

# IoT 플랫폼 활용 기초 교육

2019.02.19 ~ 2019.02.22

## 목차

---

### III. ARTIK Cloud

#### 5. 외부 API 사용하기

DSLAB

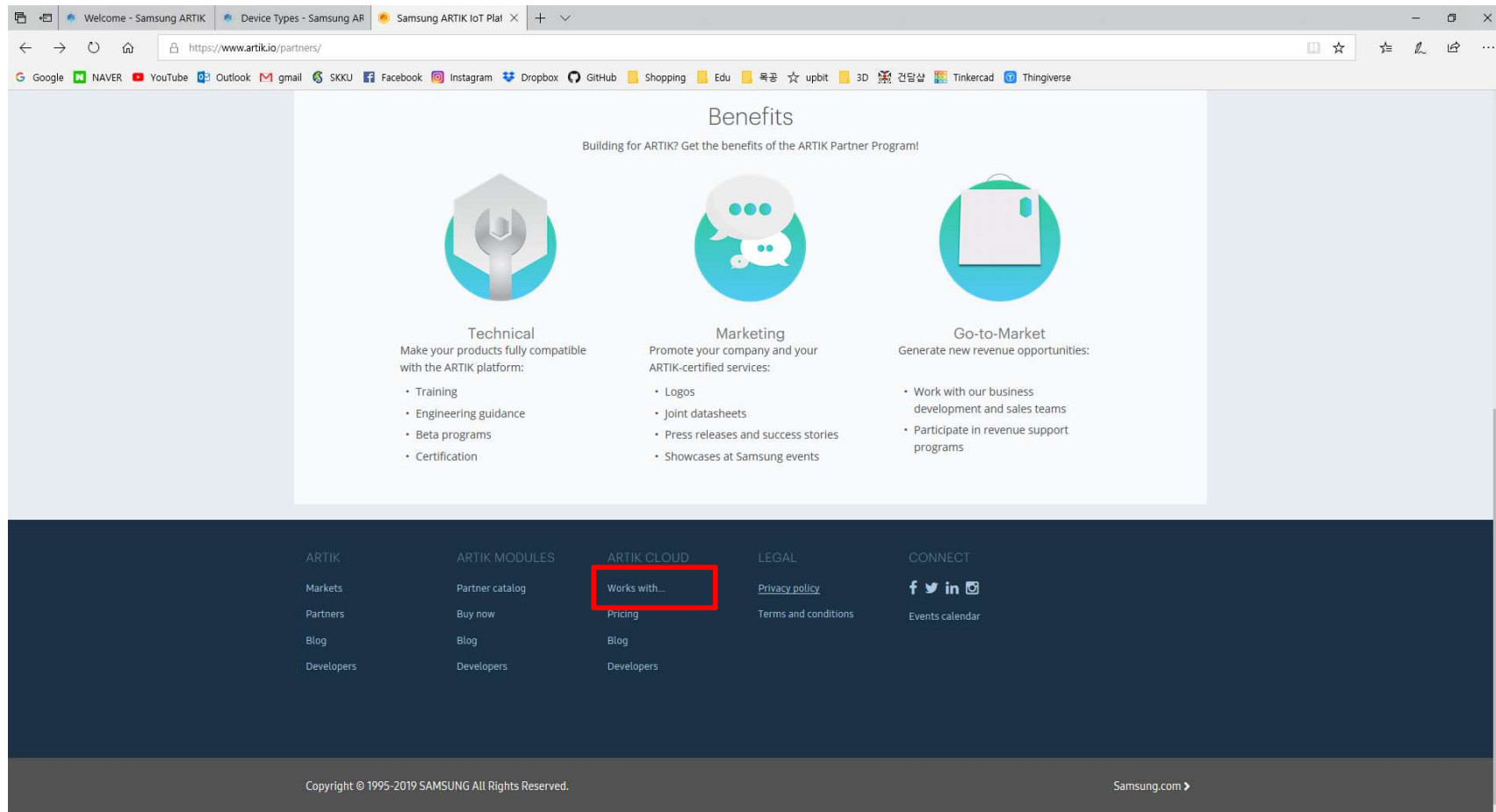
## 5. 외부 API 이용하기

- Weather API -

# Login to Open Weather Map (1)

## ❖ Open Weather Map에 접속하기

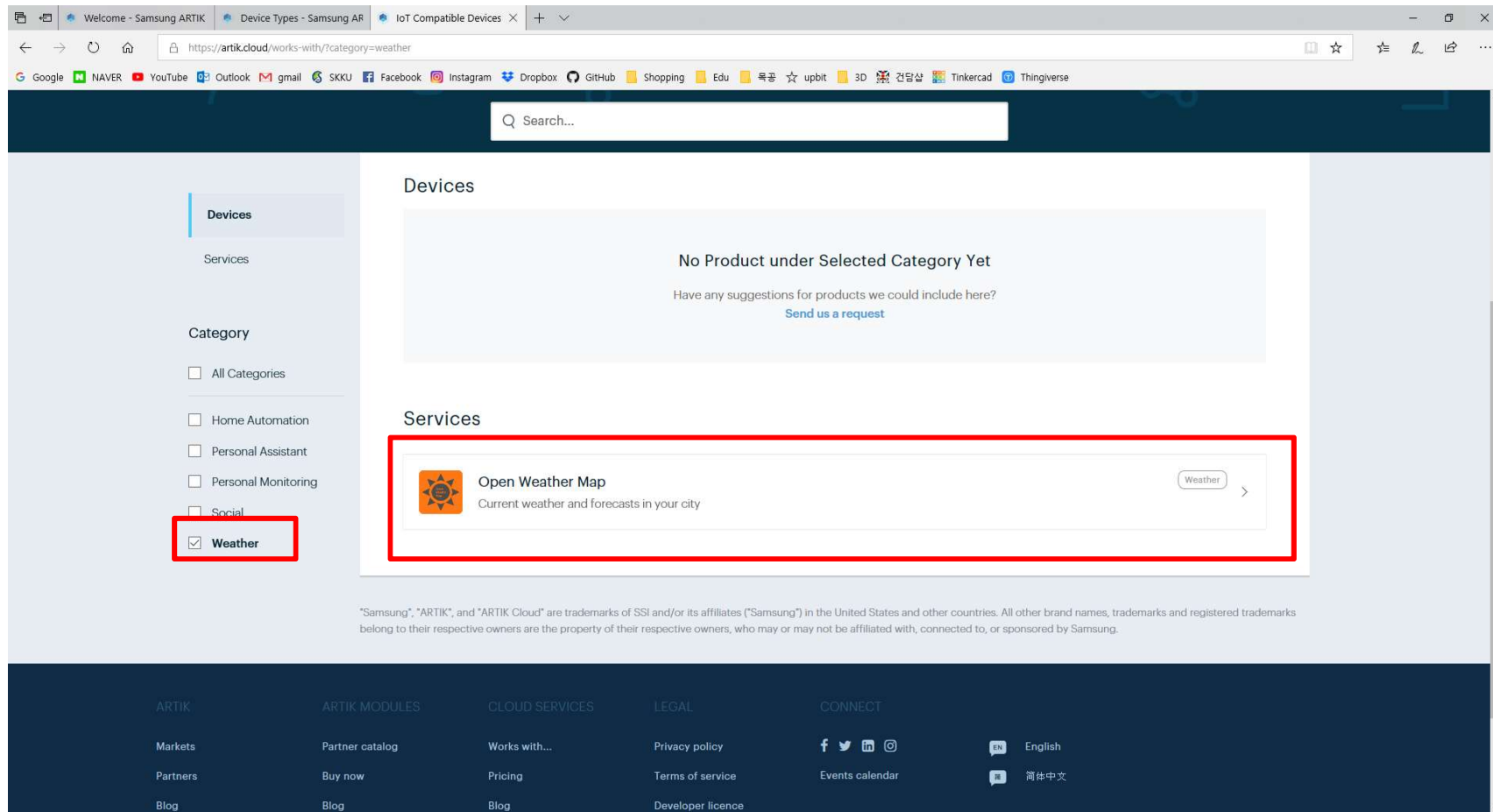
➤ ARTIK Cloud 맨 아래에 [Work with...] 클릭



# Login to Open Weather Map (2)

## ❖ ARTIK Cloud에서 지원하는 서비스

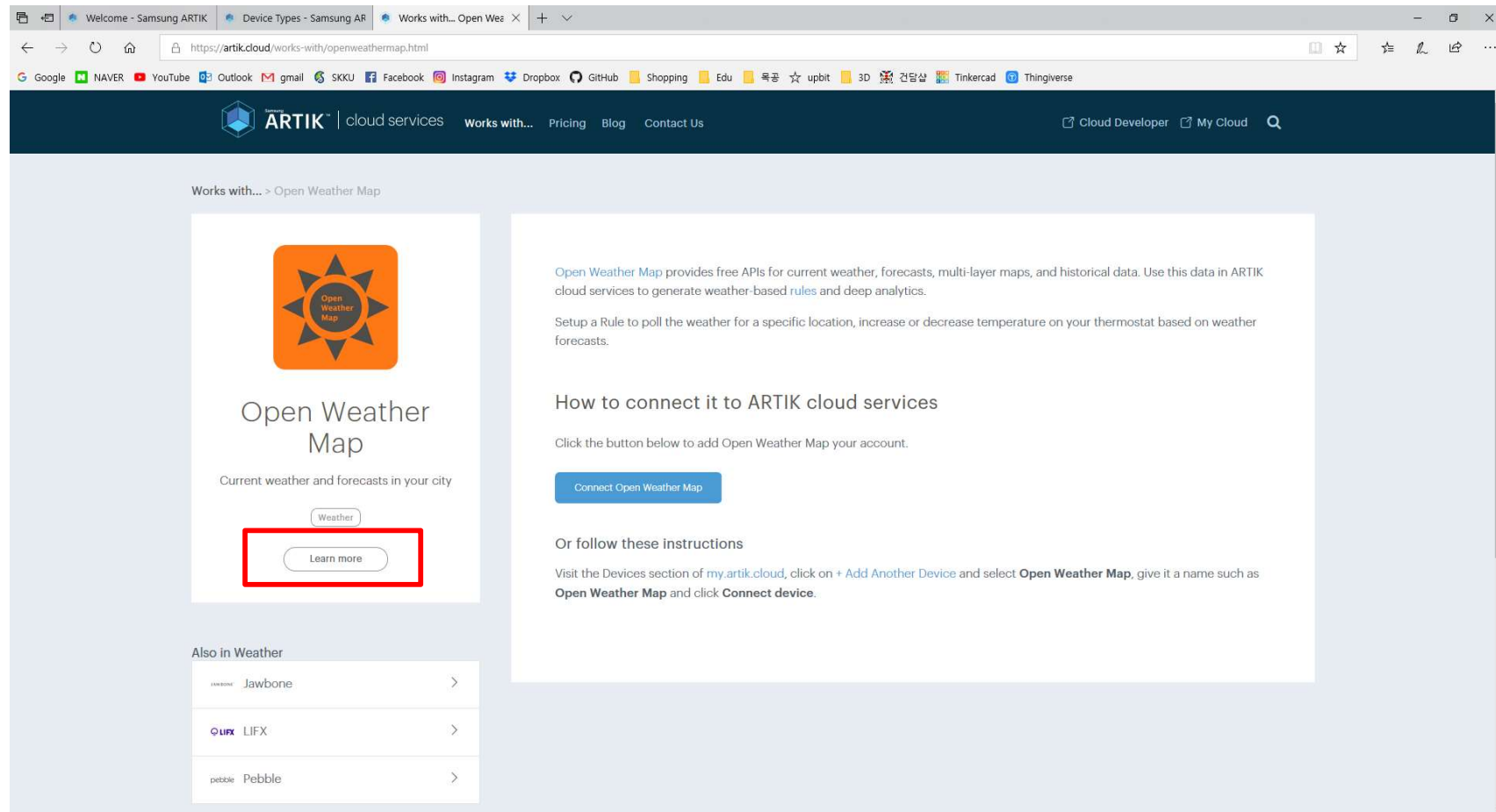
### ➤ 다양한 서비스 중 [Weather]를 선택하고 Services 중 Open Weather Map 클릭



# Login to Open Weather Map (3)

## ❖ Open Weather Map에 접속

➤ [learn more]을 클릭하여 Open Weather Map 사이트에 접속





# Login to Open Weather Map (4)

## ❖ Open Weather Map에 회원가입

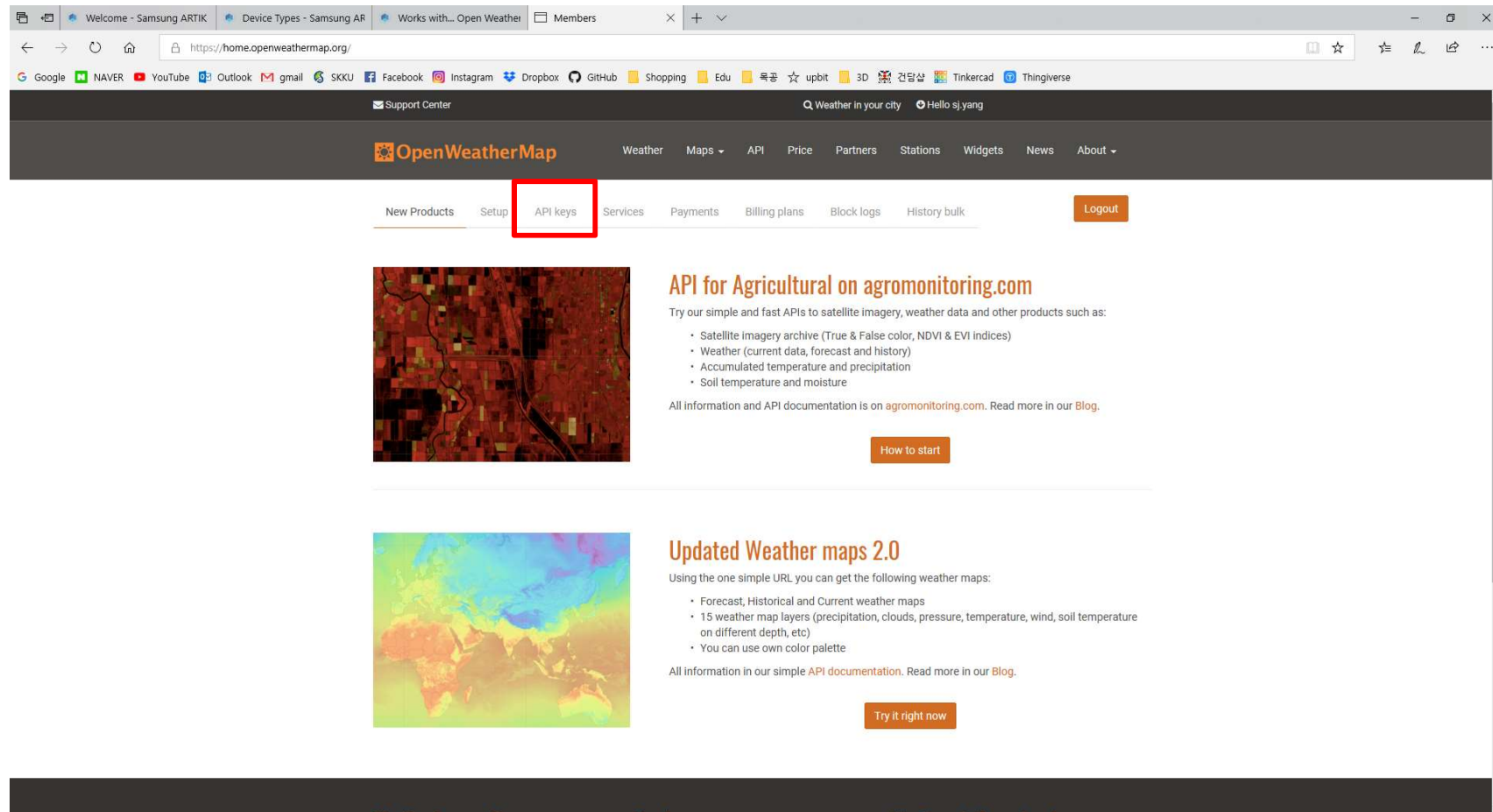
➤ sign-in을 눌러 회원가입 후 로그인 (기존 회원일 경우 스킵)

The screenshot shows the OpenWeatherMap website interface. The browser address bar displays <https://openweathermap.org/>. The navigation bar includes links for Weather, Maps, Guide, API, Price, Partners, Stations, Widgets, and Blog. The main banner features the text "We Deliver 2 Billion Forecasts Per Day" and "1,500 new subscribers a day | 1,200,000 customers | 20+ weather APIs". Below the banner is a search bar with the placeholder "Your city name" and a "Search" button. The "Sign In" button is highlighted with a red box. The page also displays the current weather and forecasts for Pusan-gwangyöksi, KR, with a temperature of -1°C and a "Wrong data?" message. A table shows weather details: Wind (Light breeze, 3.1 m/s, North-northwest (330)), Cloudiness (Sky is clear), Pressure (1027 hpa), Humidity (34 %), Sunrise (07:32), and Sunset (17:26). A line graph shows temperature and precipitation trends. The footer includes a cookie notice: "We use cookies to enhance your experience of our site. By using openweathermap.org, you agree to our [privacy policy](#)."

# Login to Open Weather Map (5)

## ❖ login 후 API Key 받기

### ➤ 메뉴의 [API Keys]를 클릭하여 API Key 확인





# Login to Open Weather Map (6)

## ❖ API Key 받기

➤ 몇 분에서 몇시간 정도 지난 후 사용 가능

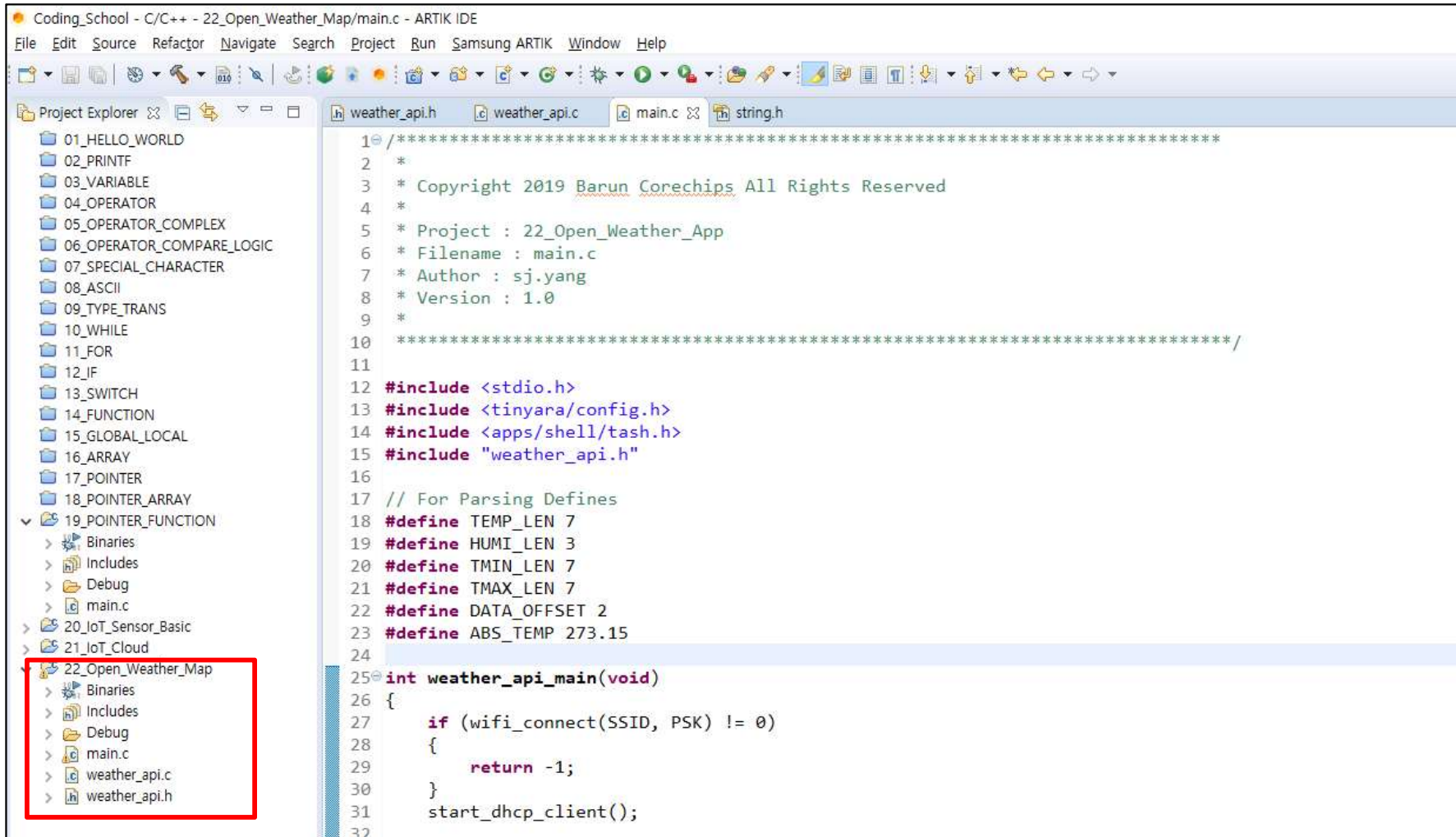
The screenshot shows the OpenWeatherMap API keys management page. The browser address bar displays `https://home.openweathermap.org/api_keys`. The page has a dark header with the OpenWeatherMap logo and navigation links. Below the header, there's a sub-header with links like 'New Products', 'Setup', 'API keys', 'Services', 'Payments', 'Billing plans', 'Block logs', and 'History bulk'. A blue notification box states: 'API key will be activated and ready for using within a couple of hours. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.'

The main content area contains a table with one key listed. The 'Key' column value is `a667a1402db485e82f9eb341a088445d`, which is highlighted with a red rectangular box. The 'Name' column for this key is 'Default'. To the right of the table is a 'Create key' section with a text input field labeled '\* Name' and a 'Generate' button.

The footer of the page contains four columns of links: 'Weather in your city' (Find your city, Weather maps), 'Map layers' (Examples of weather map layers, Map styles legend, Libraries to connect weather layers), 'Weather station network' (How to connect your weather station), and 'About'.

# Source Code Flash

## ❖ 22\_Open\_Weather\_Map 프로젝트를 ARTIK에 flash



The screenshot displays the ARTIK IDE interface. On the left, the Project Explorer shows a list of projects from 01\_HELLO\_WORLD to 22\_Open\_Weather\_Map. The 22\_Open\_Weather\_Map project is selected and highlighted with a red rectangle. Its sub-items, including Binaries, Includes, Debug, main.c, weather\_api.c, and weather\_api.h, are also visible. The main editor window shows the source code for weather\_api.h, which includes copyright information, project details, and code for parsing defines and the main function.

```
1 /*****  
2 *  
3 * Copyright 2019 Barun Corechips All Rights Reserved  
4 *  
5 * Project : 22_Open_Weather_App  
6 * Filename : main.c  
7 * Author : sj.yang  
8 * Version : 1.0  
9 *  
10 *****/  
11  
12 #include <stdio.h>  
13 #include <tinyara/config.h>  
14 #include <apps/shell/tash.h>  
15 #include "weather_api.h"  
16  
17 // For Parsing Defines  
18 #define TEMP_LEN 7  
19 #define HUMI_LEN 3  
20 #define TMIN_LEN 7  
21 #define TMAX_LEN 7  
22 #define DATA_OFFSET 2  
23 #define ABS_TEMP 273.15  
24  
25 int weather_api_main(void)  
26 {  
27     if (wifi_connect(SSID, PSK) != 0)  
28     {  
29         return -1;  
30     }  
31     start_dhcp_client();  
32 }
```

# Source Code Flash

## ❖ Source code files

The screenshot displays the ARTIK IDE interface with the Project Explorer on the left and the main editor on the right. The Project Explorer shows a tree structure of files and folders. The main editor displays the code for `main.c`. Three colored boxes with arrows point to specific files in the Project Explorer and their corresponding code sections in the main editor:

- Red box:** Points to `main.c` in the Project Explorer and the `main.c` file in the main editor. The text inside the box is: `<main.c>`  
Open Weather Map의 데이터 수신 및 데이터 파싱
- Blue box:** Points to `weather_api.c` in the Project Explorer and the `weather_api.c` file in the main editor. The text inside the box is: `<weather_api.c>`  
wifi 연결을 위한 함수 정의
- Green box:** Points to `weather_api.h` in the Project Explorer and the `weather_api.h` file in the main editor. The text inside the box is: `<weather_api.h>`  
wifi 연결을 위한 함수 선언 및 변수 선언

The main editor shows the following code snippets:

```
1- /* **** */
2-
3-
4-
5-
6-
7-
8-
9-
10- **
11-
12- #include <stdio.h>
13- #include <stdlib.h>
14- #include <string.h>
15- #include <unistd.h>
16-
17- // For Parsing Defines
18- #define TEMP_LEN 7
19- #define HUM_LEN 7
20- #define TMI_LEN 7
21- #define TMA_LEN 7
22- #define DA_LEN 7
23- #define ABS_LEN 7
24-
25- int weather_api_main(void)
26- {
27-     if (wifi_connect(SSID, PSK) != 0)
28-     {
29-         return -1;
30-     }
31-     start_dhcp_client();
32- }
```



# Source Code 설정

## ❖ weather\_api.h

Coding\_School - C/C++ - 22\_Open\_Weather\_Map/weather\_api.h - ARTIK IDE

File Edit Source Refactor Navigate Search Project Run Samsung ARTIK Window Help

Project Explorer

- 01\_HELLO\_WORLD
- 02\_PRINTF
- 03\_VARIABLE
- 04\_OPERATOR
- 05\_OPERATOR\_COMPLEX
- 06\_OPERATOR\_COMPARE\_LOGIC
- 07\_SPECIAL\_CHARACTER
- 08\_ASCII
- 09\_TYPE\_TRANS
- 10\_WHILE
- 11\_FOR
- 12\_IF
- 13\_SWITCH
- 14\_FUNCTION
- 15\_GLOBAL\_LOCAL
- 16\_ARRAY
- 17\_POINTER
- 18\_POINTER\_ARRAY
- 19\_POINTER\_FUNCTION
  - Binaries
  - Includes
  - Debug
  - main.c
- 20\_IoT\_Sensor\_Basic
- 21\_IoT\_Cloud
- 22\_Open\_Weather\_Map
  - Binaries
  - Includes

weather\_api.h

```
15 #include <stdio.h>
16 #include <string.h>
17 #include <errno.h>
18 #include <stdlib.h>
19 #include <artikel_module.h>
20 #include <artikel_wifi.h>
21 #include <artikel_network.h>
22 #include <readline.h>
23 #include <netdb.h>
24 #include <net/lwip/sockets.h>
25
26 #define WIFI_SCAN_TIMEOUT 15
27 #define WIFI_CONNECT_TIMEOUT 30
28 #define WIFI_DISCONNECT_TIMEOUT 10
29
30 #define BACKLOG 10
31 #define MAX_DATA_SIZE 1500
32
33 #define SSID "B_C0050UTPS" //FIX
34 #define PSK "12345678901" //FIX
35
36 struct callback_result {
37     sem_t sem;
38     artikel_wifi_connection_info info;
39     artikel_error error;
40 };
41
```

SSID : 접속할 wifi 이름  
PSK : 접속할 wifi 비밀번호

# Source Code 설정

## ❖ main.c

sp/main.c - ARTIK IDE

Project Run Samsung ARTIK Window Help

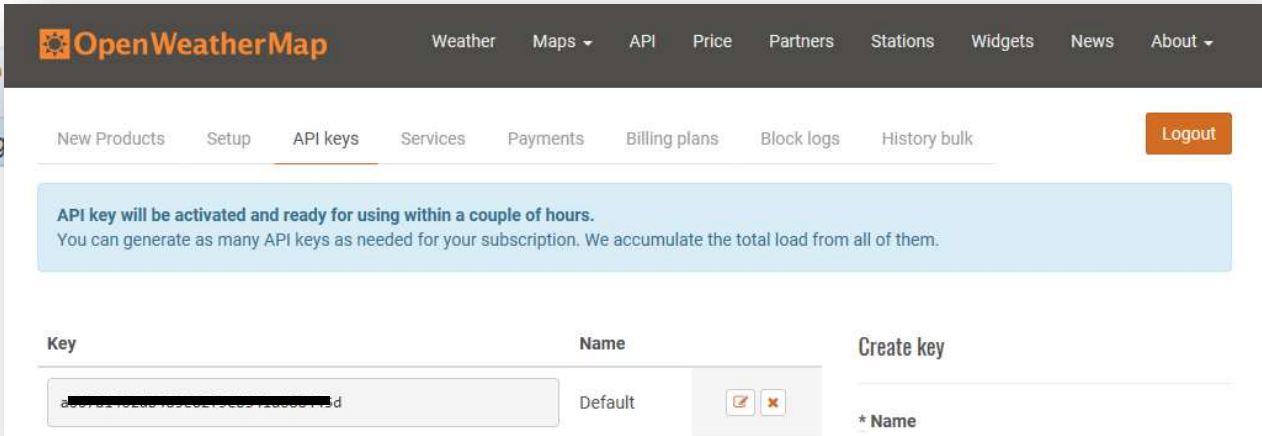
weather\_api.h weather\_api.c \*main.c string

```

19 #define HUMI_LEN 3
20 #define TMIN_LEN 7
21 #define TMAX_LEN 7
22 #define DATA_OFFSET 2
23 #define ABS_TEMP 273.15
24
25 int weather_api_main(void)
26 {
27     if (wifi_connect(SSID, PSK) != 0)
28     {
29         return -1;
30     }
31     start_dhcp_client();
32
33     int sockfd, len;
34     int ret;
35     struct sockaddr_in server_addr;
36     char str[MAX_DATA_SIZE]={0};
37     //char *message = "POST /data/2.5/weather?q=<city name>&appid=<my api key> HTTP/1.0\r\n\r\n";
38     //q=<원하는 도시명>,appid=<my api key>
39     char *message = "POST /data/2.5/weather?q=suigen&appid=a6b7d1162db483c02f5c651d880445d HTTP/1.0\r\n\r\n";
40     char *host = "api.openweathermap.org"; //weather api
  
```

도시 이름(수원 = suigen)

앞에서 확인한 API Key 입력







# 날씨 정보 수신

## ❖ main.c

```

Project Run Samsung ARTIK Window Help
weather_api.h weather_api.c *main.c string.h
70
71 // connect
72 //printf("Connect to server...\n");
73 ret = lwip_connect(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
74 if (ret == -1) {
75     printf("[Error] lwip_connect()\n");
76     return -1;
77 }
78
79 len = strlen(message);
80 ret = lwip_write(sockfd, message, len);
81 if (ret != (int)len) {
82     printf("[Error] lwip_write()\n");
83     return -1;
84 }
85 lwip_read(sockfd, str, MAX_DATA_SIZE);
86 printf("\nFull data : %s\n\n", str); //Use api by parsing this string
87

```

Open Weather Map으로 부터 날씨정보 수신하는 부분  
날씨정보의 모든 내용이 str문자열에 저장

받은 날씨정보 출력

```

{"coord":{"lon":127.01,"lat":37.29},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"base":"stations","ma
in":{"temp":271.31,"pressure":1028,"humidity":38,"temp_min":270.15,"temp_max":272.25},"visibility":10000,"wind":{"speed":0.5,"deg":140
},"clouds":{"all":1},"dt":1546767000,"sys":{"type":1,"id":5509,"message":0.0057,"country":"KR","sunrise":1546728386,"sunset":154676334
9},"id":1835553,"name":"Suigen","cod":200}

```

## 날씨 정보 구성

```
{
  "coord": {"lon": 126.98, "lat": 37.57},
  "weather": [{"id": 803, "main": "Clouds", "description": "broken clouds", "icon": "04d"}],
  "base": "stations",
  "main": {"temp": 275.74, "pressure": 1026, "humidity": 40, "temp_min": 274.15, "temp_max": 277.15},
  "visibility": 10000,
  "wind": {"speed": 1.5, "deg": 340},
  "clouds": {"all": 75},
  "dt": 1543984200,
  "sys": {"type": 1, "id": 5509, "message": 0.0046, "country": "KR", "sunrise": 1543962690, "sunset": 1543997613},
  "id": 1835848,
  "name": "Seoul",
  "cod": 200
}
```

- Coord(좌표) lon : 경도, lat : 위도
- Weather
- Base
- Main(main정보) temp : 온도, pressure : 기압
- Wind(바람) speed : 바람 세기, deg : 바람 방향
- Clouds(구름)
- Dt(시간)
- Name(도시 이름)

## 필요한 날씨정보 파싱

### ❖ 파싱이란?



[출처] <https://terms.naver.com/entry.nhn?docId=2454969&cid=42346&categoryId=42346>

- 즉, 여러 개의 데이터 중 필요한 데이터만 골라 내는 것
- 현재기온, 최고기온, 최저기온, 습도 데이터만 파싱



## 필요한 날씨정보 파싱

## ❖ main.c

현재 기온 (temp)의 값을  
배열(temp\_arr)에 저장

습도 (humidity)의 값을  
배열(humi\_arr)에 저장

최저 기온 (temp\_min)의 값을  
배열(tmin\_arr)에 저장

최고 기온 (temp\_max)의 값을  
배열(tmax\_arr)에 저장

배열 값을 숫자로 변환 후 출력  
(온도) 절대온도 → 섭씨온도

```

Project  Build  Samsung ARTIK  WINDOW  Help
-----
weather_api.h  c weather_api.c  *main.c  string.h
85  lwip_read(sockfd, str, MAX_DATA_SIZE);
86  printf("\nFull data : %s\n\n", str); //Use api by parsing this string
87
88  // Parsing
89  temp_offset = strstr(str, "temp") + strlen("temp") + DATA_OFFSET;
90  for(i = 0 ; i < TEMP_LEN-1 ; i++)
91  {
92      if (*(temp_offset+(TEMP_LEN-1))==",") break;
93      temp_arr[i] = *(temp_offset+i);
94  }
95
96  humi_offset = strstr(str, "humidity") + strlen("humidity") + DATA_OFFSET;
97  for(i = 0 ; i < HUMI_LEN-1 ; i++)
98  {
99      if (*(humi_offset+(HUMI_LEN-1))==",") break;
100     humi_arr[i] = *(humi_offset+i);
101 }
102
103 tmin_offset = strstr(str, "temp_min") + strlen("temp_min") + DATA_OFFSET;
104 for(i = 0 ; i < TEMP_LEN-1 ; i++)
105 {
106     if (*(tmin_offset+(TEMP_LEN-1))==",") break;
107     tmin_arr[i] = *(tmin_offset+i);
108 }
109
110 tmax_offset = strstr(str, "temp_max") + strlen("temp_max") + DATA_OFFSET;
111 for(i = 0 ; i < TEMP_LEN-1 ; i++)
112 {
113     if (*(tmax_offset+(TEMP_LEN-1))==",") break;
114     tmax_arr[i] = *(tmax_offset+i);
115 }
116 printf("\n*** The Weather of Suwon(suigen) ***\n");
117 printf("| Current Temp. : %.2f degree\n", atof(temp_arr)-ABS_TEMP);
118 printf("| Maximum Temp. : %.2f degree\n", atof(tmax_arr)-ABS_TEMP);
119 printf("| Minimum Temp. : %.2f degree\n", atof(tmin_arr)-ABS_TEMP);
120 printf("| Humidity : %.2f%% \n", atof(humi_arr));
121 printf("*****\n");
122

```

## 필요한 날씨정보 파싱

## ❖ main.c

현재 기온 (temp)의 값을  
배열(temp\_arr)에 저장

습도 (humidity)의 값을  
배열(humi\_arr)에 저장

최저 기온 (temp\_min)의 값을  
배열(tmin\_arr)에 저장

최고 기온 (temp\_max)의 값을  
배열(tmax\_arr)에 저장

배열 값을 숫자로 변환 후 출력  
(온도) 절대온도 → 섭씨온도

```

Project  Build  Samsung ARTIK  WINDOW  Help
-----
weather_api.h  c weather_api.c  *main.c  string.h
85  lwip_read(sockfd, str, MAX_DATA_SIZE);
86  printf("\nFull data : %s\n\n", str); //Use api by parsing this string
87
88  // Parsing
89  temp_offset = strstr(str, "temp") + strlen("temp") + DATA_OFFSET;
90  for(i = 0 ; i < TEMP_LEN-1 ; i++)
91  {
92      if (*(temp_offset+(TEMP_LEN-1))==",") break;
93      temp_arr[i] = *(temp_offset+i);
94  }
95
96  humi_offset = strstr(str, "humidity") + strlen("humidity") + DATA_OFFSET;
97  for(i = 0 ; i < HUMI_LEN-1 ; i++)
98  {
99      if (*(humi_offset+(HUMI_LEN-1))==",") break;
100     humi_arr[i] = *(humi_offset+i);
101 }
102
103 tmin_offset = strstr(str, "temp_min") + strlen("temp_min") + DATA_OFFSET;
104 for(i = 0 ; i < TEMP_LEN-1 ; i++)
105 {
106     if (*(tmin_offset+(TEMP_LEN-1))==",") break;
107     tmin_arr[i] = *(tmin_offset+i);
108 }
109
110 tmax_offset = strstr(str, "temp_max") + strlen("temp_max") + DATA_OFFSET;
111 for(i = 0 ; i < TEMP_LEN-1 ; i++)
112 {
113     if (*(tmax_offset+(TEMP_LEN-1))==",") break;
114     tmax_arr[i] = *(tmax_offset+i);
115 }
116
117 printf("\n*** The Weather of Suwon(suigen) ***\n");
118 printf("| Current Temp. : %.2f degree\n", atof(temp_arr)-ABS_TEMP);
119 printf("| Maximum Temp. : %.2f degree\n", atof(tmax_arr)-ABS_TEMP);
120 printf("| Minimum Temp. : %.2f degree\n", atof(tmin_arr)-ABS_TEMP);
121 printf("| Humidity : %.2f%% \n", atof(humi_arr));
122 printf("*****\n");

```

## 필요한 날씨정보 파싱

## ❖ parsing algorithm(1)

```
temp_offset = strstr(str, "temp") + strlen("temp") + DATA_OFFSET;
for(i = 0 ; i < TEMP_LEN-1 ; i++)
{
    if (*(temp_offset+(TEMP_LEN-1)) != ",") break;
    temp_arr[i] = *(temp_offset+i);
}
```

strstr()함수 : 문자열 찾는 함수  
str문자열 중 temp를 포함하고  
있는 위치의 포인터 반환

temp 길이만큼 포인터 이동

DATA\_OFFSET = 2 이므로  
" 와 : 문자열을 무시(포인터 이동)

```
{ "coord": { "lon": 127.01, "lat": 37.
in": { "temp": 27.31, "pressure": 10
}, "clouds": { "all": 1 }, "dt": 154678
9}, "id": 1835553, "name": "Suigen",
```



## 필요한 날씨정보 파싱

### ❖ parsing algorithm(2)

```
temp_offset = strstr(str, "temp") + strlen("temp") + DATA_OFFSET;  
for(i = 0 ; i < TEMP_LEN-1 ; i++)  
{  
    if (*(temp_offset+(TEMP_LEN-1))==" ,") break;  
    temp_arr[i] = *(temp_offset+i);  
}
```

결국, temp\_offset은 temp  
데이터의 첫 값인 2를 가리키는 포인터

다음 , 를 만나기 전까지  
배열에 값 저장

```
{ "coord": { "lon": 127.01, "lat": 37.  
in": { "temp" 271.31, "pressure": 10  
}, "clouds": { "all": 1 }, "dt": 154678  
9}, "id": 1835553, "name": "Suigen",
```

## Reference

---

- ❖ Weather API 사용법(<http://junolefou.tistory.com/1>)
- ❖ Json 파싱(직접 파싱) (<https://dojang.io/mod/page/view.php?id=724>)

DSL

**Thank you**

DSL<sup>LAB</sup>