)
 1. HTTP 서버 생성 (이때, 서버의 Port 번호 지정)
 2. Middleware(혹은 node module)와 HTTP 서버 연동
 3. 그 외 필요한 Web 서버의 기능 초기화
- **public**: 그림, 스타일, 스크립트와 같은 모든 정적 파일 포함
- **routes**: 요청에 대한 응답 처리 (웹 페이지 랜더링)
- **views**: 웹페이지 템플릿
- **app.js**: express 객체를 생성 및 실질적인 서비스 호스팅
 1. node module을 로딩
 2. 서비스에 필요한 변수와 객체 선언
 3. 요청을 수신했을 때 Router에 유입이 이루어 지는 유입점의 역할
- **package.json**: node module dependency 정의
 1. 프로젝트에 필요한 노드 모듈 정의 및 관리

Express Project Review & Run

- Express Project Run

- package.json 수정

```
1  {  
2    "name": "iot-practice",  
3    "version": "0.0.0",  
4    "private": true,  
5    "scripts": {  
6      "start": "node ./bin/www"  
7    },  
8    "dependencies": {  
9      "cookie-parser": "~1.4.4",  
10     "debug": "~2.6.9",  
11     "express": "~4.16.1",  
12     "http-errors": "~1.6.3",  
13     "jade": "~1.11.0",  
14     "morgan": "~1.9.1"  
15   }  
16 }
```



```
1  {  
2    "name": "iot-practice",  
3    "version": "0.0.0",  
4    "private": true,  
5    "scripts": {  
6      "start": "node ./bin/www"  
7    },  
8    "dependencies": {  
9      "cookie-parser": "~1.4.4",  
10     "debug": "~2.6.9",  
11     "express": "~4.16.1",  
12     "http-errors": "~1.6.3",  
13     "jade": "~1.11.0",  
14     "morgan": "~1.9.1",  
15     "mqtt": "^4.3.7",  
16     "mongodb": "^4.6.0",  
17     "socket.io": "^4.5.0"  
18   }  
19 }
```

Express Project Review & Run

- **Express Project Run**

- 터미널에서 IoT_practice 프로젝트 폴더로 위치 이동
 - ✓ 명령어: **cd IoT_practice**

```
PS C:\MQTT_practice> cd IoT_practice
PS C:\MQTT_practice\IoT_practice> |
```

- 터미널에서 IoT_practice의 필요한 node module 설치
 - ✓ 명령어: **npm install**

```
PS C:\MQTT_practice\IoT_practice> npm install
npm WARN deprecated constantinople@3.0.2: Please update to at least constantinople 3.1.1
npm WARN deprecated transformers@2.1.0: Deprecated, use jstransformer
npm WARN deprecated jade@1.11.0: Jade has been renamed to pug, please install the latest version of pug instead of jade

added 99 packages, and audited 100 packages in 7s

5 vulnerabilities (1 low, 4 critical)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Express Project Review & Run

- **Express Project Run**

- IoT_practice 프로젝트 실행

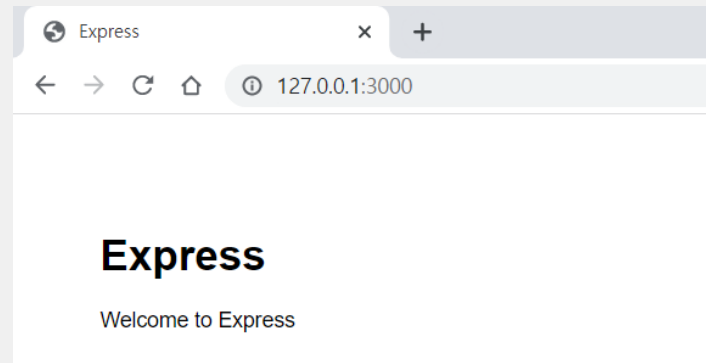
- ✓ 명령어: **npm start**

- ✓ 실행중단: **Ctrl + c** → **yes** 입력

```
PS C:\MQTT_practice\IoT_practice> npm start  
  
> iot-practice@0.0.0 start  
> node ./bin/www  
  
█
```

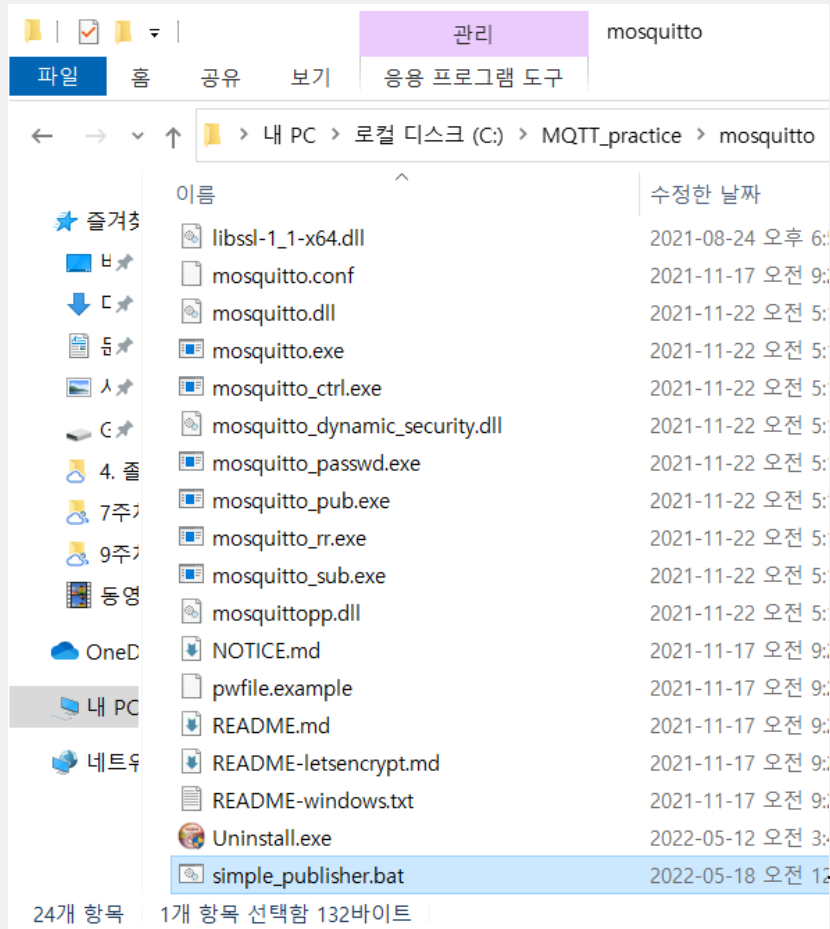
- IoT_practice 프로젝트 실행 후 웹 브라우저로 서버 동작 확인

- ✓ URL: 127.0.0.1:3000 혹은 localhost:3000



MQTT Publisher using Mosquitto

- Mosquitto를 사용한 MQTT Publisher
 - Mosquitto가 설치된 폴더에 다음과 같이 bat 파일 저장

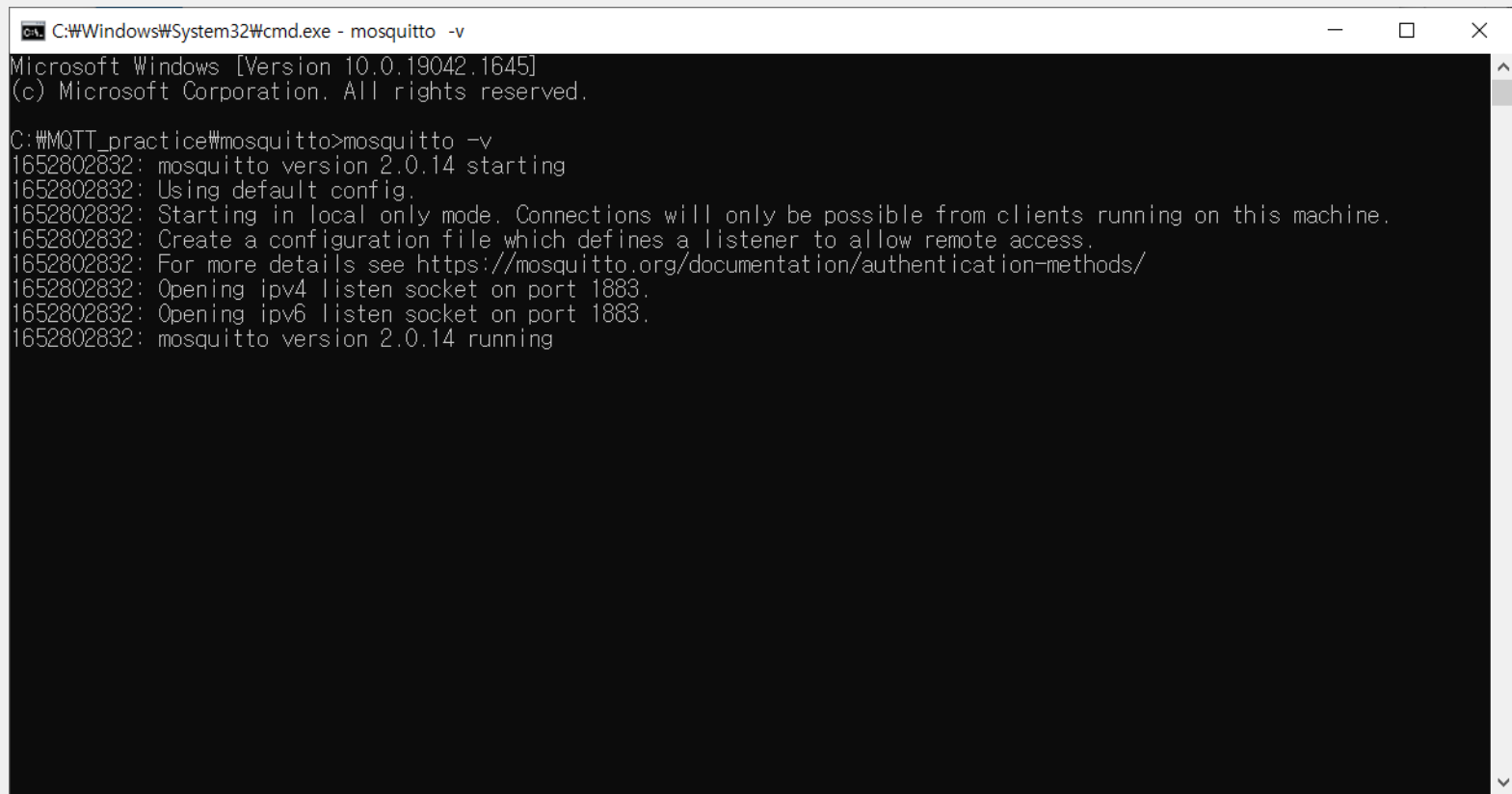


```
@echo on
:_loop
set /a temp=%random% %%30+1
mosquitto_pub -t tmp -m "{W\"tmpW\" : %temp%}"
timeout /t 2 /nobreak > nul
goto _loop
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

MQTT Publisher using Mosquitto

- Mosquitto를 사용한 MQTT Publisher
 - Mosquitto MQTT Broker 실행
 - ✓ 명령어: mosquitto -v



```
C:\Windows\System32\cmd.exe - mosquitto -v
Microsoft Windows [Version 10.0.19042.1645]
(c) Microsoft Corporation. All rights reserved.

C:\MQTT_practice\mosquitto>mosquitto -v
1652802832: mosquitto version 2.0.14 starting
1652802832: Using default config.
1652802832: Starting in local only mode. Connections will only be possible from clients running on this machine.
1652802832: Create a configuration file which defines a listener to allow remote access.
1652802832: For more details see https://mosquitto.org/documentation/authentication-methods/
1652802832: Opening ipv4 listen socket on port 1883.
1652802832: Opening ipv6 listen socket on port 1883.
1652802832: mosquitto version 2.0.14 running
```

MQTT Publisher using Mosquitto

- Mosquitto를 사용한 MQTT Publisher
 - MQTT publisher 실행

```
C:\WINDOWS\system32\cmd.exe
C:\MQTT_practice\mosquitto>set /a temp=473 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{\"tmp\" : 24}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
C:\MQTT_practice\mosquitto>goto _loop
C:\MQTT_practice\mosquitto>set /a temp=5480 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{\"tmp\" : 21}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
C:\MQTT_practice\mosquitto>goto _loop
C:\MQTT_practice\mosquitto>set /a temp=21954 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{\"tmp\" : 25}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
C:\MQTT_practice\mosquitto>goto _loop
C:\MQTT_practice\mosquitto>set /a temp=11922 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{\"tmp\" : 13}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
```

MQTT Client in Node.js

- MQTT Client 생성 및 구독

- Express 프로젝트의 www 파일을 다음과 같이 수정

```
18  ✓ /**
19    * Create HTTP server.
20    */
21
22    var server = http.createServer(app);
23
24  ✓ /**
25    * Listen on provided port, on all network interfaces.
26    */
```

```
18  /**
19    * Create HTTP server.
20    */
21
22    var server = http.createServer(app);
23
24    /**
25     * Create MQTT client
26     */
27    var mqtt = require('mqtt');
28    var mqtt_client = mqtt.connect("mqtt://127.0.0.1:1883");
29    console.log("test");
30    mqtt_client.on("connect", function(){
31      mqtt_client.subscribe("tmp");
32      console.log("Subscribing tmp");
33    });
34
35    mqtt_client.on("message", function(topic, message){
36      console.log(topic+": " +message.toString());
37      var obj=JSON.parse(message);
38      obj.creat_at = new Date();
39      console.log(obj);
40    });
```

MQTT Client in Node.js

- MQTT Client 동작 확인

- Express 프로젝트, MQTT publisher, MQTT broker 실행

```
문제 출력 터미널 디버그 콘솔

{ tmp: 30, creat_at: 2022-05-17T17:04:10.225Z }
tmp:{"tmp" : 17}
{ tmp: 17, creat_at: 2022-05-17T17:04:12.278Z }
tmp:{"tmp" : 4}
{ tmp: 4, creat_at: 2022-05-17T17:04:14.231Z }
tmp:{"tmp" : 7}
{ tmp: 7, creat_at: 2022-05-17T17:04:16.275Z }
tmp:{"tmp" : 16}
{ tmp: 16, creat_at: 2022-05-17T17:04:18.216Z }
tmp:{"tmp" : 29}
{ tmp: 29, creat_at: 2022-05-17T17:04:20.264Z }
tmp:{"tmp" : 26}
{ tmp: 26, creat_at: 2022-05-17T17:04:22.208Z }
tmp:{"tmp" : 10}
{ tmp: 10, creat_at: 2022-05-17T17:04:24.270Z }
tmp:{"tmp" : 13}
{ tmp: 13, creat_at: 2022-05-17T17:04:26.220Z }
tmp:{"tmp" : 26}
{ tmp: 26, creat_at: 2022-05-17T17:04:28.276Z }
tmp:{"tmp" : 8}
{ tmp: 8, creat_at: 2022-05-17T17:04:30.236Z }
tmp:{"tmp" : 22}
{ tmp: 22, creat_at: 2022-05-17T17:04:32.172Z }
tmp:{"tmp" : 19}
{ tmp: 19, creat_at: 2022-05-17T17:04:34.280Z }
tmp:{"tmp" : 29}
{ tmp: 29, creat_at: 2022-05-17T17:04:36.247Z }
tmp:{"tmp" : 26}

C:\Windows\System32\cmd.exe - mosquitto -v
1652807116: New connection from ::1:64711 on port 1883.
1652807116: New client connected from ::1:64711 as auto-17F3A9FE-4672-BB68-A908-BF991161D42E (p2, c1, k6
0).
1652807116: No will message specified.
1652807116: Sending CONNACK to auto-17F3A9FE-4672-BB68-A908-BF991161D42E (0, 0)
1652807116: Received PUBLISH from auto-17F3A9FE-4672-BB68-A908-BF991161D42E (d0, q0, r0, m0, 'tmp', ...
(12 bytes))
1652807116: Sending PUBLISH to mqttjs_dd689d25 (d0, q0, r0, m0, 'tmp', ... (12 bytes))
1652807116: Received DISCONNECT from auto-17F3A9FE-4672-BB68-A908-BF991161D42E
1652807116: Client auto-17F3A9FE-4672-BB68-A908-BF991161D42E disconnected.

C:\WINDOWS\system32\cmd.exe
C:\MQTT_practice\mosquitto>goto _loop
C:\MQTT_practice\mosquitto>set /a temp=17855 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{"tmp" : 6}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
C:\MQTT_practice\mosquitto>goto _loop
C:\MQTT_practice\mosquitto>set /a temp=4887 %30+1
C:\MQTT_practice\mosquitto>mosquitto_pub -t tmp -m "{"tmp" : 28}"
C:\MQTT_practice\mosquitto>timeout /t 2 /nobreak 1>nul
```