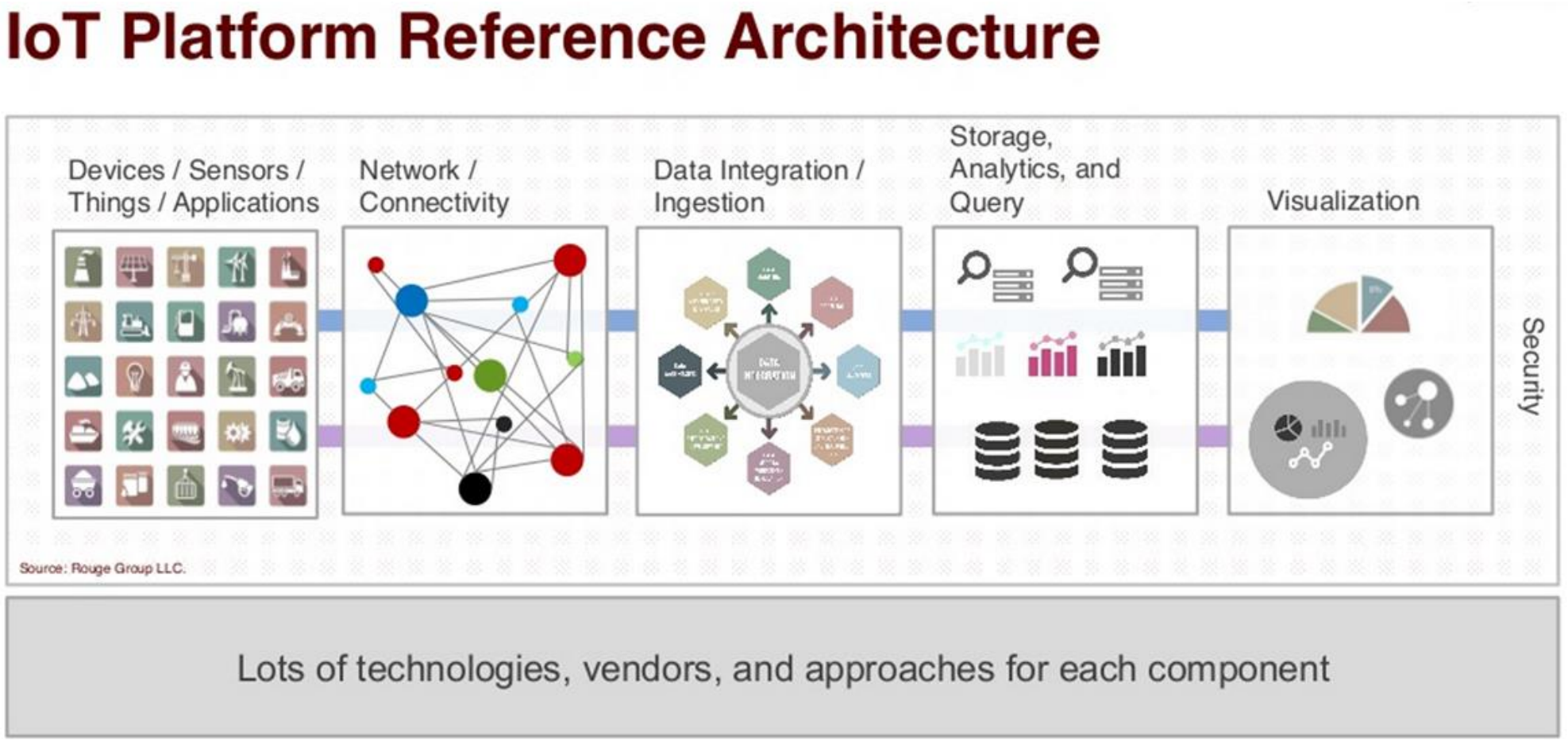
IoT 클라우드 플랫폼(ThinkSpeak)을 이용한 실시간 온습도 모니터링

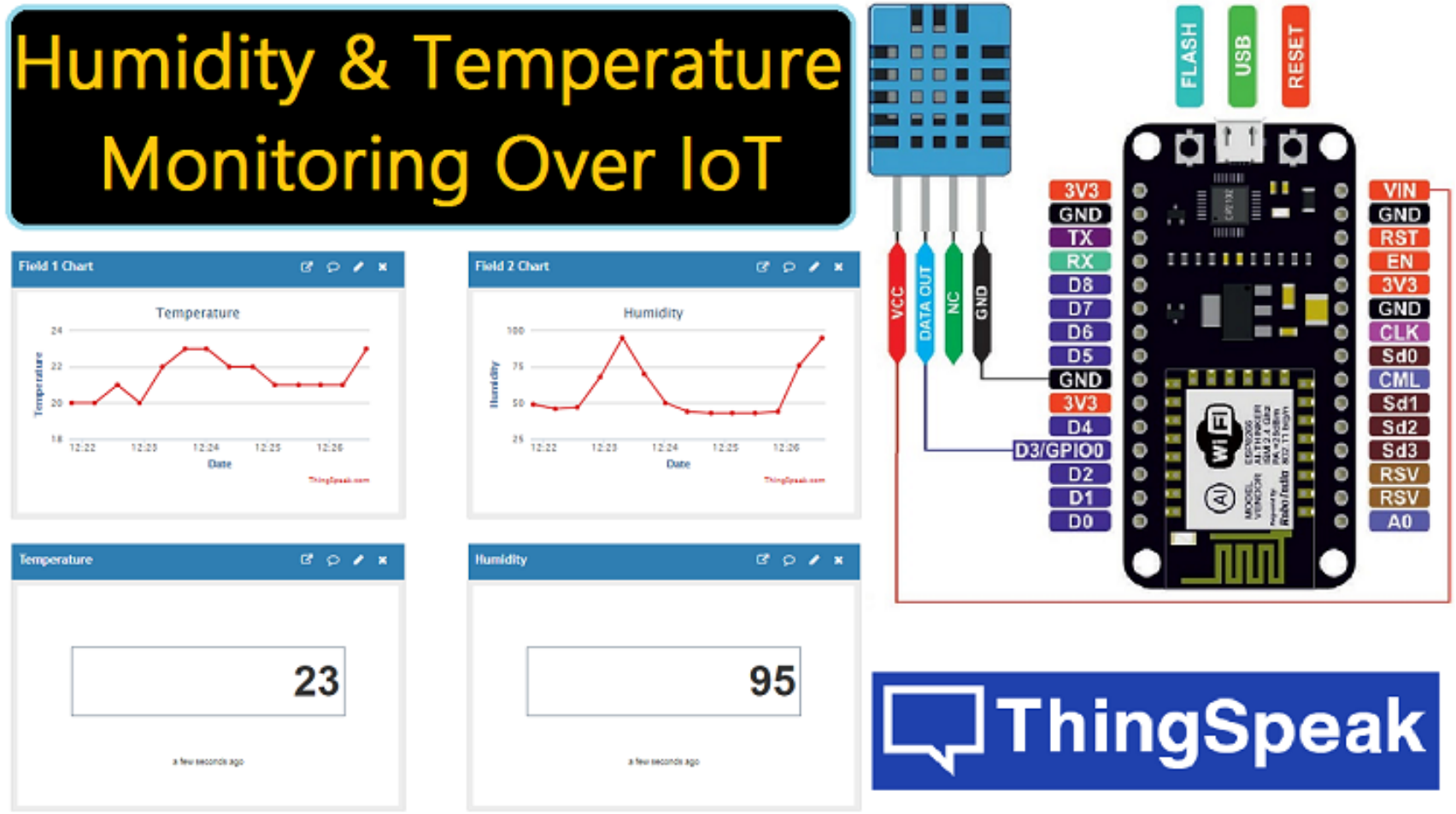
Week08

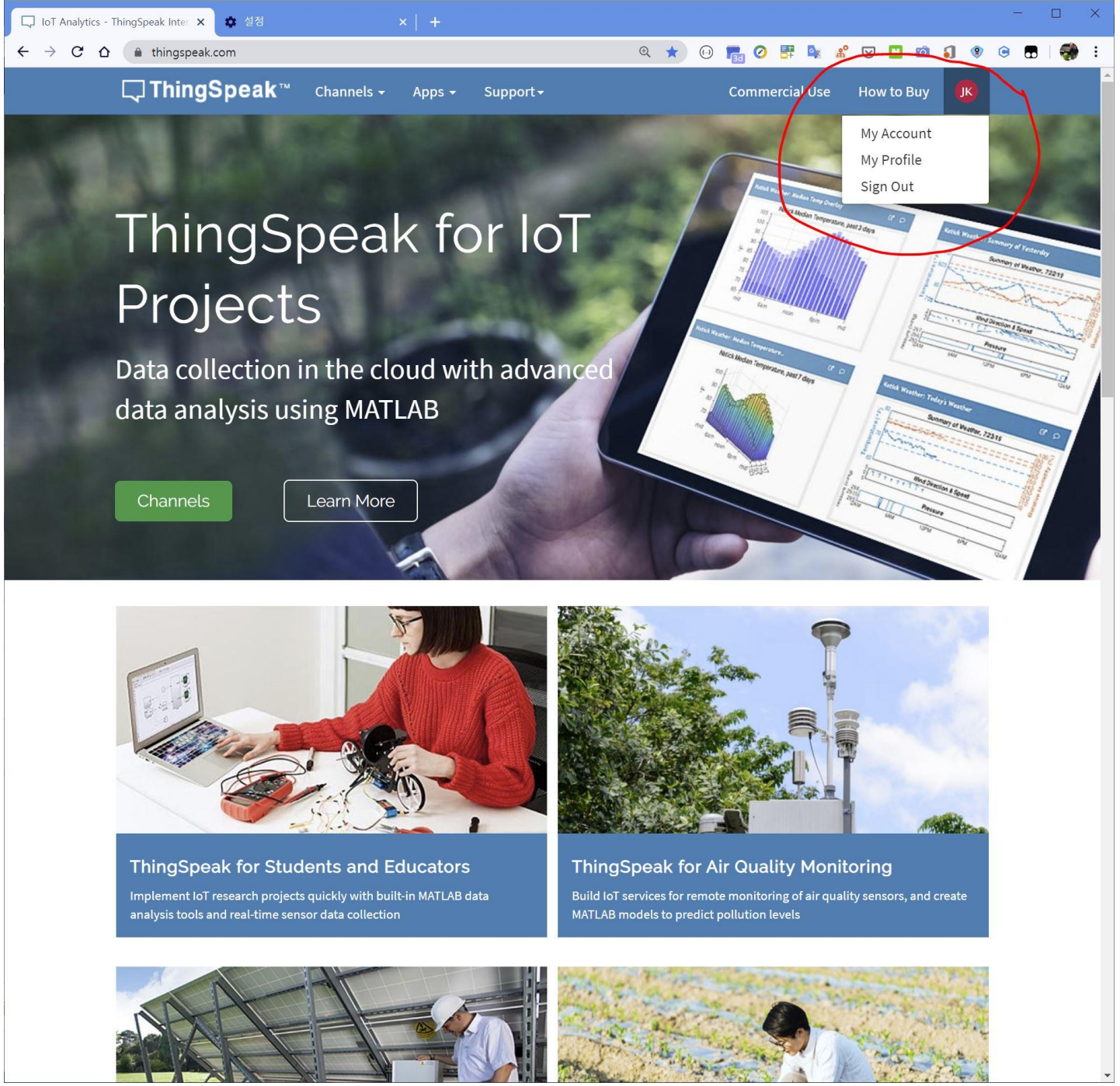
IoT Cloud Platform Landscape

- https://www.postscapes.com/internet-of-things-platforms/

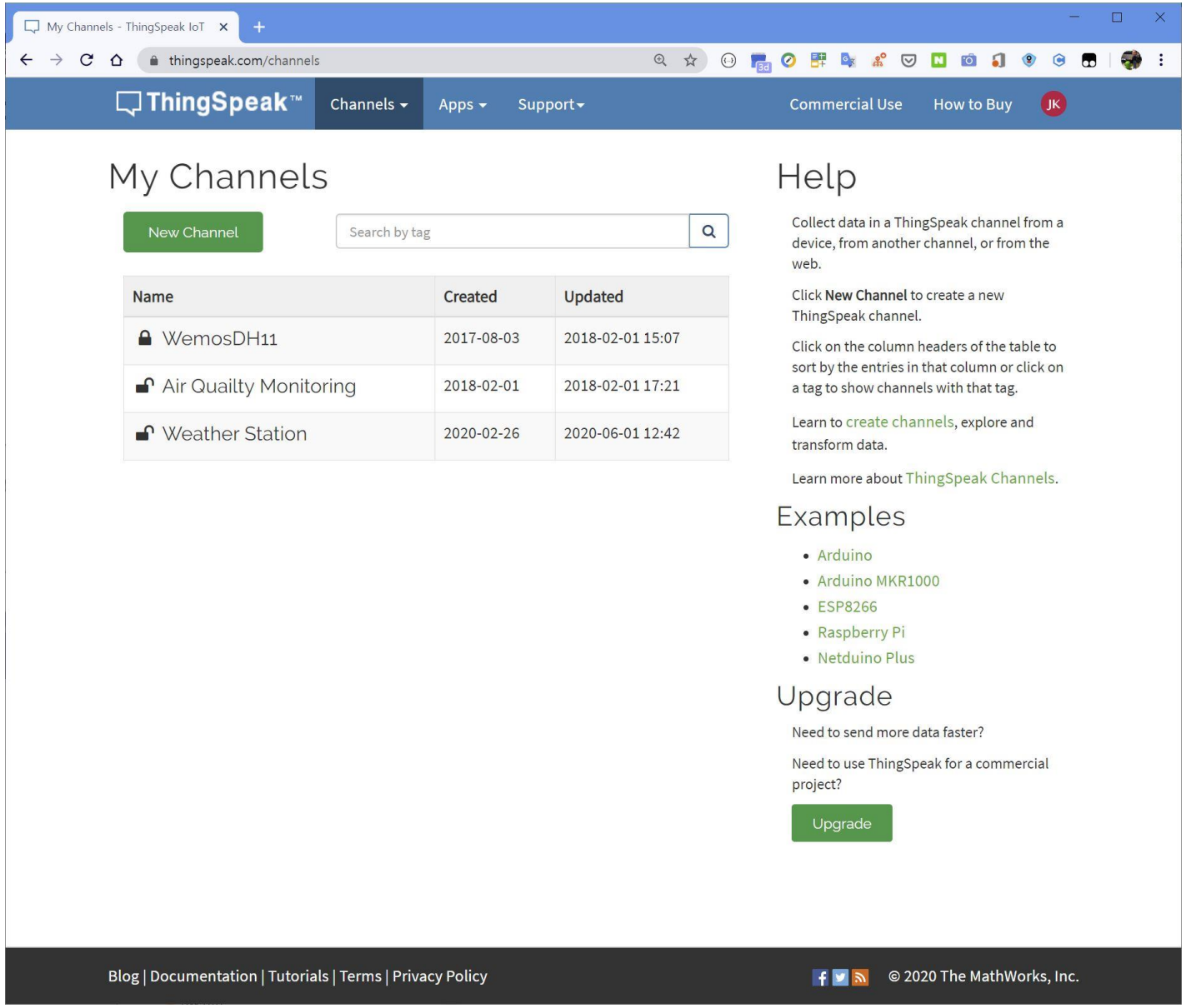


ThinkSpeak : https://thingspeak.com/

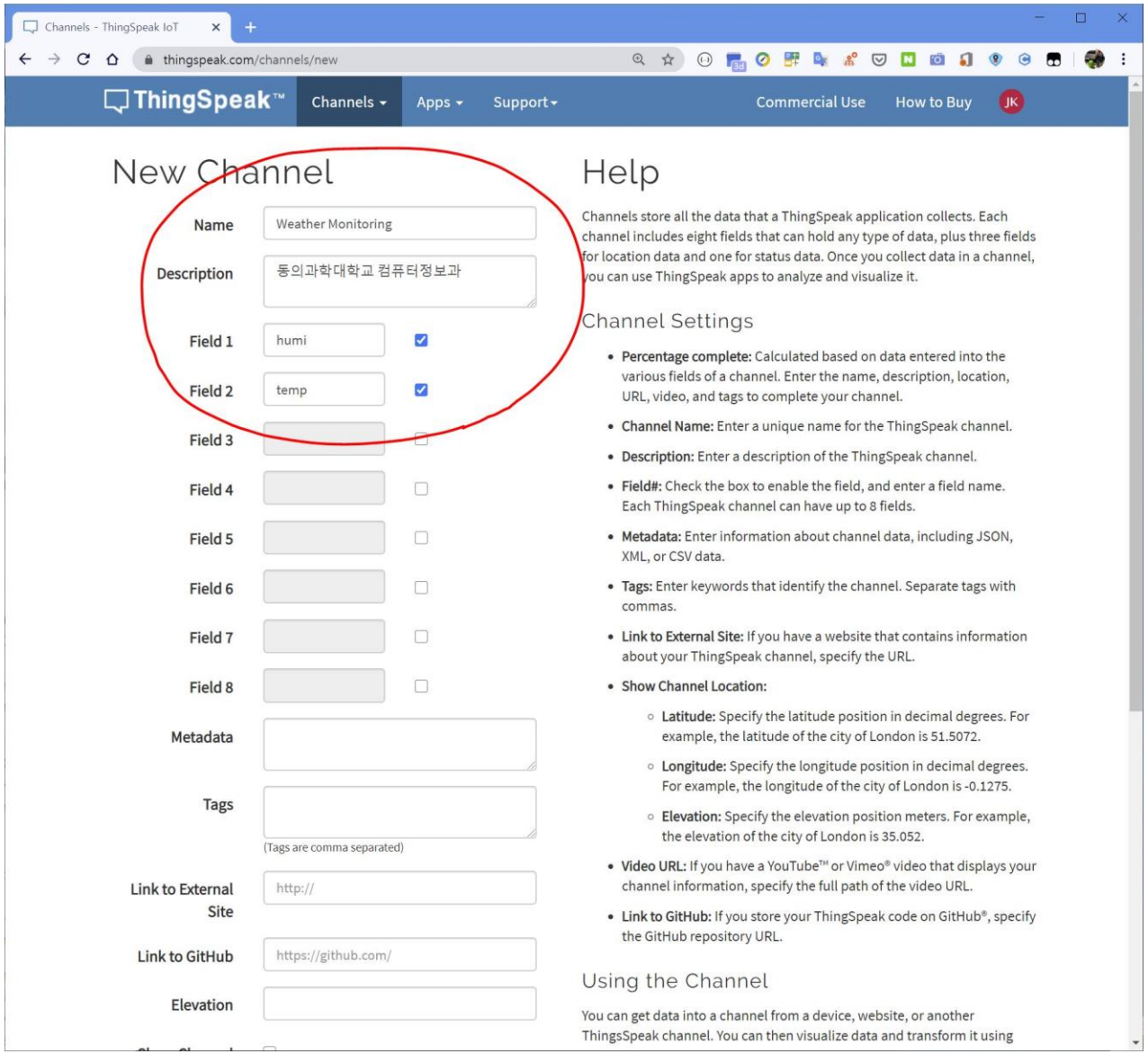



IoT Analytics - ThingSpeak Inter
설정
thingspeak.com
ThingSpeak™
Channels
Apps
Support
Commercial Use
How to Buy
JK
My Account
My Profile
Sign Out
ThingSpeak for IoT Projects
Data collection in the cloud with advanced data analysis using MATLAB
Channels
Learn More
ThingSpeak for Students and Educators
Implement IoT research projects quickly with built-in MATLAB data analysis tools and real-time sensor data collection
ThingSpeak for Air Quality Monitoring
Build IoT services for remote monitoring of air quality sensors, and create MATLAB models to predict pollution levels

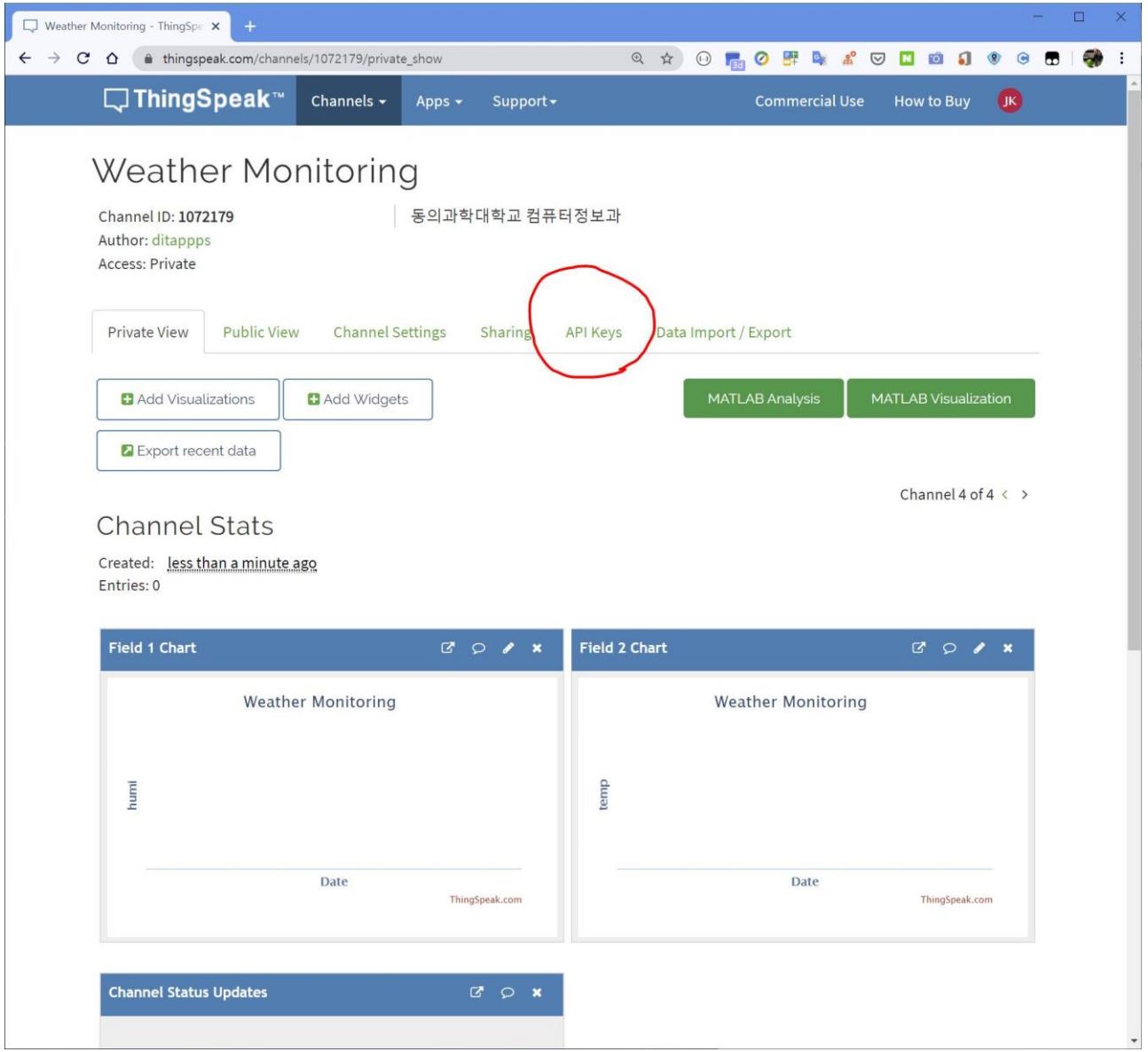
My Channels -> New Channel



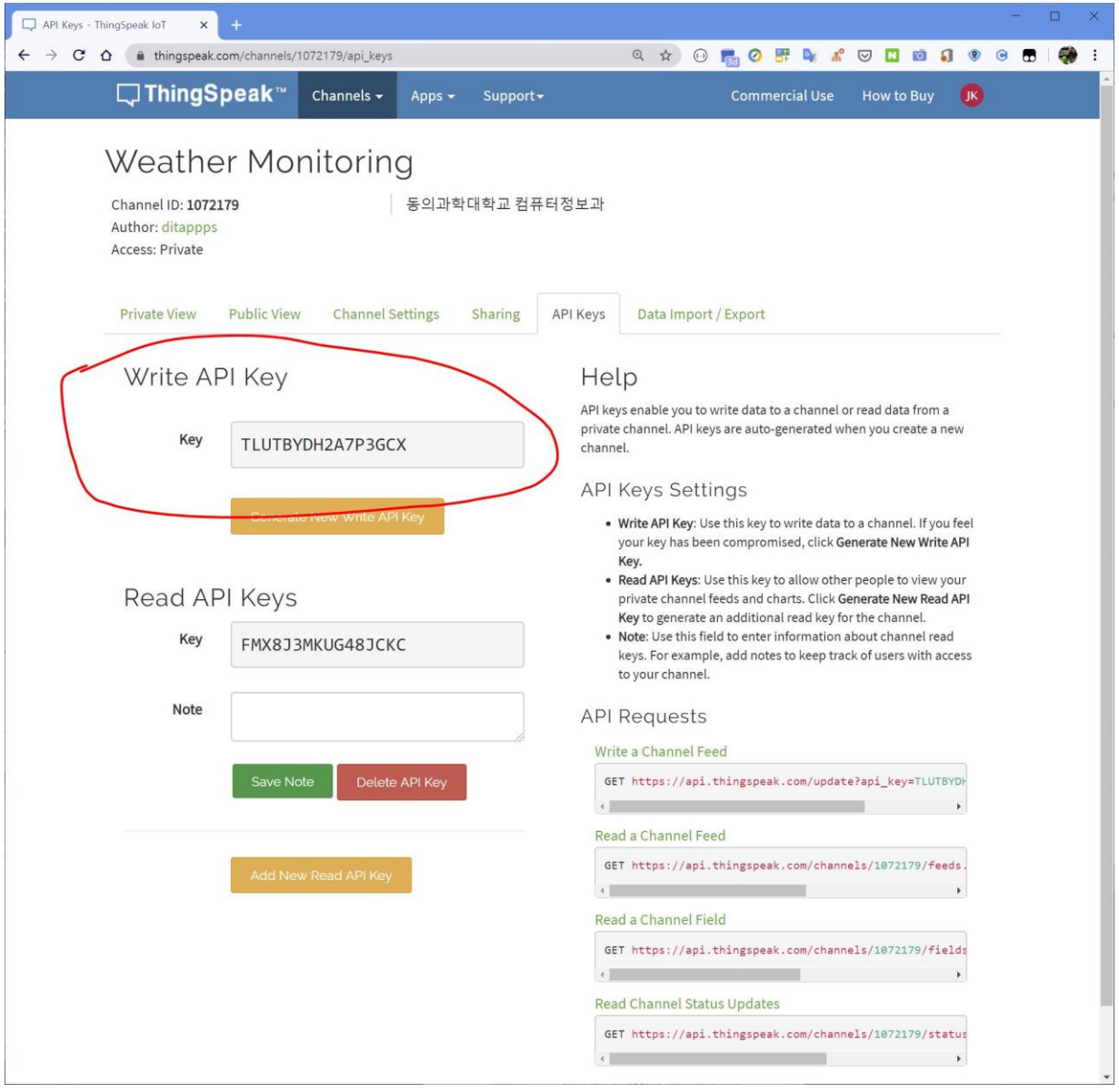
New Channels(Weather Monitoring) -> Save



Weather Monitoring 대시보드



API Keys : Write API Keys



필요한 부품들

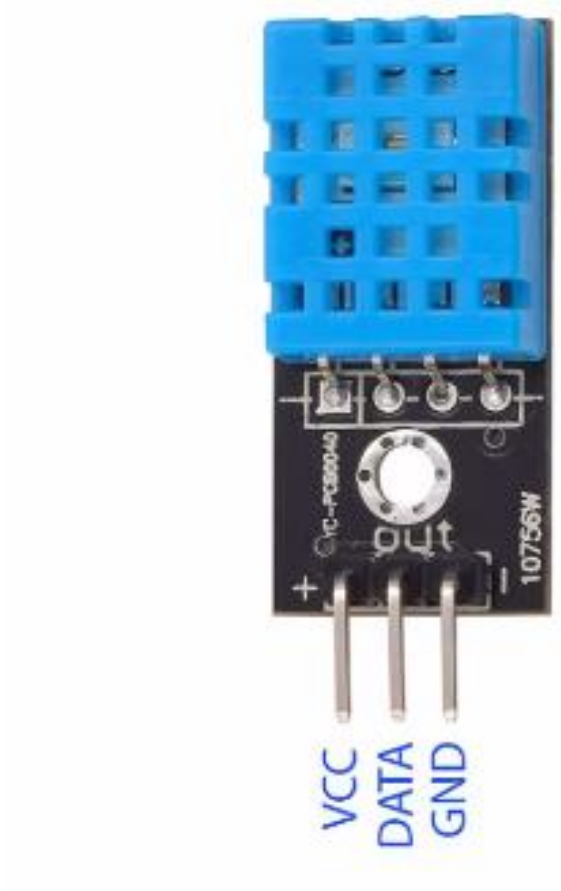
NodeMCU

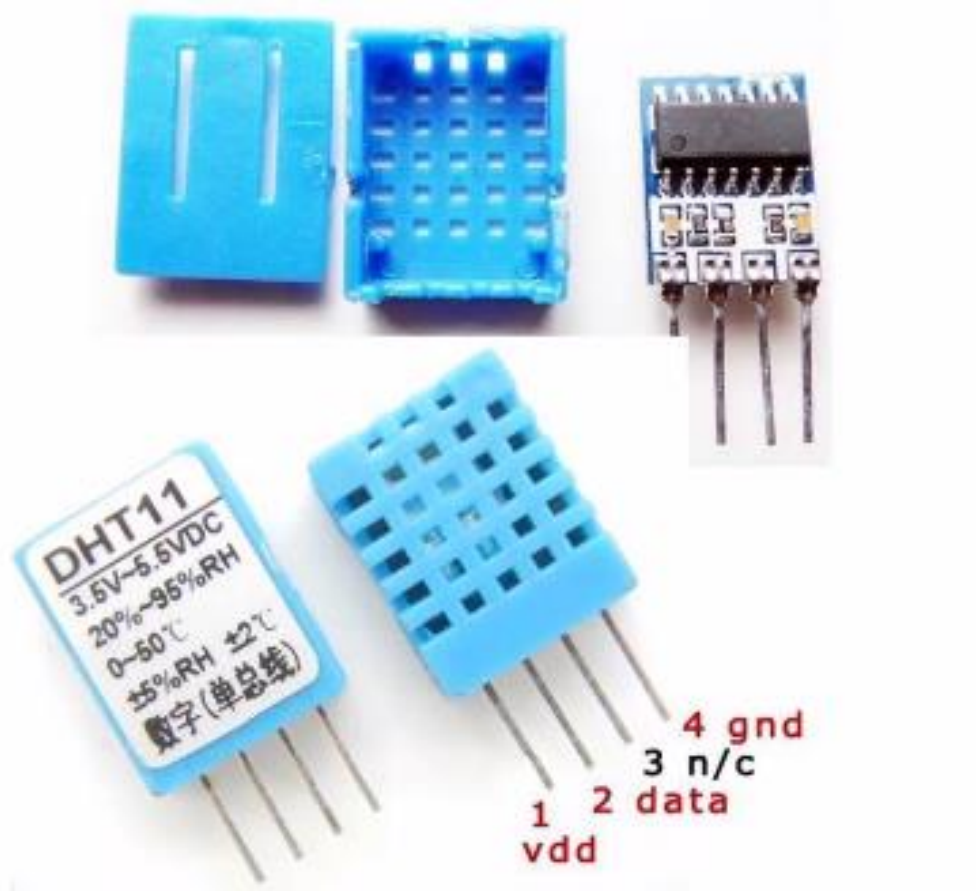
DHT11

Breadboard

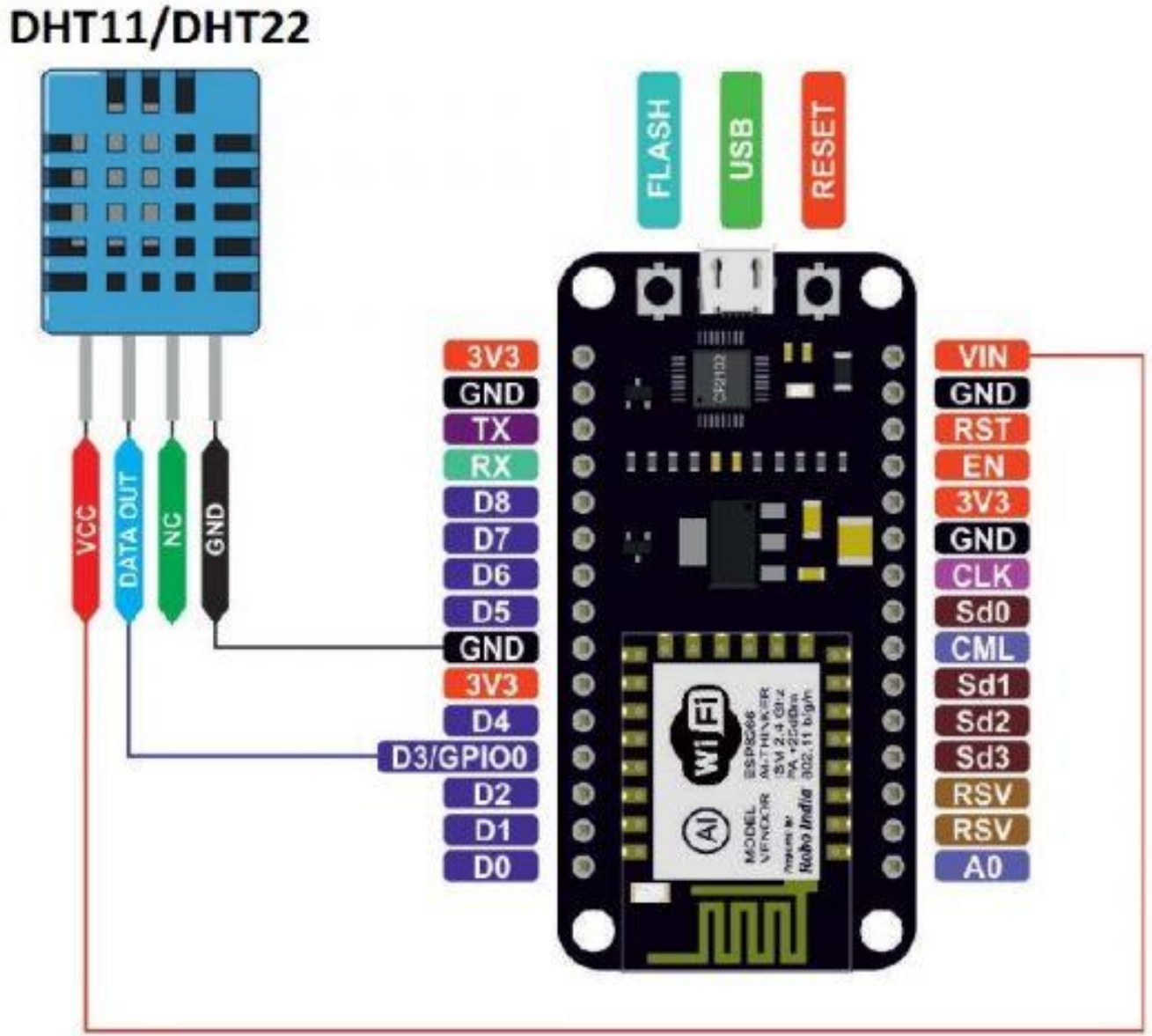
Jumper Wires

DHT11 온습도 센서


out
VCC
DATA
GND


DHT11
3.5V~5.5VDC
20%~95%RH
0~50℃
±5%RH ±2℃
4 gnd
3 n/c
2 data
1
vdd

회로 연결


DHT11/DHT22
VCC
DATA OUT
NC
GND
FLASH
USB
RESET
3V3
GND
TX
RX
D8
D7
D6
D5
GND
3V3
D4
D3/GPIO0
D2
D1
D0
VIN
GND
RST
EN
3V3
GND
CLK
Sd0
CML
Sd1
Sd2
Sd3
RSV
RSV
A0

이두이노 소스 코드 : setup()

DHT11_ThinkSpeak_NodeMCU_Weather_station §

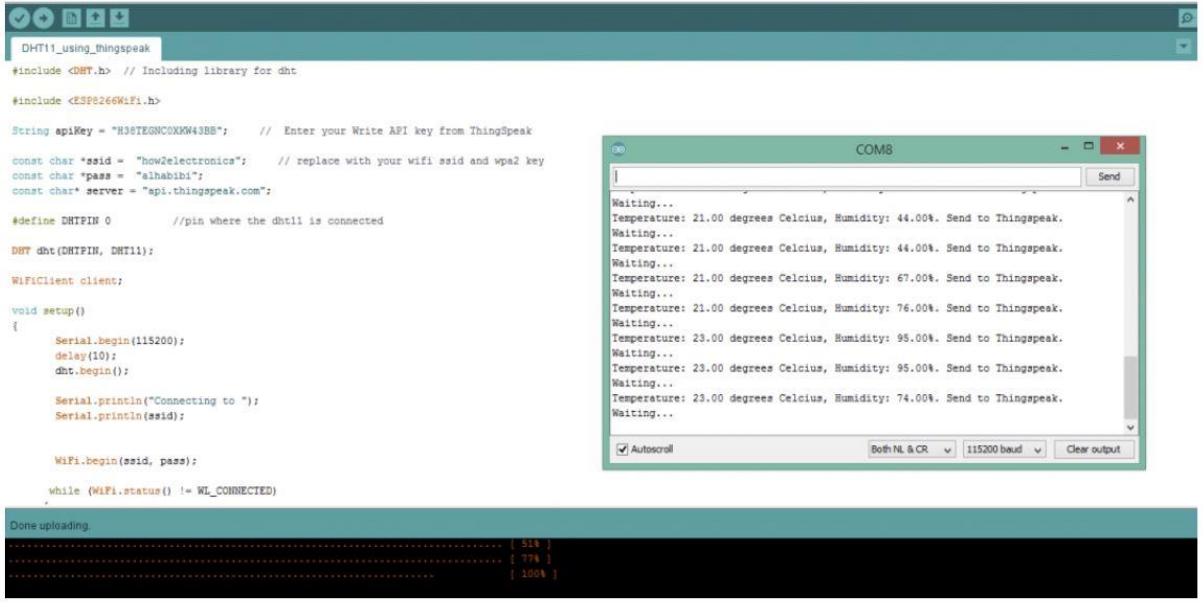
```
#include <DHT.h>
#include <ESP8266WiFi.h>
#define DHTPIN  D4  // D0
#define DHTTYPE DHT11

const char* ssid = "melon";
const char* password = "deitcs3217";

const char* server = "api.thingspeak.com";
String apiKey = "VJF12D7B55H1ONG1"; // Write API Key

WiFiClient client;
DHT dht(DHTPIN, DHTTYPE, D4);

void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.println("ThingSpeak");
  // DHT11 초기화
  dht.begin();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  // Wi-Fi 최기화
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected!!");
}
```

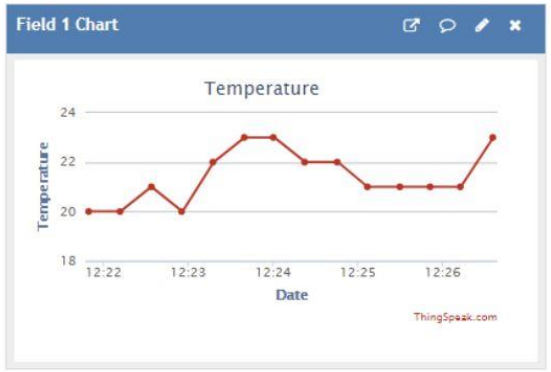
아두이노 소스 코드 : loop()

DHT11_ThinkSpeak_NodeMCU_Weather_station §

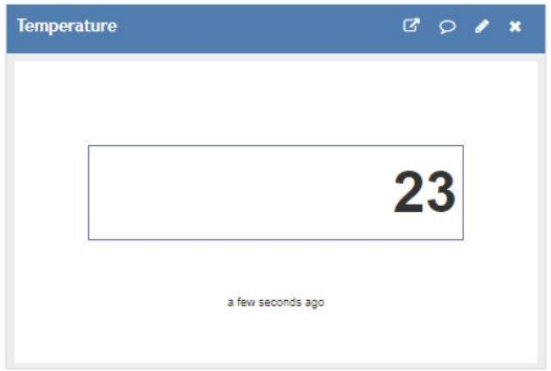
```
34
35 void loop() {
36   float t, h;
37   t = dht.readTemperature();
38   h = dht.readHumidity();
39
40   if (isnan(h) || isnan(t)) {
41     Serial.println("Failed to read!!");
42     return;
43   }
44
45   if (client.connect(server,80)) {// "184.106.153.149" or api.thingspeak.com
46     String postStr = apiKey;
47            postStr +="&field1=";
48            postStr += String(t);
49            postStr +="&field2=";
50            postStr += String(h);
51            postStr += "\r\n\r\n";
52
53      // Rest API POST 요청 : 웹서바에 resource 생성 요청
54      client.print("POST /update HTTP/1.1\n");
55      client.print("Host: api.thingspeak.com\n");
56
57      // 온도 데이터를 한번 전송하고 client와 연결을 끊고 15초 시간 지연 대기 한다.
58      client.print("Connection: close\n");
59      client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
60      client.print("Content-Type: application/x-www-form-urlencoded\n");
61      client.print("Content-Length: ");
62      client.print(postStr.length());
63      client.print("\n\n");
64      client.print(postStr);
65
66      Serial.print("Temp : ");
67      Serial.print(t);
68      Serial.print("\t Humidity: ");
69      Serial.print(h);
70      Serial.println("%. Send to Thingspeak.");
71   }
72   client.stop();
73
74   Serial.println("Waiting...");
75   // ThingSpeak 무료 계정은 업데이트 주기가 15초임
76   delay(15000);
77 }
```

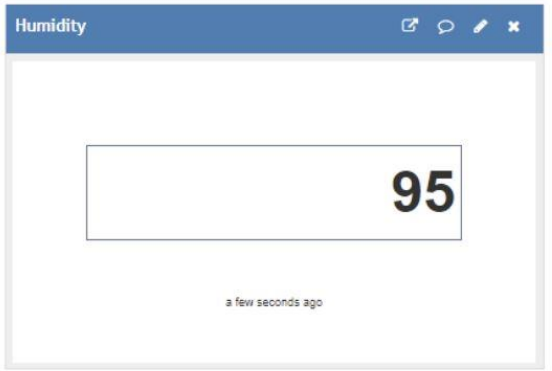
실행 결과


DHT11_using_thingspeak
#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>
String apiKey = "H38TEGNC0XKW43BB"; // Enter your Write API key from ThingSpeak
const char *ssid = "how2electronics"; // replace with your wifi ssid and wpa2 key
const char *pass = "alhabibi";
const char* server = "api.thingspeak.com";
#define DHTPIN 0 //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
Serial.begin(115200);
delay(10);
dht.begin();
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
Done uploading.
COM8
Send
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 44.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 44.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 67.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 76.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 95.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 95.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 74.00%. Send to Thingspeak.
Waiting...
Autoscroll
Both NL & CR
115200 baud
Clear output

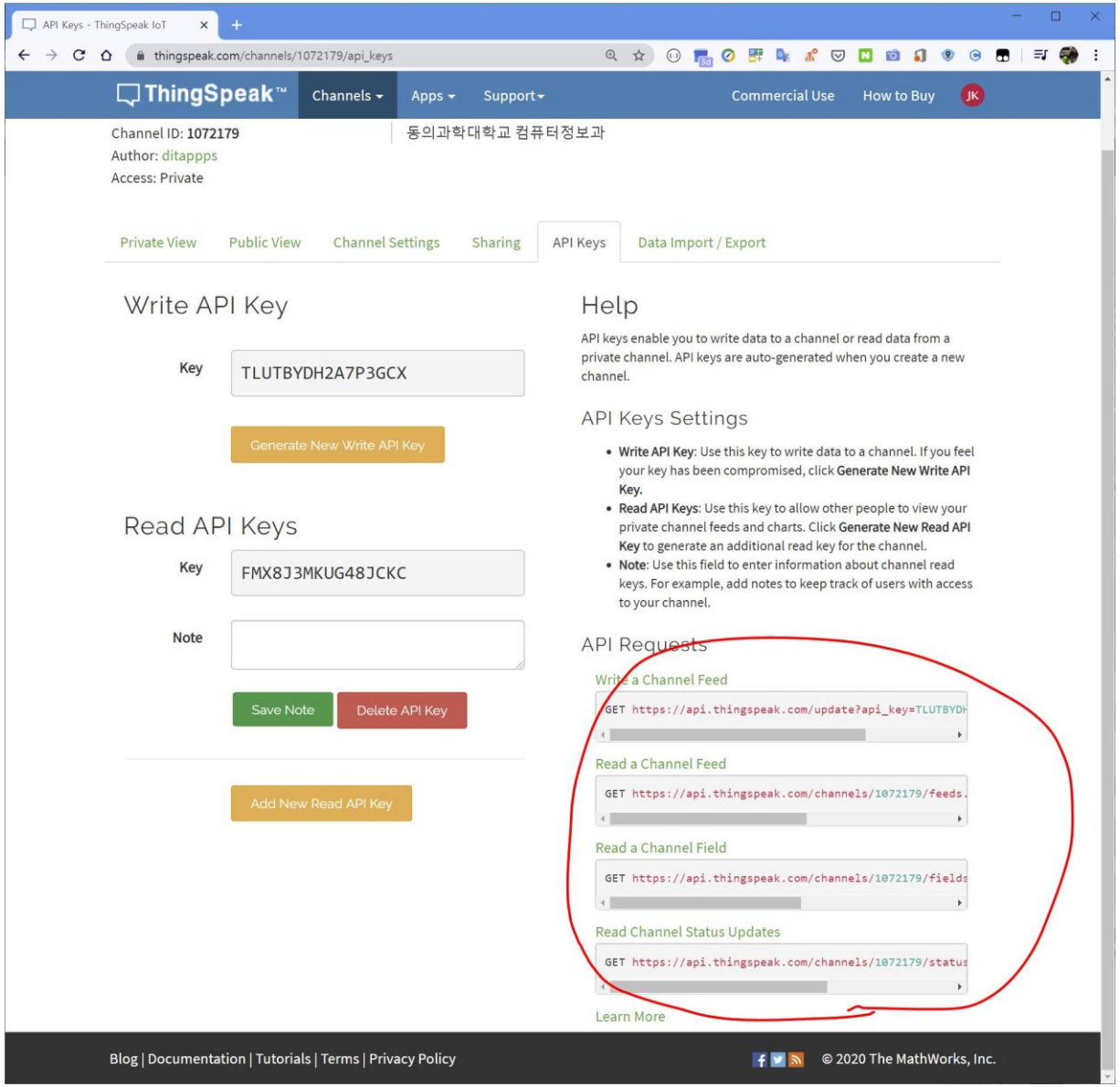

Field 1 Chart
Temperature
24
22
20
18
Temperature
12:22
12:23
12:24
12:25
12:26
Date
ThingSpeak.com


Field 2 Chart
Humidity
100
75
50
25
Humidity
12:22
12:23
12:24
12:25
12:26
Date
ThingSpeak.com

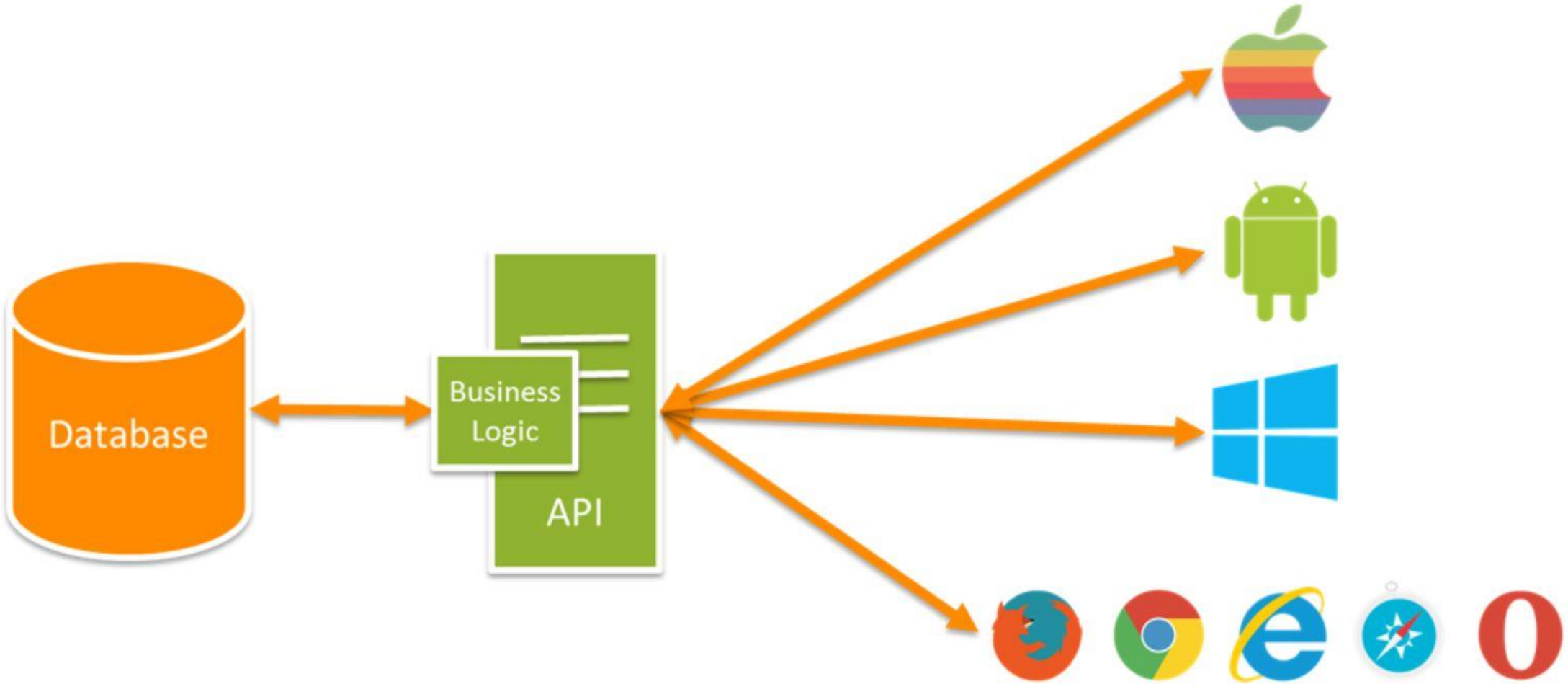

Temperature
23
a few seconds ago


Humidity
95
a few seconds ago

REST API 사용하기

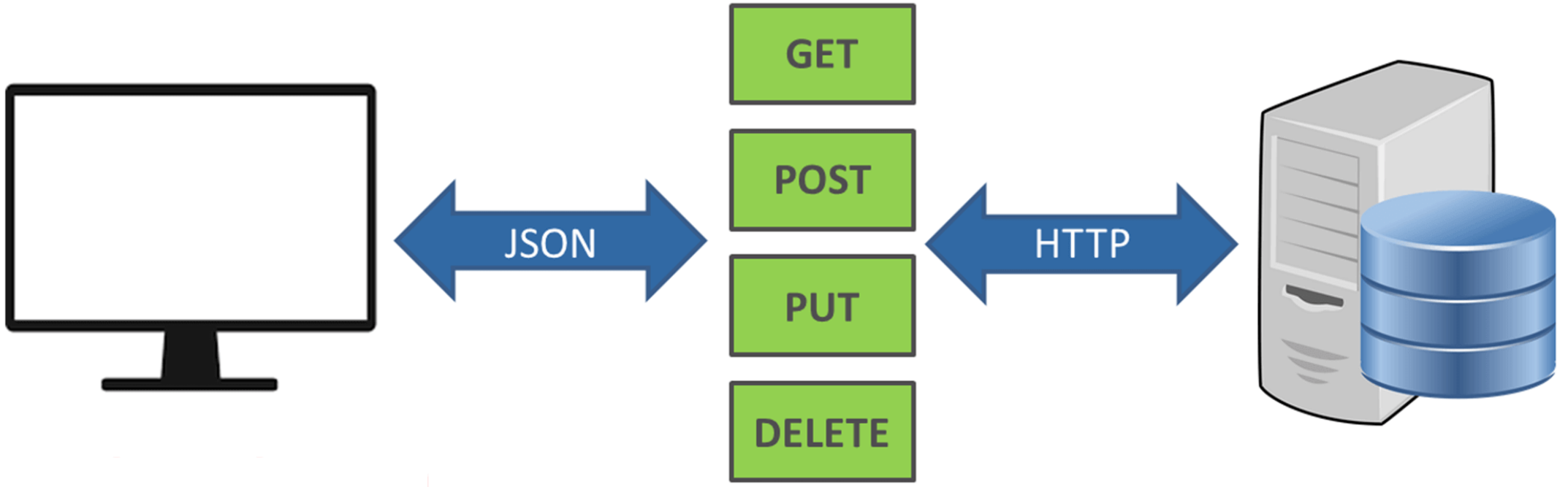


API : Application Programming Interface



REST API란?

- REST
  - HTTP 기반으로 필요한 지원에 접근 하는 방식을 정해 놓은 구조
  - REST 규칙에 따라 설계된 API를 REST API 라고 한다.



REST API 란?

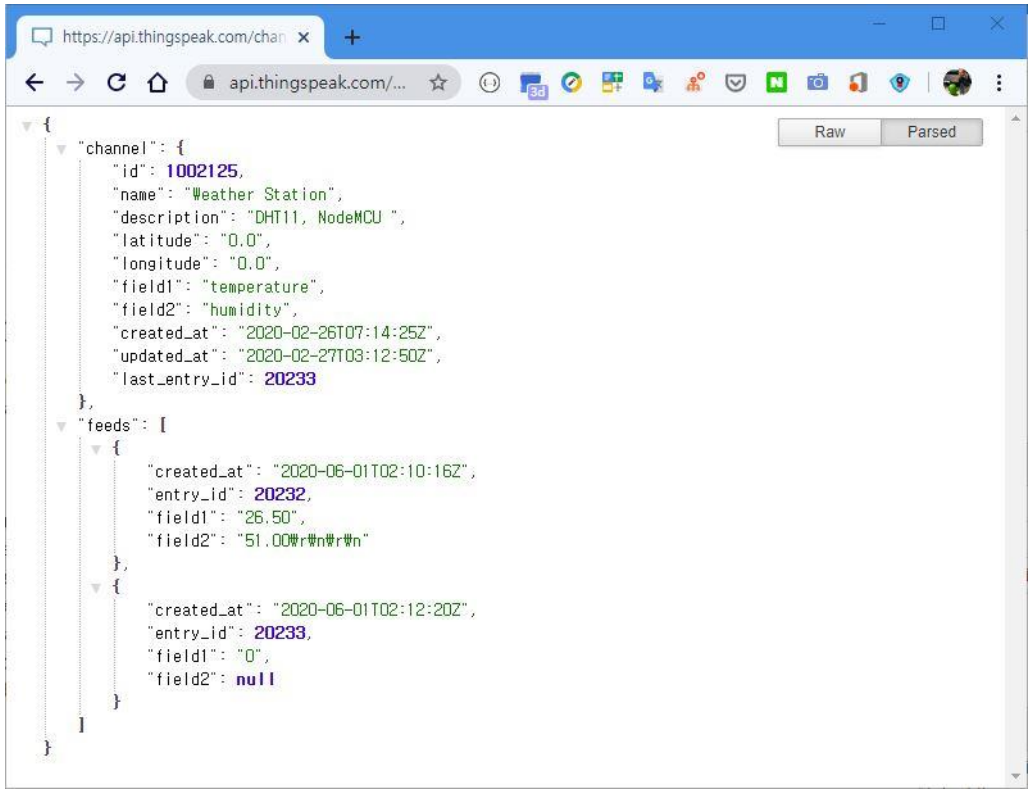
CRUD	HTTP verbs	Route
resource들의 목록을 표시	GET	/resource
resource 하난의 내용을 표시	GET	/resource/:id
resource를 생성	POST	/resource
resource를 수정	PUT	/resource/:id
resource를 삭제	DELETE	/resource/:id

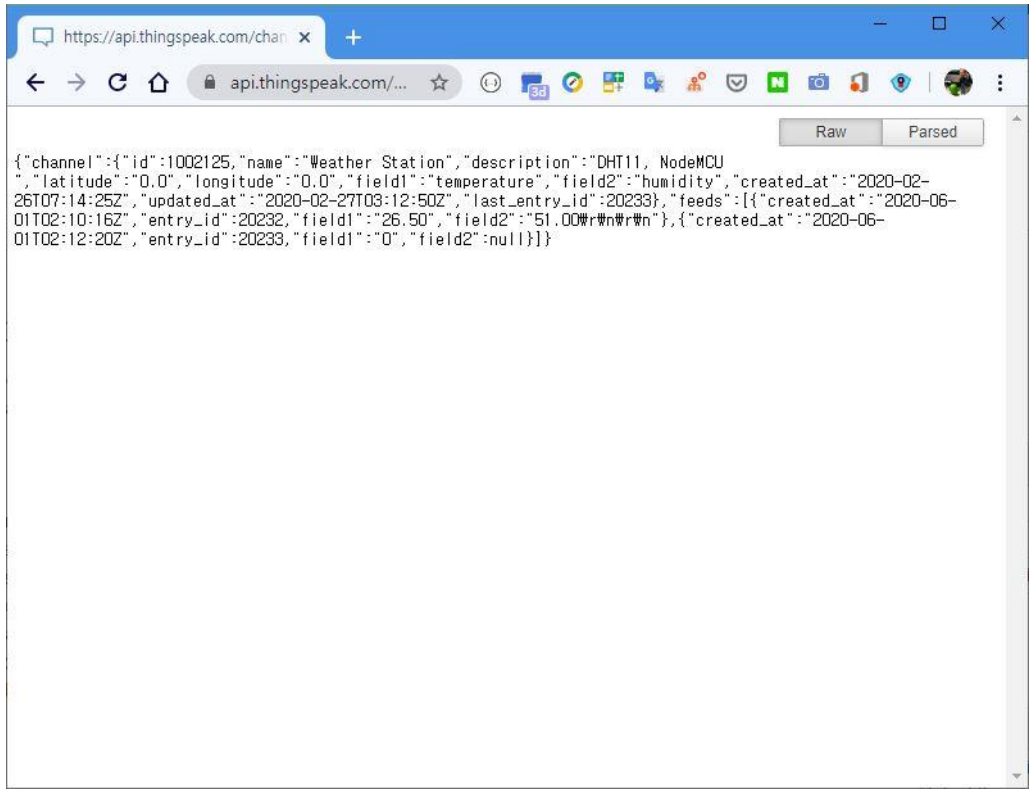
“REST”

GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

API Requests : Read a Channel Feed

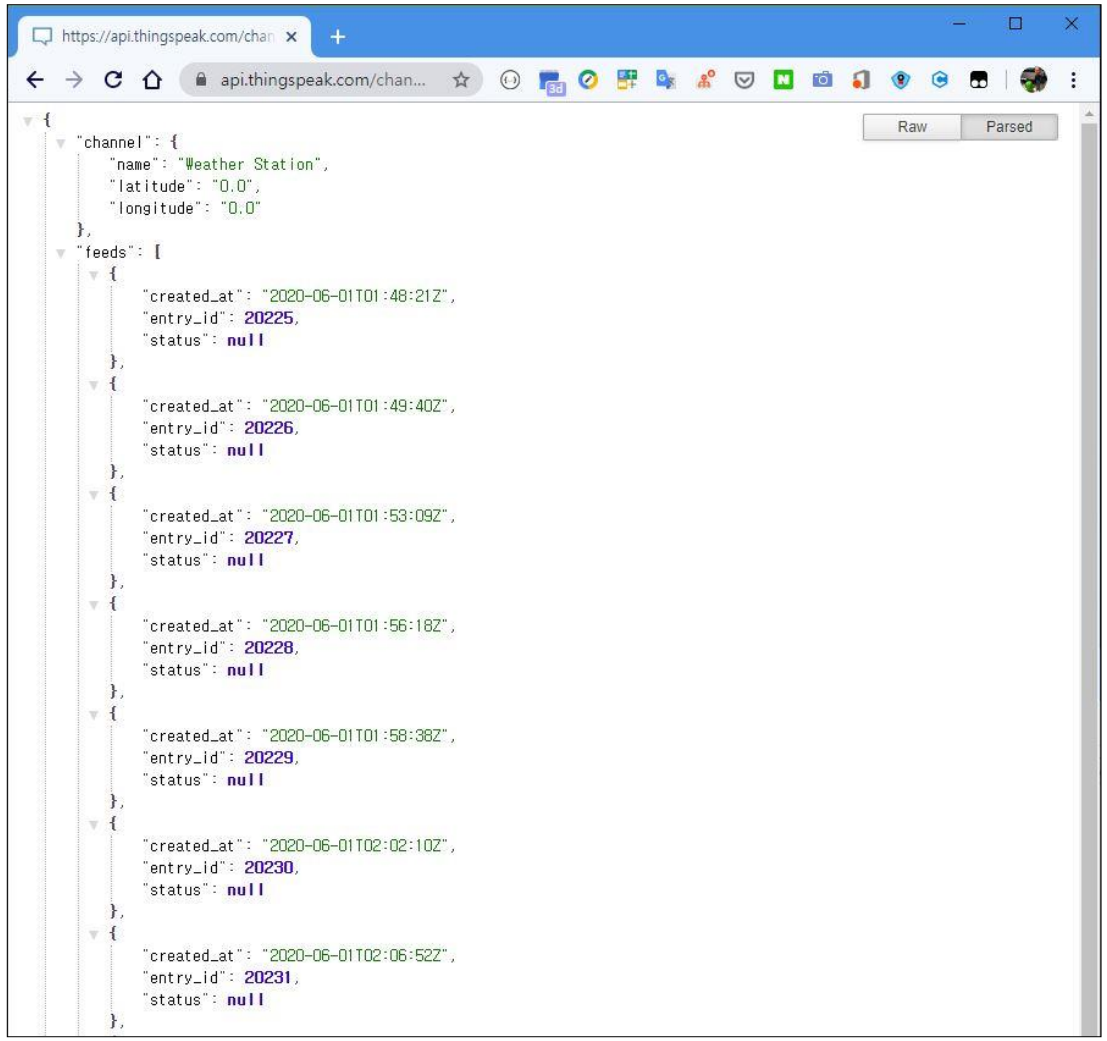
- GET https://api.thingspeak.com/channels/1072179/fields/1.json?api_key=FMX8J3MKUG48JCKC&results=2





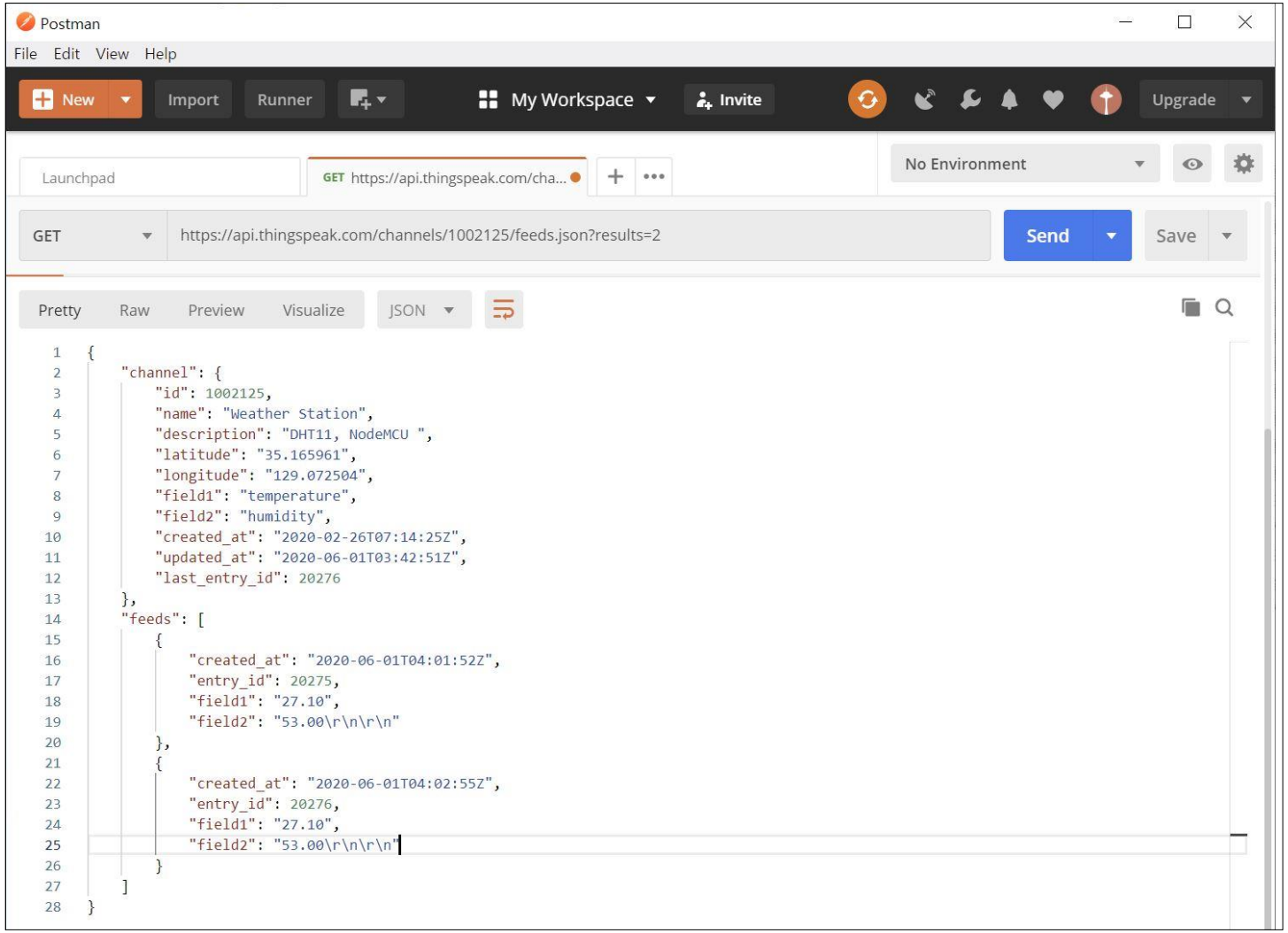
API Requests : Read a Status Update

- GET https://api.thingspeak.com/channels/1072179/status.json?api_key=FMX8J3MKUG48JCKC



Postman : https://www.postman.com/

- 개발한 API를 테스트하고, 테스트 결과를 공유하여 API 개발의 생산성을 높여주는 플랫폼



실습 과제

- 수업한 온습도 모니터링은 ThinkSpeak에서 15초 간격으로 NodeMCU에서 센서 데이터를 측정하여 서버에 보내준다.
- LED 1개를 추가하여 온습도 센서가 ThinkSpeak 클라우드에 정상적으로 보낸 직후에 LED를 한번 ON/OF 하도록 구성하시오.
- 데이터가 정상적으로 보내지면 LED는 15초 간격으로 ON/OFF 하고, 그렇지 않으면 LED는 작동하지 않은다.
- 과제 제출 기한 : 다음주 월요일 까지(6/8), jhkim3217@gmail.com

참고

- YouTube Tutorial : https://youtu.be/jYjuxWUefhg