

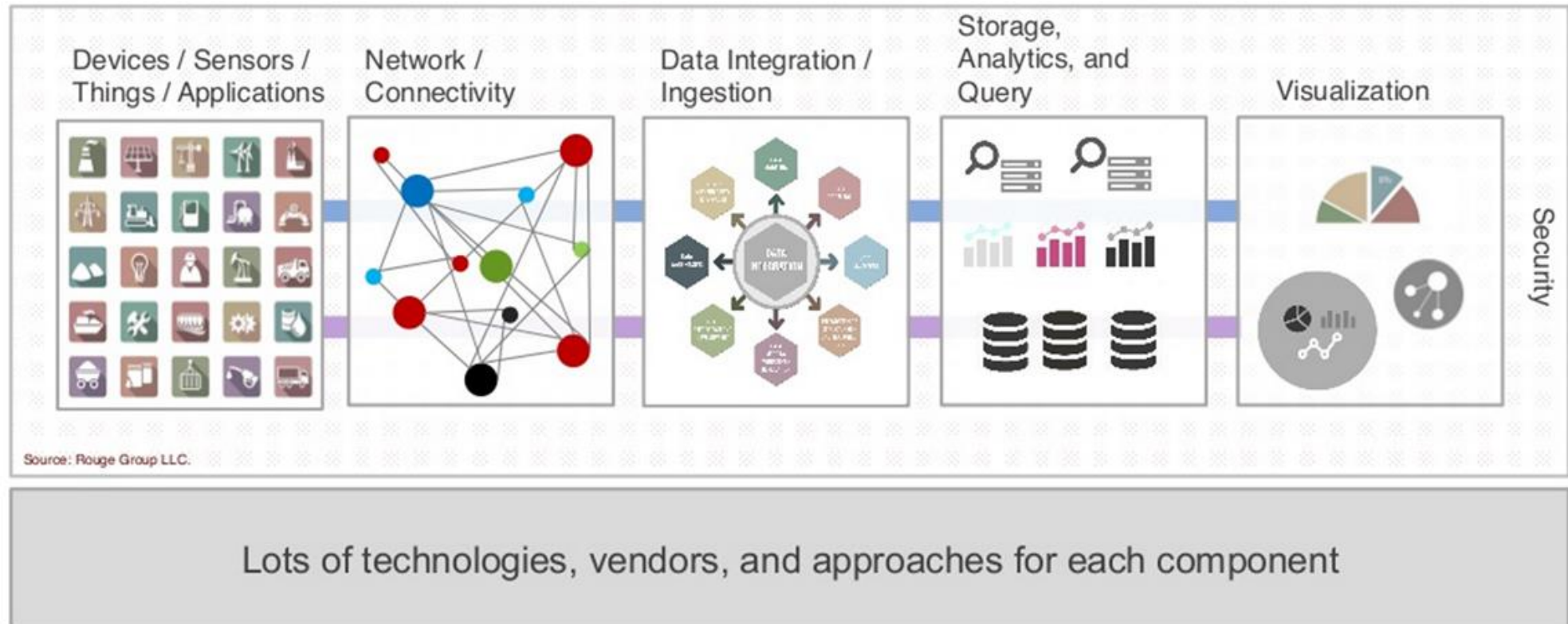
IoT 클라우드 플랫폼(ThinkSpeak)을 이용한 실시간 온습도 모니터링

Week08

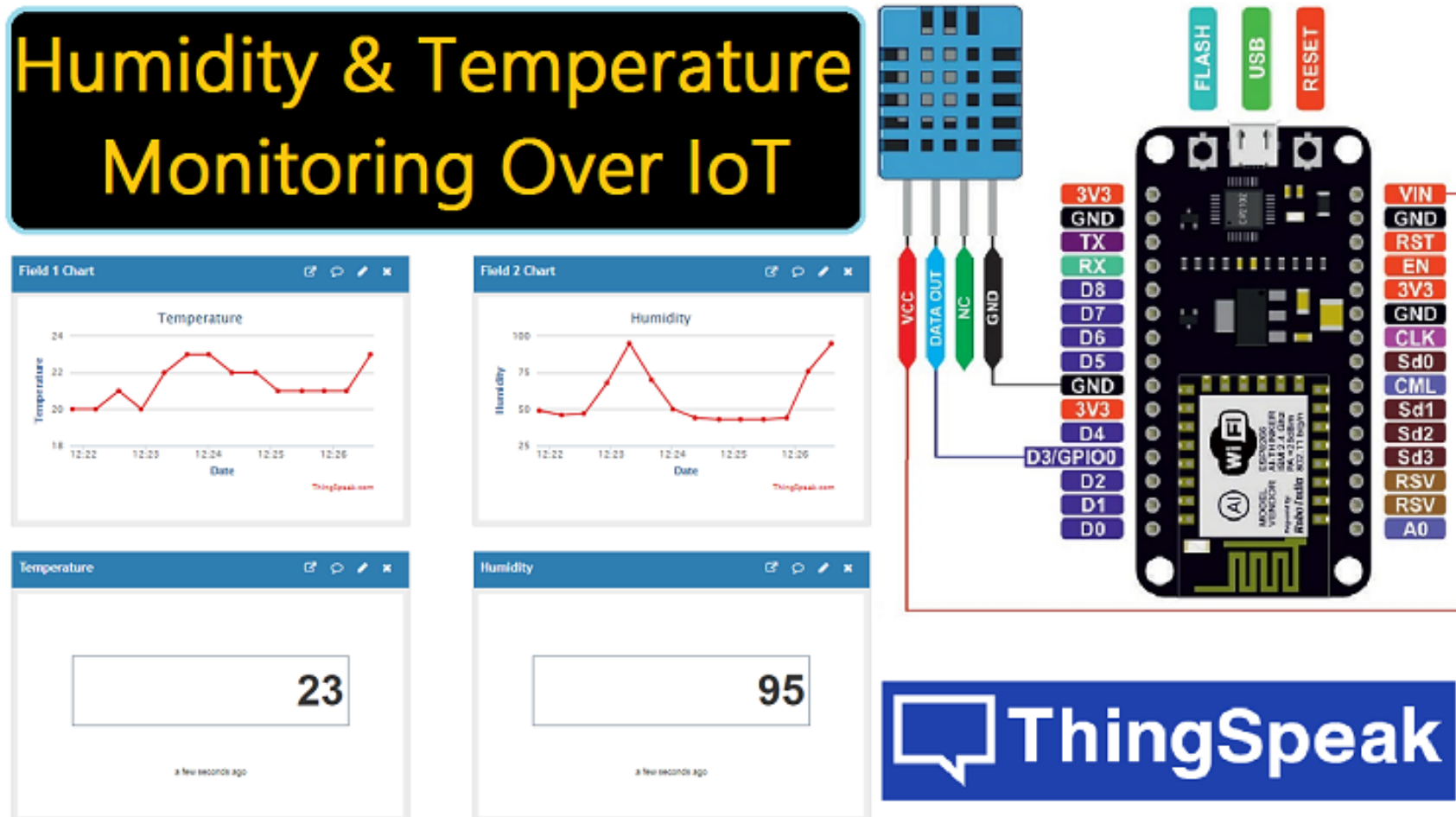
IoT Cloud Platform Landscape

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

IoT Platform Reference Architecture



ThinkSpeak : <https://thingspeak.com/>



IoT Analytics - ThingSpeak Inter x 설정 x | +

thingspeak.com

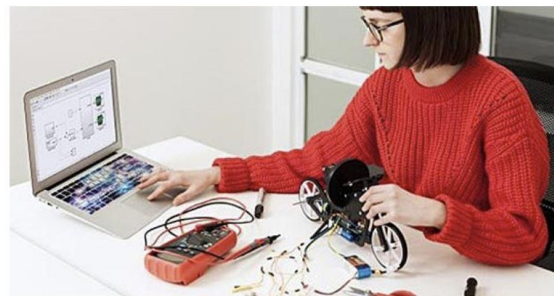
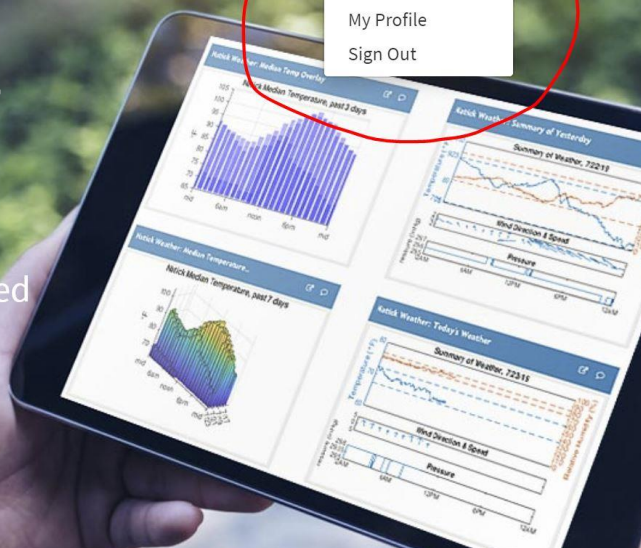
ThingSpeak™ Channels Apps Support Commercial Use How to Buy JK

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

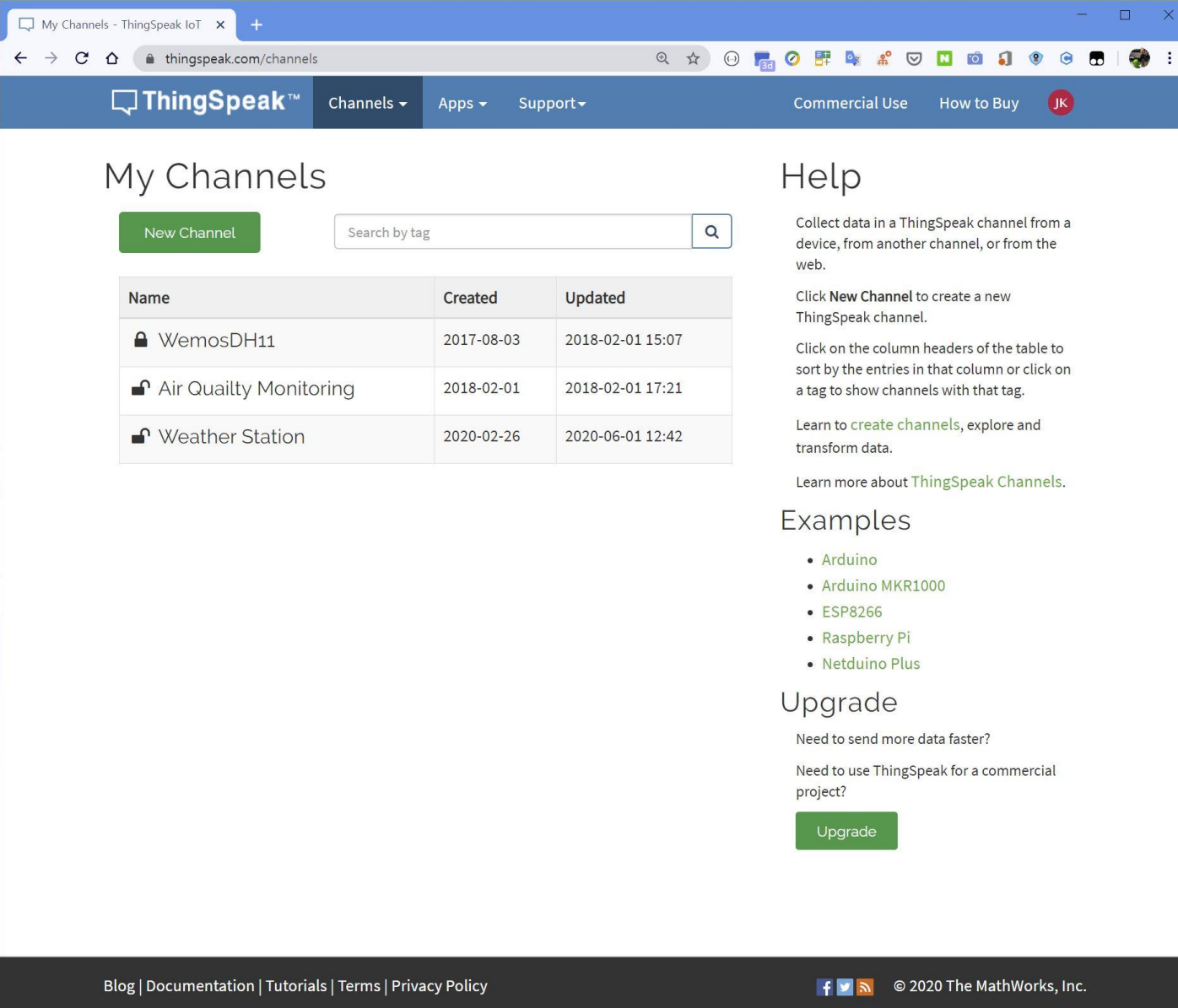


ThingSpeak for Air Quality Monitoring

Build IoT services for remote monitoring of air quality sensors, and create MATLAB models to predict pollution levels



My Channels -> New Channel



My Channels - ThingSpeak IoT

thingspeak.com/channels

ThingSpeak™ Channels Apps Support Commercial Use How to Buy JK

My Channels

[New Channel](#) Search by tag

Name	Created	Updated
WemosDH11	2017-08-03	2018-02-01 15:07
Air Quailty Monitoring	2018-02-01	2018-02-01 17:21
Weather Station	2020-02-26	2020-06-01 12:42

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

Examples

- [Arduino](#)
- [Arduino MKR1000](#)
- [ESP8266](#)
- [Raspberry Pi](#)
- [Netduino Plus](#)

Upgrade

Need to send more data faster?

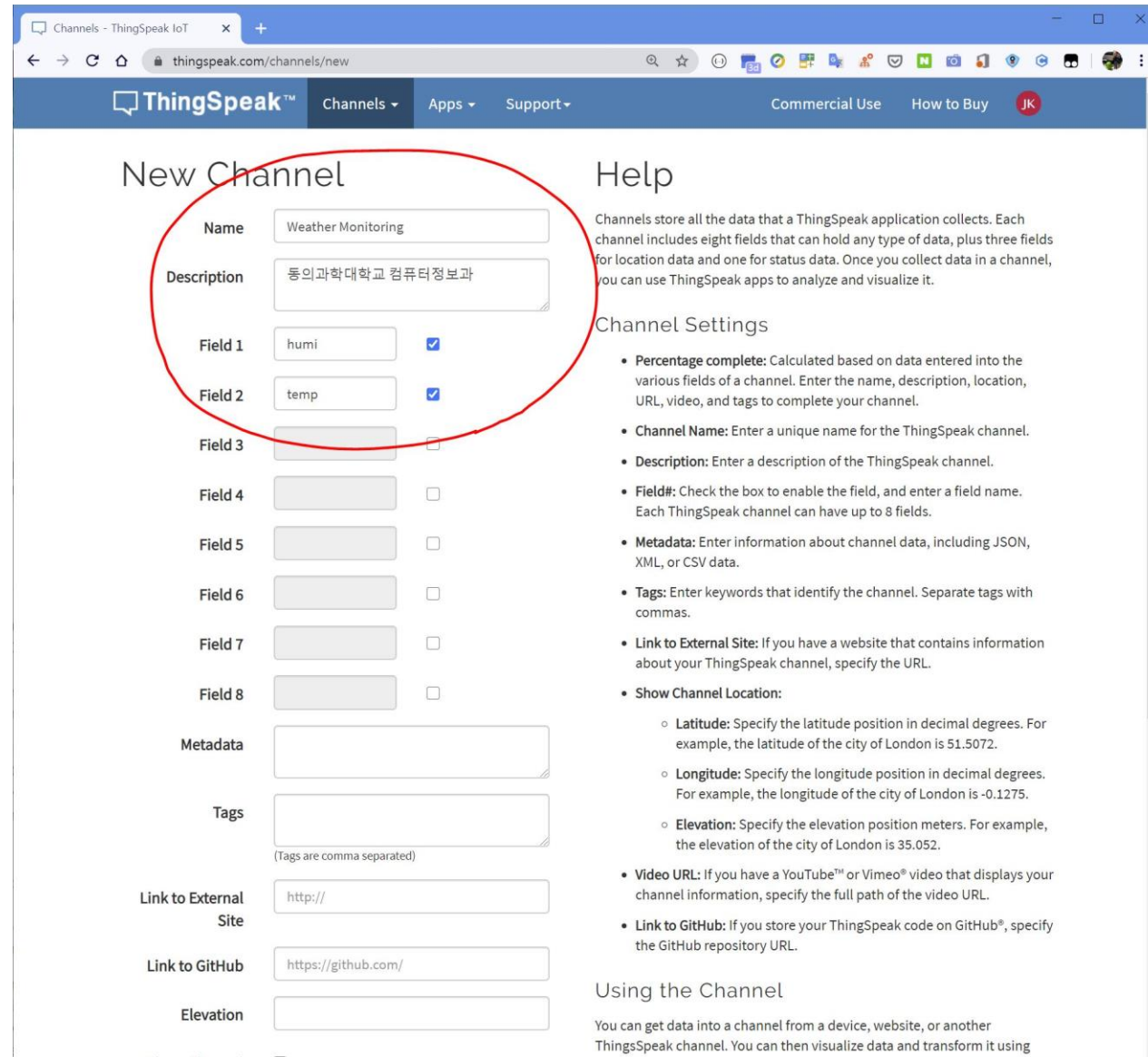
Need to use ThingSpeak for a commercial project?

[Upgrade](#)

Blog | Documentation | Tutorials | Terms | Privacy Policy

© 2020 The MathWorks, Inc.

New Channels(Weather Monitoring) -> Save



The screenshot shows the 'New Channel' form on the ThingSpeak website. A red circle highlights the 'Name', 'Description', and the first two 'Field' entries. The form includes fields for Name, Description, eight Fields (each with a name input and an enable checkbox), Metadata, Tags, Link to External Site, Link to GitHub, and Elevation. The right side of the page contains a 'Help' section with 'Channel Settings' and 'Using the Channel' information.

New Channel

Name Weather Monitoring

Description 동의과학대학교 컴퓨터정보과

Field 1 humi ☒

Field 2 temp ☒

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

Metadata

Tags
(Tags are comma separated)

Link to External Site http://

Link to GitHub https://github.com/

Elevation

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
 - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- **Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Using the Channel

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using

Weather Monitoring 대시보드

Weather Monitoring - ThingSpeak

thingspeak.com/channels/1072179/private_show

ThingSpeak™ Channels Apps Support Commercial Use How to Buy JK

Weather Monitoring

Channel ID: 1072179 | 동의과학대학교 컴퓨터정보과
Author: ditappps
Access: Private

Private View Public View Channel Settings Sharing **API Keys** Data Import / Export

+ Add Visualizations + Add Widgets

Export recent data

MATLAB Analysis MATLAB Visualization

Channel 4 of 4 < >

Channel Stats

Created: less than a minute ago
Entries: 0

Field 1 Chart

Weather Monitoring

humid

Date

ThingSpeak.com

Field 2 Chart

Weather Monitoring

temp

Date

ThingSpeak.com

Channel Status Updates

API Keys : Write API Keys

The screenshot shows the 'API Keys' page for a channel named 'Weather Monitoring' on the ThingSpeak IoT platform. The page is divided into two main sections: 'Write API Key' and 'Read API Keys'. The 'Write API Key' section is highlighted with a red oval. It displays a key 'TLUTBYDH2A7P3GCX' and a 'Generate New Write API Key' button. The 'Read API Keys' section shows a key 'FMX8J3MKUG48JCKC', a 'Note' field, and buttons for 'Save Note' and 'Delete API Key'. On the right side, there is a 'Help' section explaining API keys, an 'API Keys Settings' section with bullet points, and an 'API Requests' section with example GET requests for various endpoints.

API Keys - ThingSpeak IoT

thingspeak.com/channels/1072179/api_keys

ThingSpeak™ Channels Apps Support Commercial Use How to Buy JK

Weather Monitoring

Channel ID: 1072179 | 동의과학대학교 컴퓨터정보과
Author: ditappps
Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key TLUTBYDH2A7P3GCX

Generate New Write API Key

Read API Keys

Key FMX8J3MKUG48JCKC

Note

Save Note Delete API Key

Add New Read API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=TLUTBYDH
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1072179/feeds.
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/1072179/fields
```

Read Channel Status Updates

```
GET https://api.thingspeak.com/channels/1072179/status
```


필요한 부품들



NodeMCU



DHT11

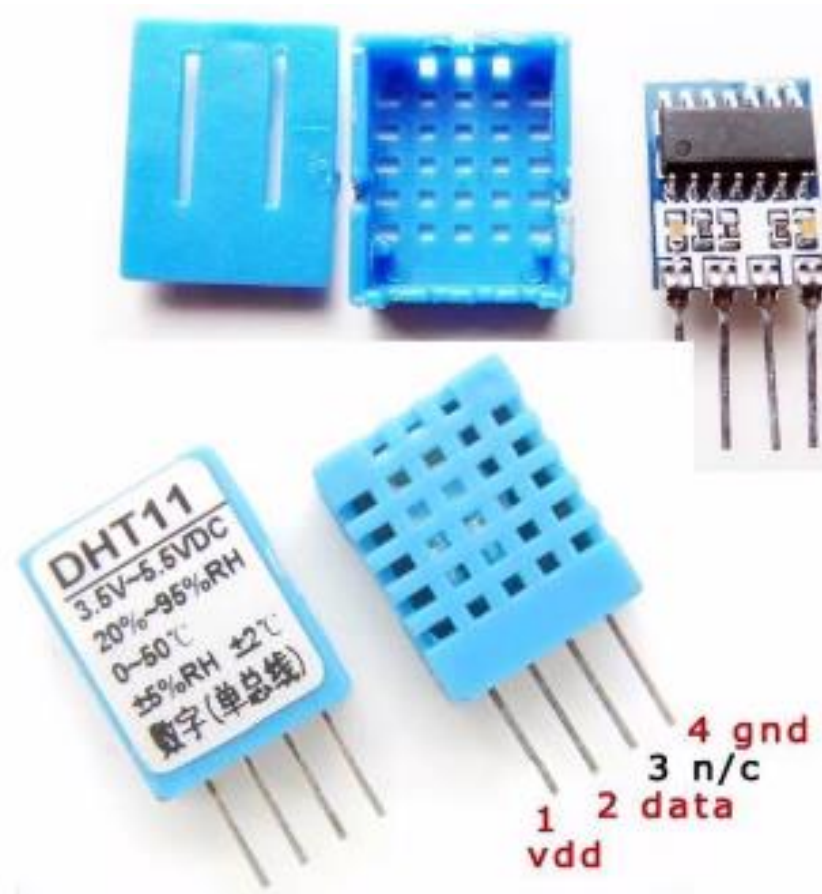


Breadboard



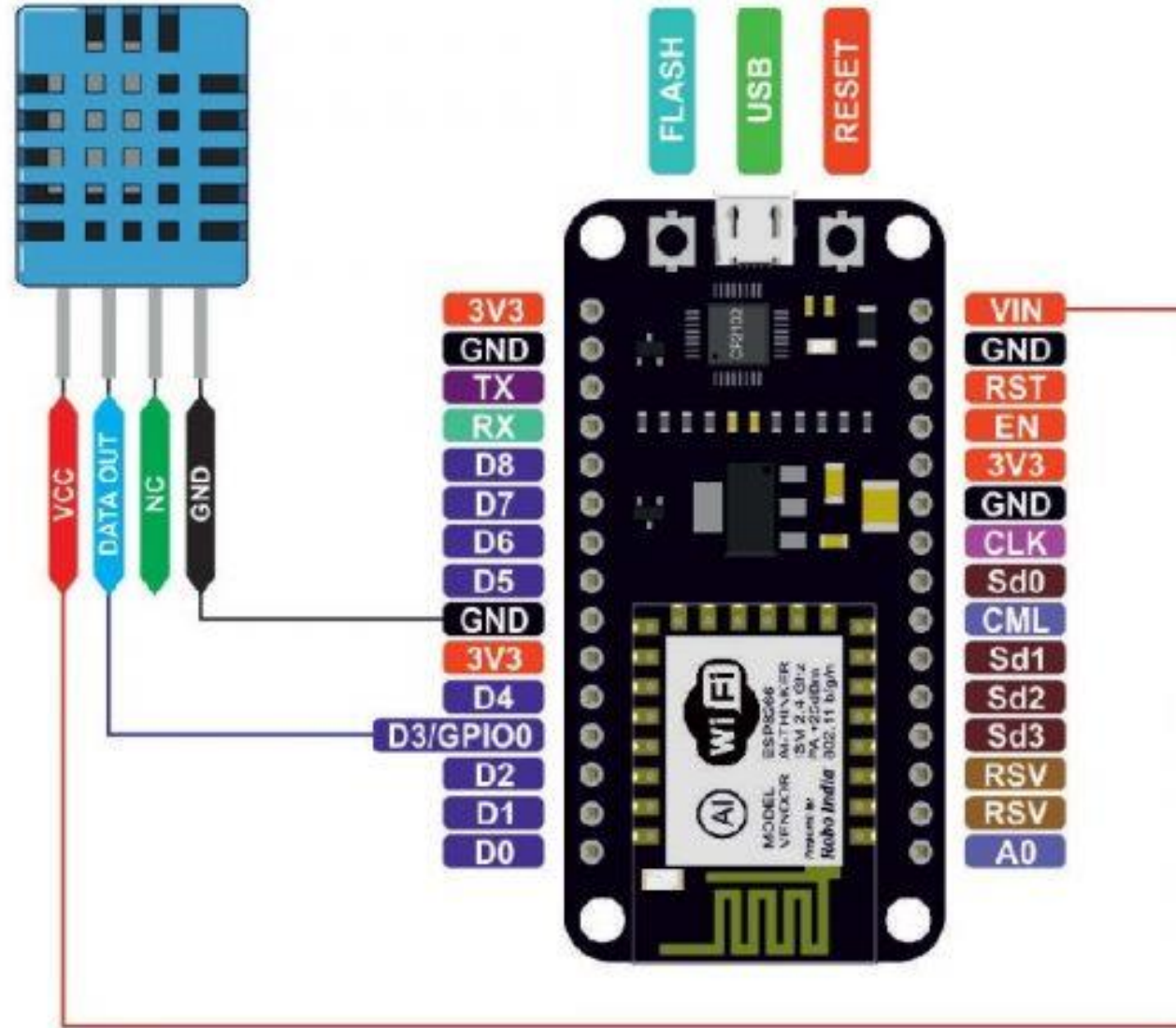
Jumper Wires

DHT11 온습도 센서



회로 연결

DHT11/DHT22



이두이노 소스 코드 :
setup()

```
DHT11_ThinkSpeak_NodeMCU_Weather_station $
1 #include <DHT.h>
2 #include <ESP8266WiFi.h>
3 #define DHTPIN D4 // D0
4 #define DHTTYPE DHT11
5
6 const char* ssid = "melon";
7 const char* password = "deitcs3217";
8
9 const char* server = "api.thingspeak.com";
10 String apiKey = "VJF12D7B55H1ONG1"; // Write API Key
11
12 WiFiClient client;
13 DHT dht(DHTPIN, DHTTYPE, D4);
14
15 void setup() {
16     Serial.begin(9600);
17     Serial.println();
18     Serial.println("ThingSpeak");
19     // DHT11 초기화
20     dht.begin();
21     Serial.println();
22     Serial.print("Connecting to ");
23     Serial.println(ssid);
24     // Wi-Fi 초기화
25     WiFi.begin(ssid, password);
26
27     while (WiFi.status() != WL_CONNECTED) {
28         delay(500);
29         Serial.print(".");
30     }
31     Serial.println("");
32     Serial.println("WiFi connected!!");
33 }
```

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 = "H38TEGNC0X0W43BB"; // 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)
    {
        delay(5000);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

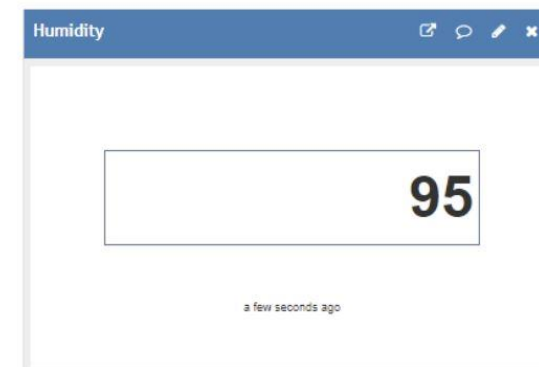
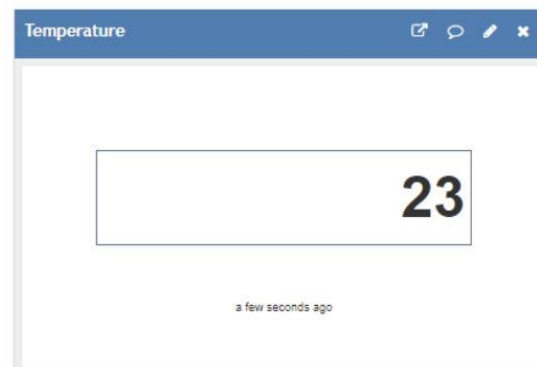
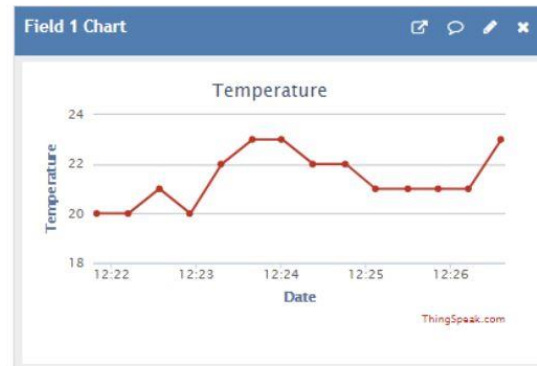
void loop()
{
    float temperature = dht.temperature();
    float humidity = dht.humidity();
    String data = "t=" + String(temperature) + "&h=" + String(humidity) + "&k=" + apiKey;
    HTTPClient http;
    http.begin("https://api.thingSpeak.com/update/1/" + data);
    int httpCode = http.GET();
    if (httpCode == HTTP_CODE_OK)
    {
        Serial.println("Data sent to ThingSpeak");
    }
    else
    {
        Serial.println("Error sending data to ThingSpeak");
    }
    delay(10000);
}
```

COM8

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...

Done uploading.

..... [51%]
..... [77%]
..... [100%]



REST API 사용하기

API Keys - ThingSpeak IoT

thingspeak.com/channels/1072179/api_keys

ThingSpeak™ Channels Apps Support Commercial Use How to Buy JK

Channel ID: 1072179
Author: ditappps
Access: Private

동의과학대학교 컴퓨터정보과

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key: TLUTBYDH2A7P3GCX

Generate New Write API Key

Read API Keys

Key: FMX8J3MKUG48JCKC

Note:

Save Note Delete API Key

Add New Read API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

GET `https://api.thingspeak.com/update?api_key=TLUTBYDH2A7P3GCX`

Read a Channel Feed

GET `https://api.thingspeak.com/channels/1072179/feeds`

Read a Channel Field

GET `https://api.thingspeak.com/channels/1072179/fields`

Read Channel Status Updates

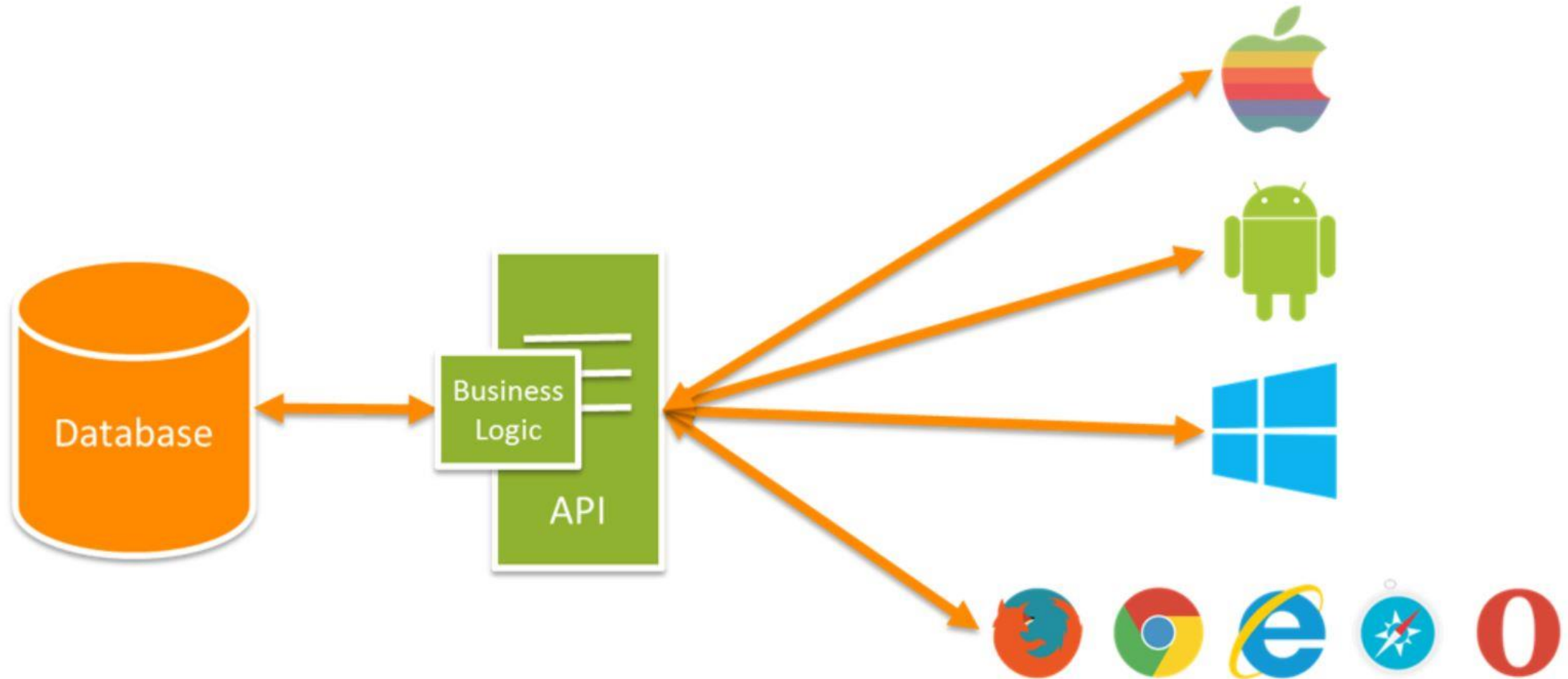
GET `https://api.thingspeak.com/channels/1072179/status`

Learn More

Blog | Documentation | Tutorials | Terms | Privacy Policy

© 2020 The MathWorks, Inc.

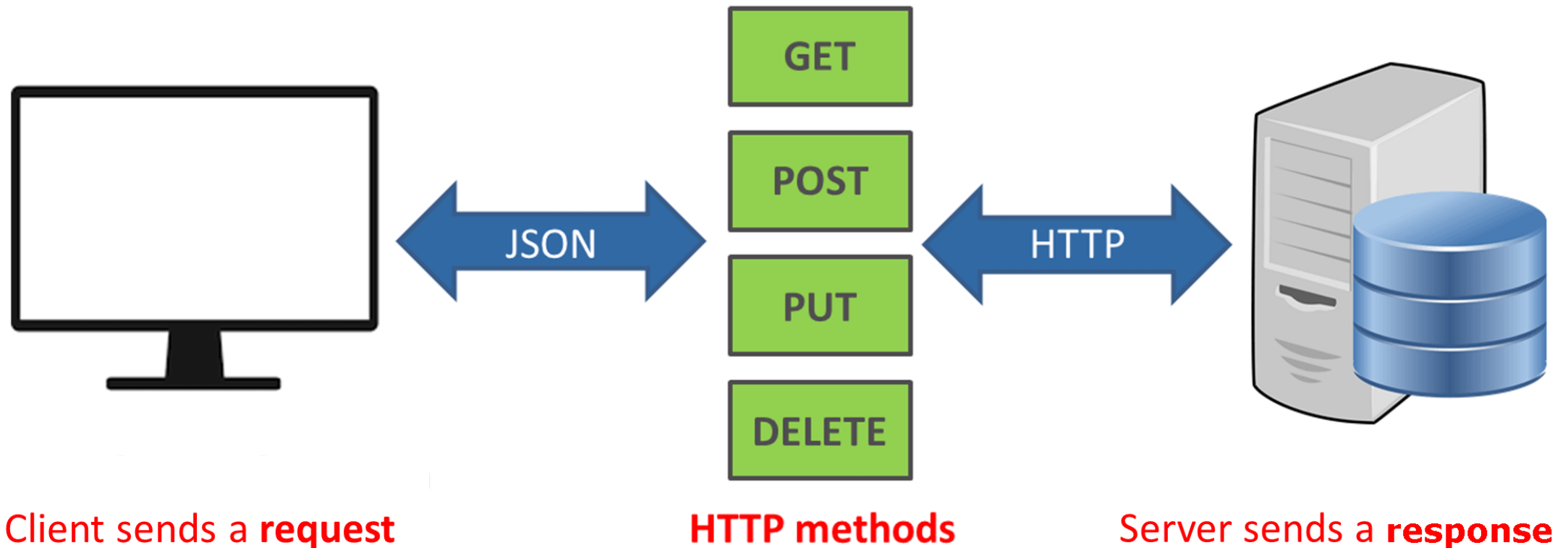
API : Application Programming Interface



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?

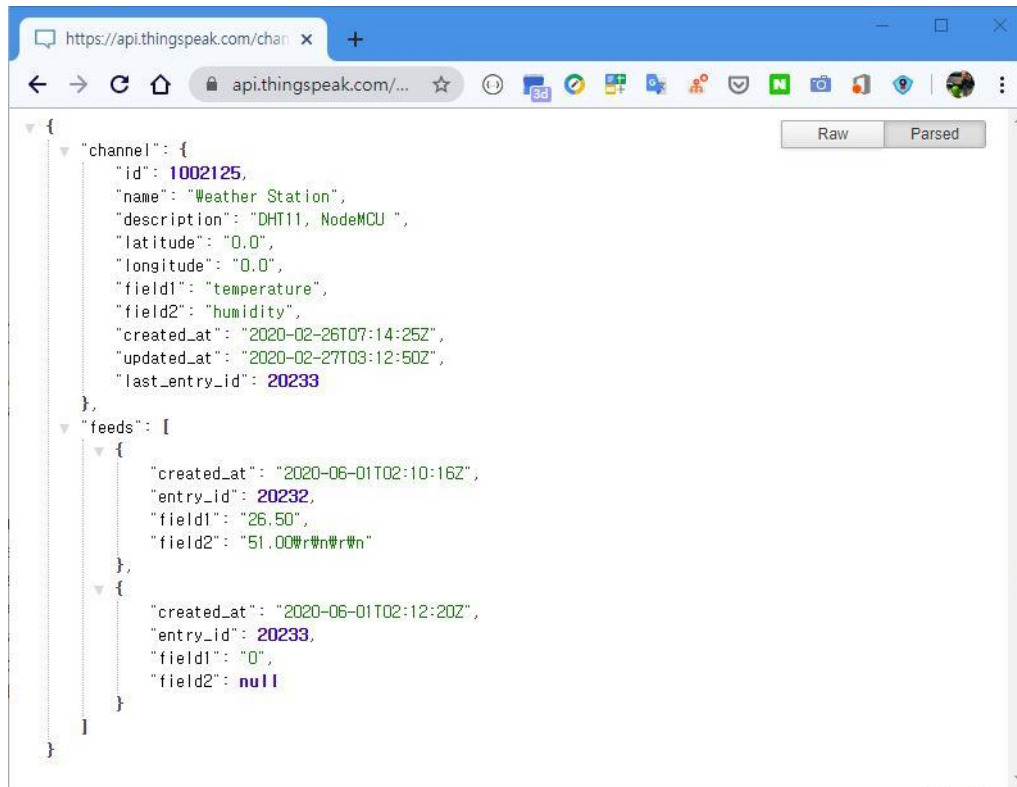
CRUD	HTTP verbs	Route
resource들의 목록을 표시	GET	/resource
resource 하난의 내용을 표시	GET	/resource/:id
resource를 생성	POST	/resource
resource를 수정	PUT	/resource/:id
resource를 삭제	DELETE	/resource/:id

“REST”

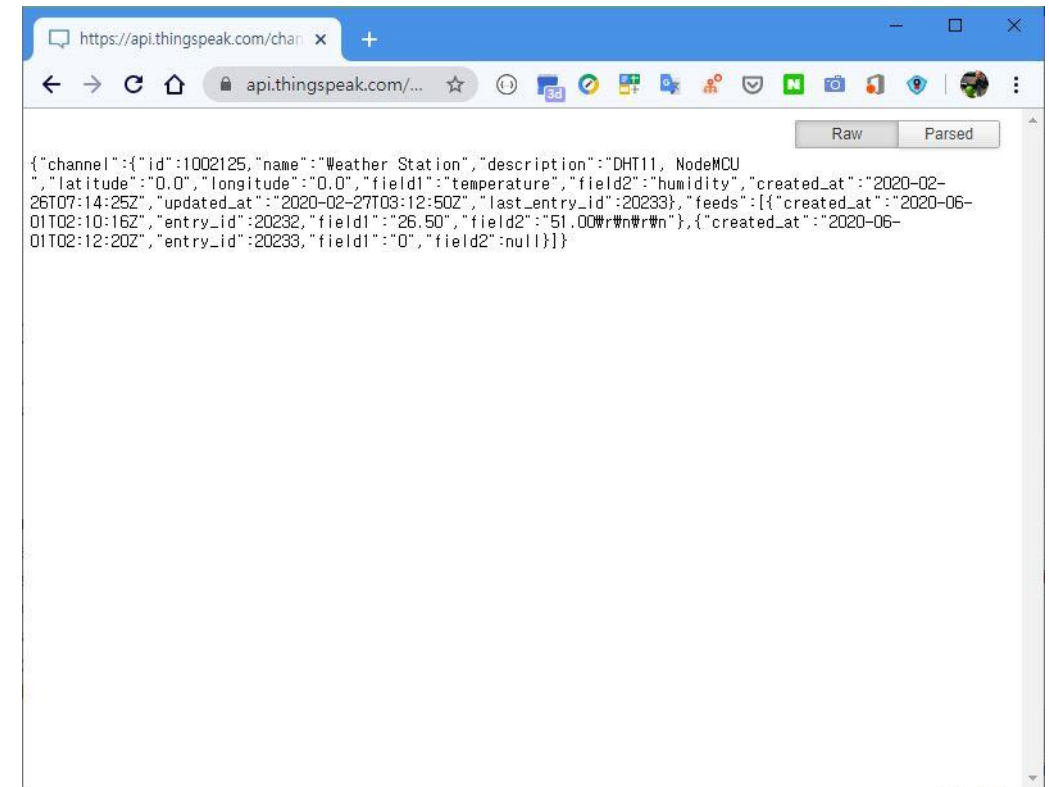
GET	/movies	Get list of movies
GET	/movies/:id	Find a movie by its ID
POST	/movies	Create a new movie
PUT	/movies	Update an existing movie
DELETE	/movies	Delete an existing movie

API Requests : Read a Channel Feed

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



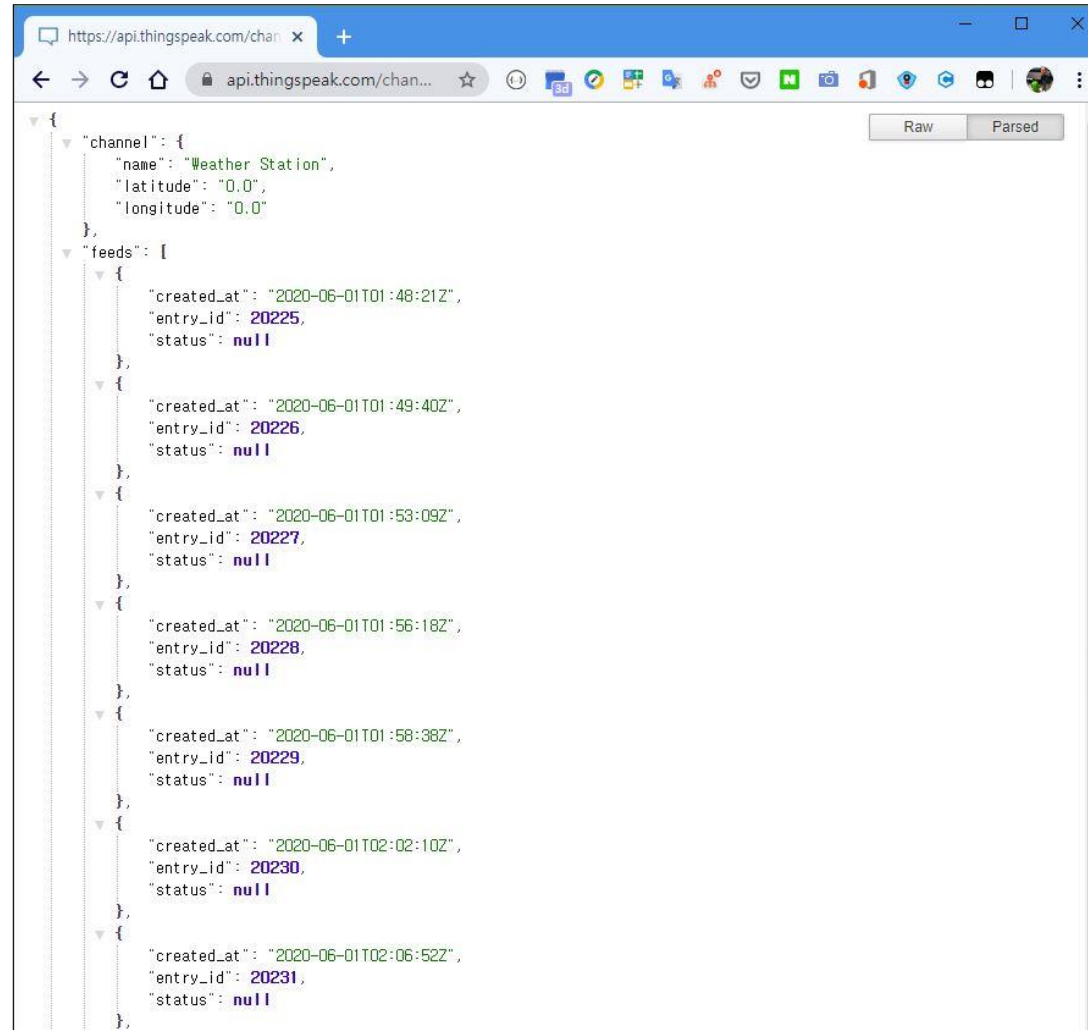
```
{
  "channel": {
    "id": 1002125,
    "name": "Weather Station",
    "description": "DHT11, NodeMCU ",
    "latitude": "0.0",
    "longitude": "0.0",
    "field1": "temperature",
    "field2": "humidity",
    "created_at": "2020-02-26T07:14:25Z",
    "updated_at": "2020-02-27T03:12:50Z",
    "last_entry_id": 20233
  },
  "feeds": [
    {
      "created_at": "2020-06-01T02:10:16Z",
      "entry_id": 20232,
      "field1": "26.50",
      "field2": "51.00\r\n\r\n"
    },
    {
      "created_at": "2020-06-01T02:12:20Z",
      "entry_id": 20233,
      "field1": "0",
      "field2": null
    }
  ]
}
```



```
{
  "channel": {
    "id": 1002125, "name": "Weather Station", "description": "DHT11, NodeMCU ",
    "latitude": "0.0", "longitude": "0.0", "field1": "temperature", "field2": "humidity", "created_at": "2020-02-26T07:14:25Z", "updated_at": "2020-02-27T03:12:50Z", "last_entry_id": 20233, "feeds": [
      {
        "created_at": "2020-06-01T02:10:16Z", "entry_id": 20232, "field1": "26.50", "field2": "51.00\r\n\r\n"
      },
      {
        "created_at": "2020-06-01T02:12:20Z", "entry_id": 20233, "field1": "0", "field2": null
      }
    ]
  }
}
```

API Requests : Read a Status Update

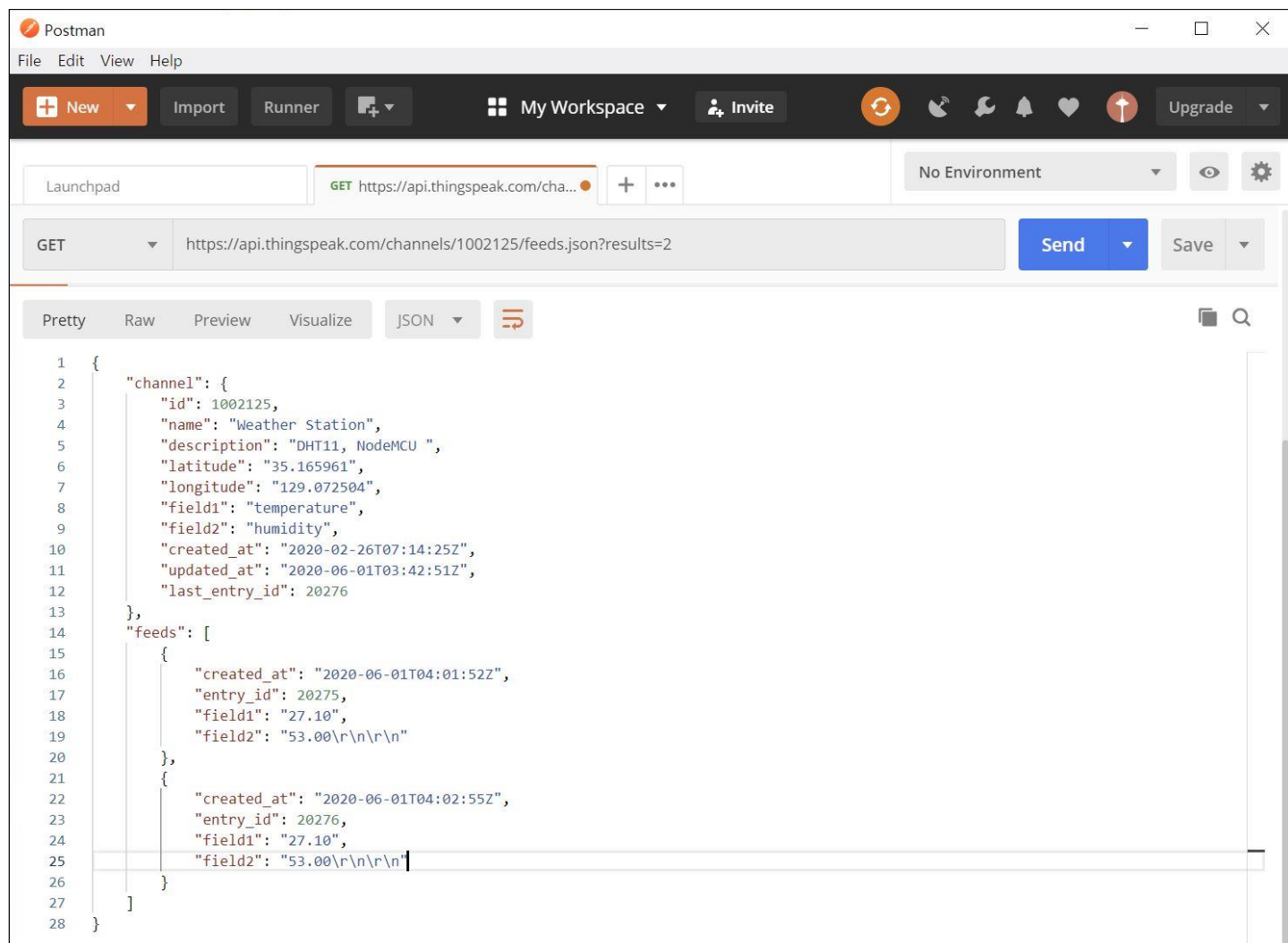
- GET https://api.thingspeak.com/channels/1072179/status.json?api_key=FMX8J3MKUG48JCKC



```
{
  "channel": {
    "name": "Weather Station",
    "latitude": "0.0",
    "longitude": "0.0"
  },
  "feeds": [
    {
      "created_at": "2020-06-01T01:48:21Z",
      "entry_id": 20225,
      "status": null
    },
    {
      "created_at": "2020-06-01T01:49:40Z",
      "entry_id": 20226,
      "status": null
    },
    {
      "created_at": "2020-06-01T01:53:09Z",
      "entry_id": 20227,
      "status": null
    },
    {
      "created_at": "2020-06-01T01:56:18Z",
      "entry_id": 20228,
      "status": null
    },
    {
      "created_at": "2020-06-01T01:58:38Z",
      "entry_id": 20229,
      "status": null
    },
    {
      "created_at": "2020-06-01T02:02:10Z",
      "entry_id": 20230,
      "status": null
    },
    {
      "created_at": "2020-06-01T02:06:52Z",
      "entry_id": 20231,
      "status": null
    }
  ]
}
```

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/OFF 하도록 구성하시오.
- 데이터가 정상적으로 보내지면 LED는 15초 간격으로 ON/OFF 하고, 그렇지 않으면 LED는 작동하지 않는다.
- 과제 제출 기한 : 다음주 월요일 까지(6/8), jhkim3217@gmail.com

참고

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