HALLYM UNIVERSITY

CIC Convergence Information & Communications

(IoT Network) Practice -9-

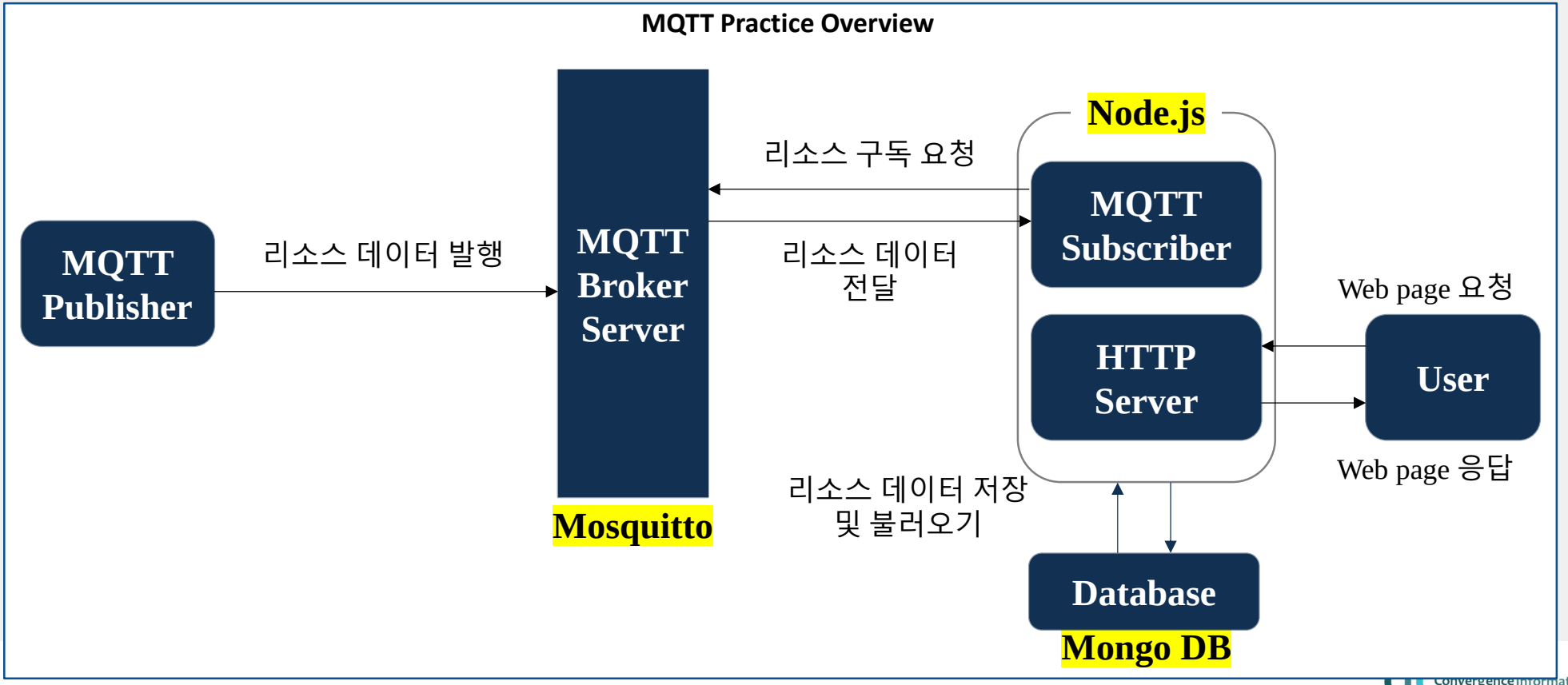
# MQTT client in Node.js

# Index



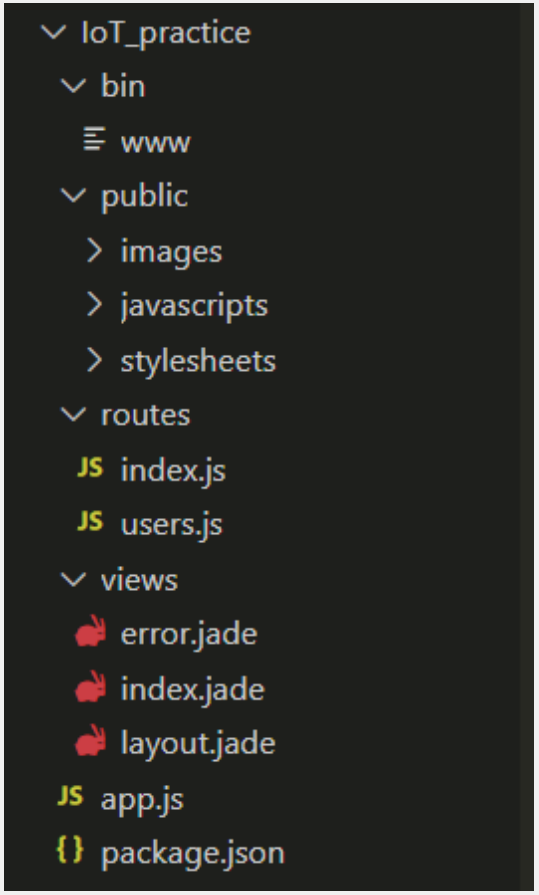
# Practice Overview

- **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**

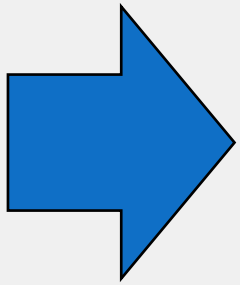



- **www**: start up script (최초로 Web 서버를 실행하기 위한 Script)
  1. HTTP 서버 생성 (이때, 서버의 Port 번호 지정)
  2. Middleware(혹은 node module)와 HTTP 서버 연동
  3. 그 외 필요한 Web 서버의 기능 초기화
- **public**: 그림, 스타일, 스크립트와 같은 모든 정적 파일 포함
- **routes**: 요청에 대한 응답 처리 (웹 페이지 랜더링)
- **views**: 웹페이지 템플릿
- **app.js**: express 객체를 생성 및 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

```json
{
  "name": "iot-practice",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "jade": "~1.11.0",
    "morgan": "~1.9.1"
  }
}
```



```json
{
  "name": "iot-practice",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "jade": "~1.11.0",
    "morgan": "~1.9.1",
    "mqtt" : "^4.3.7",
    "mongodb" : "^4.6.0",
    "socket.io" : "^4.5.0"
  }
}
```

# 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

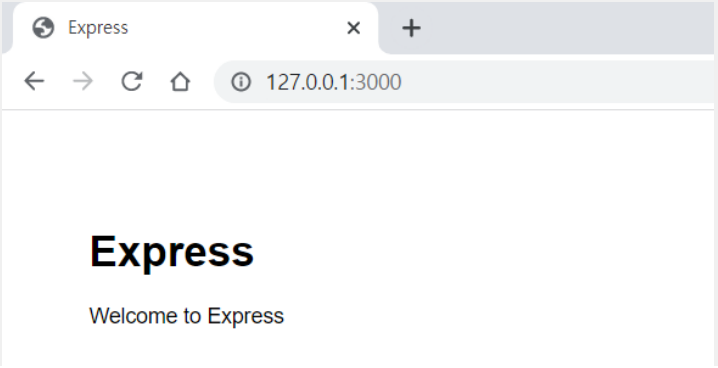
```
PS C:\MQTT_practice\IoT_practice> npm install
npm WARN deprecated constantinople@3.0.2: Please update to at least constantinople 3.1.1
npm WARN deprecated transformers@2.1.0: Deprecated, use jstransformer
npm WARN deprecated jade@1.11.0: Jade has been renamed to pug, please install the latest version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

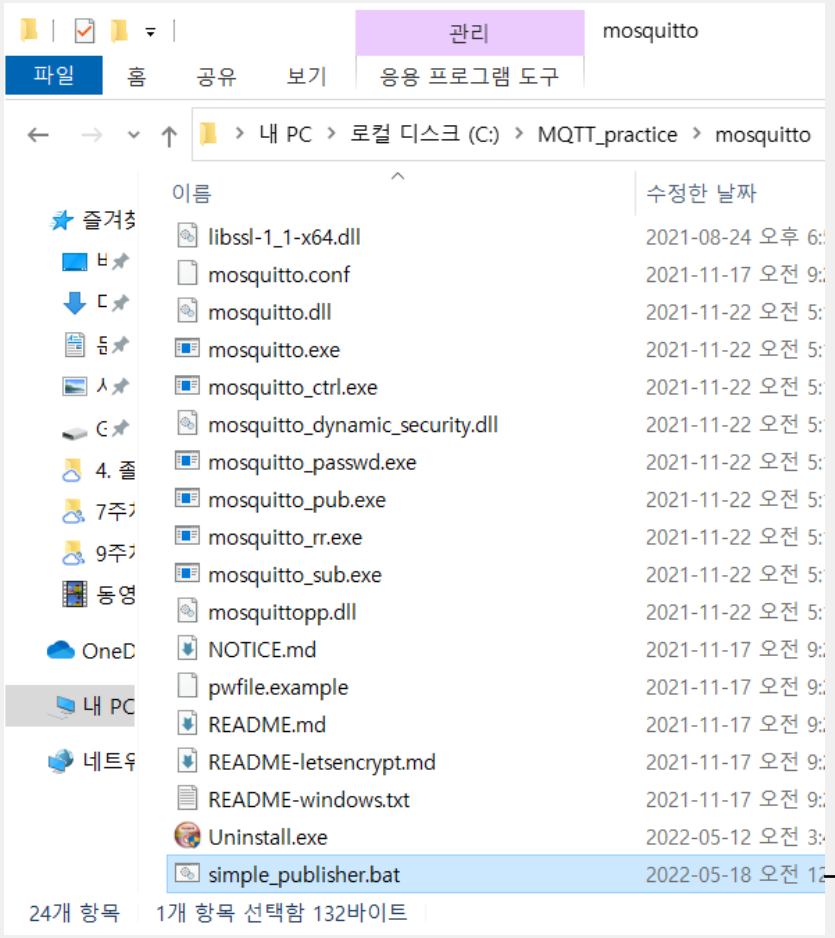
```
PS C:\MQTT_practice\IoT_practice> npm start

> iot-practice@0.0.0 start
> node ./bin/www
```

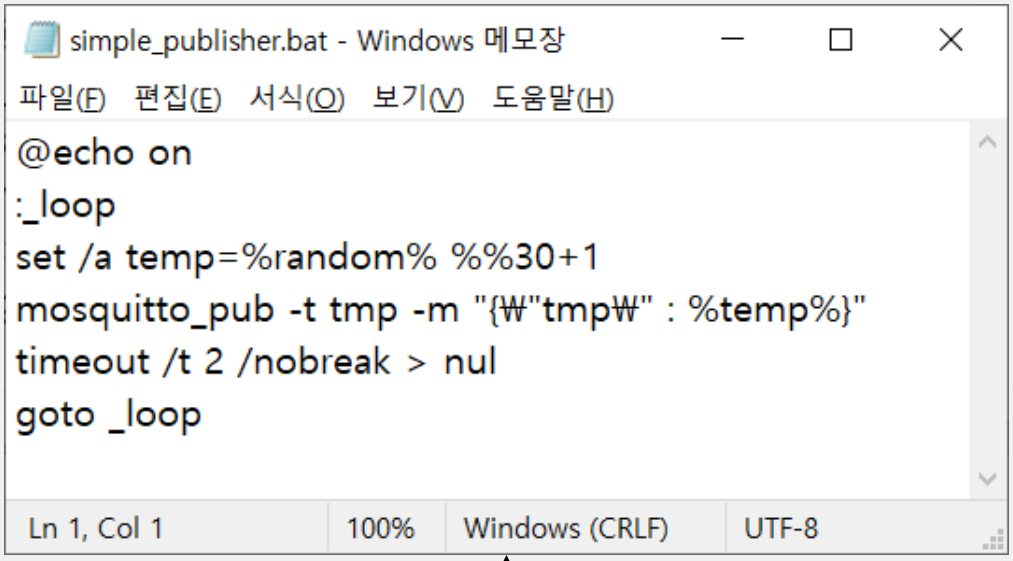
  - IoT_practice 프로젝트 실행 후 웹 브라우저로 서버 동작 확인
    - ✓ URL: 127.0.0.1:3000 혹은 localhost:3000



# MQTT Publisher using Mosquitto

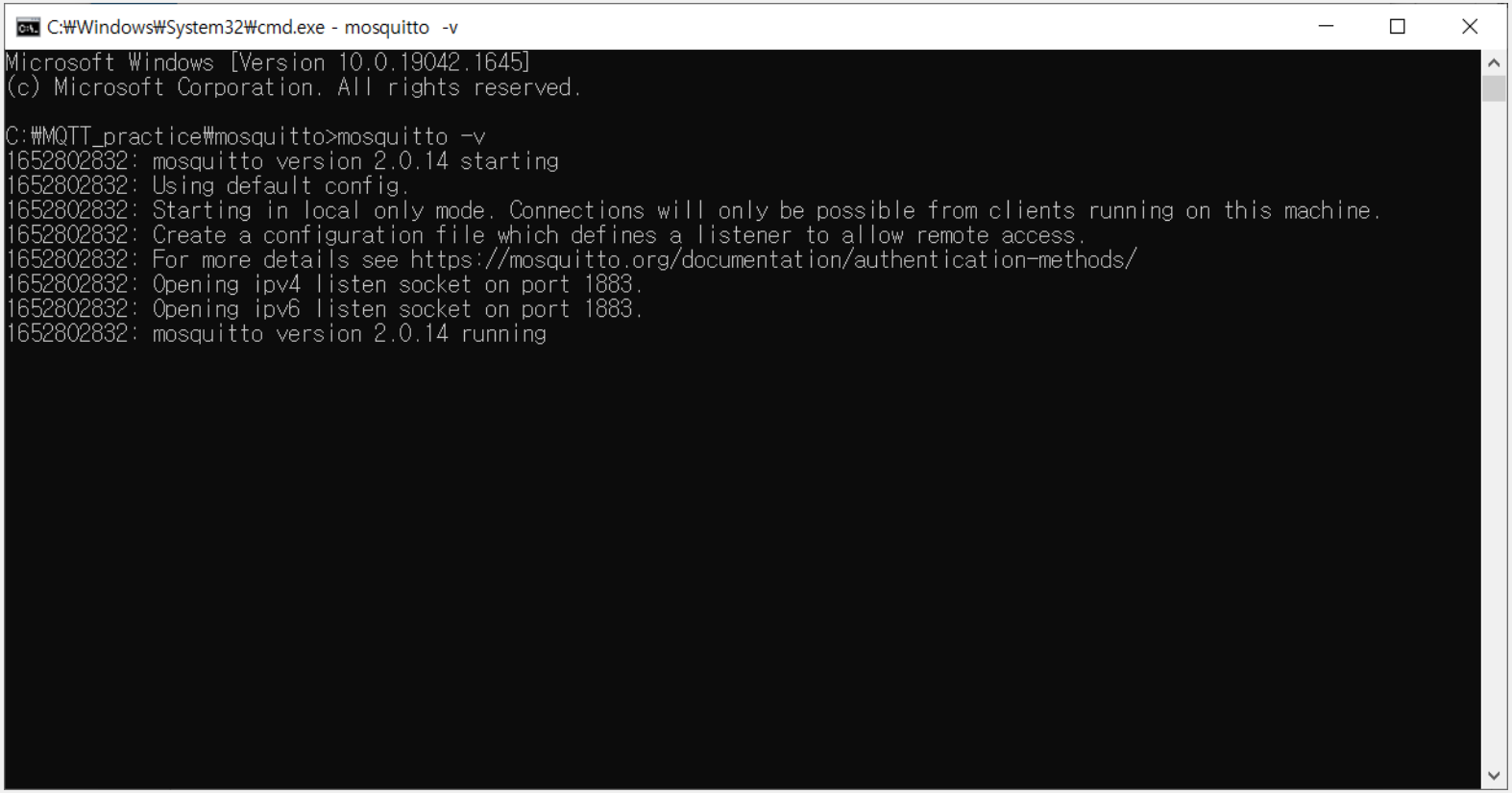
- **Mosquitto를 사용한 MQTT Publisher**
  - Mosquitto가 설치된 폴더에 다음과 같이 bat 파일 저장



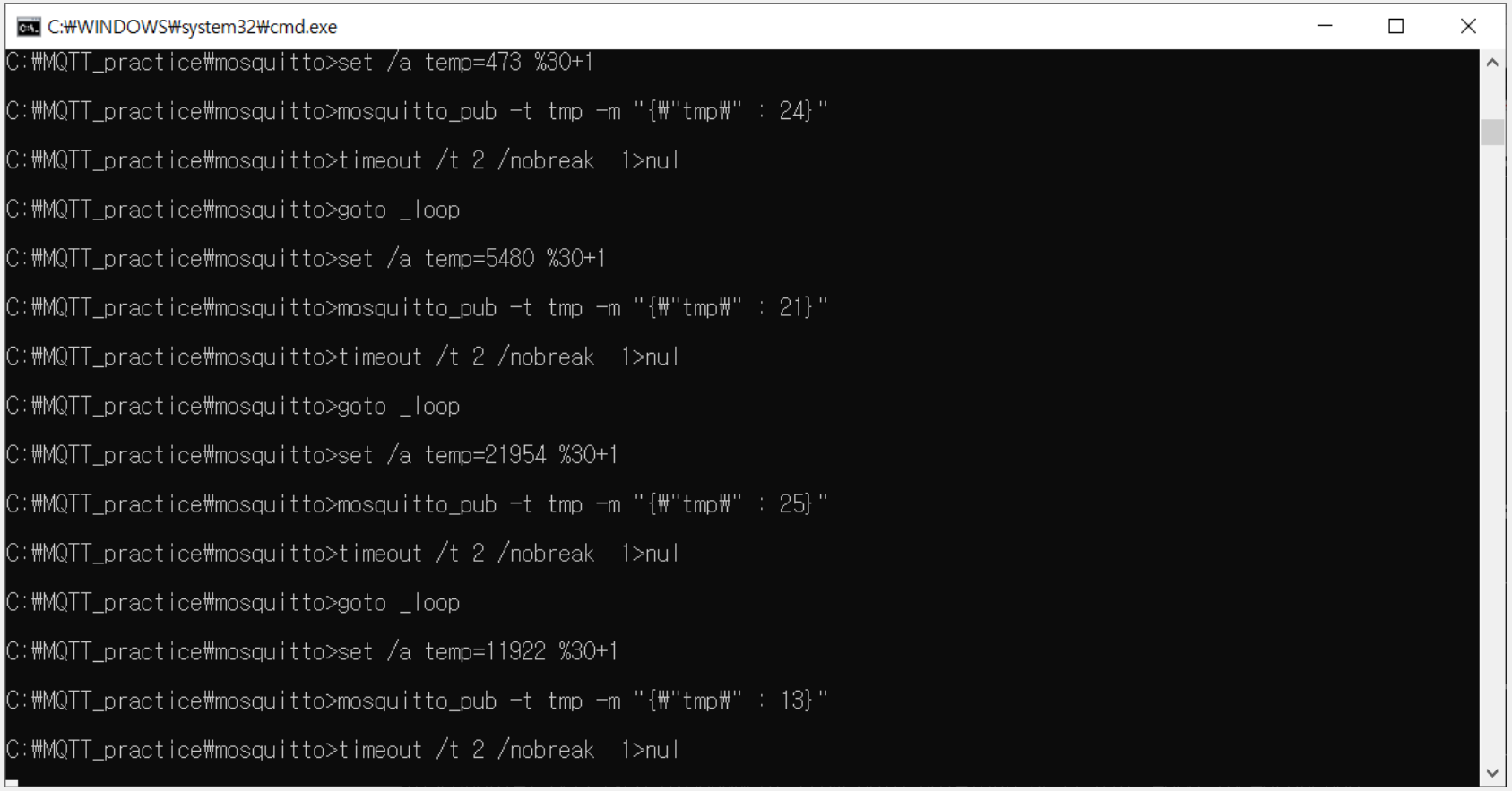
# MQTT Publisher using Mosquitto

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



# MQTT Publisher using Mosquitto

- **Mosquitto를 사용한 MQTT Publisher**
  - MQTT publisher 실행



# 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