IoT 플랫폼 활용 기초 교육

2019.02.19 ~2019.02.22

목차

III. ARTIK Cloud

5. 외부 API 사용하기

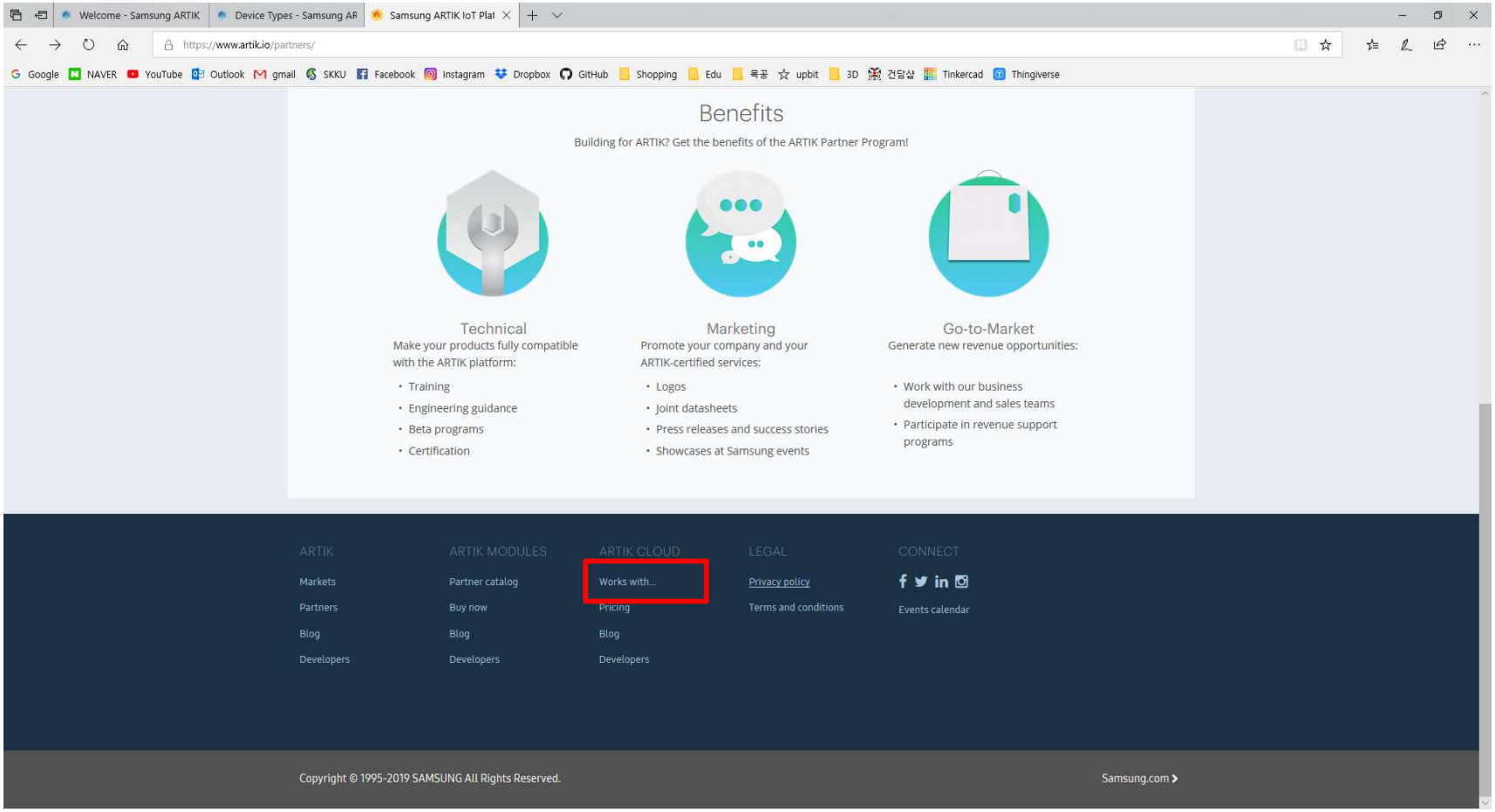
5. 외부 API 이용하기

- Weather API -

Login to Open Weather Map (1)

❖ Open Weather Map에 접속하기

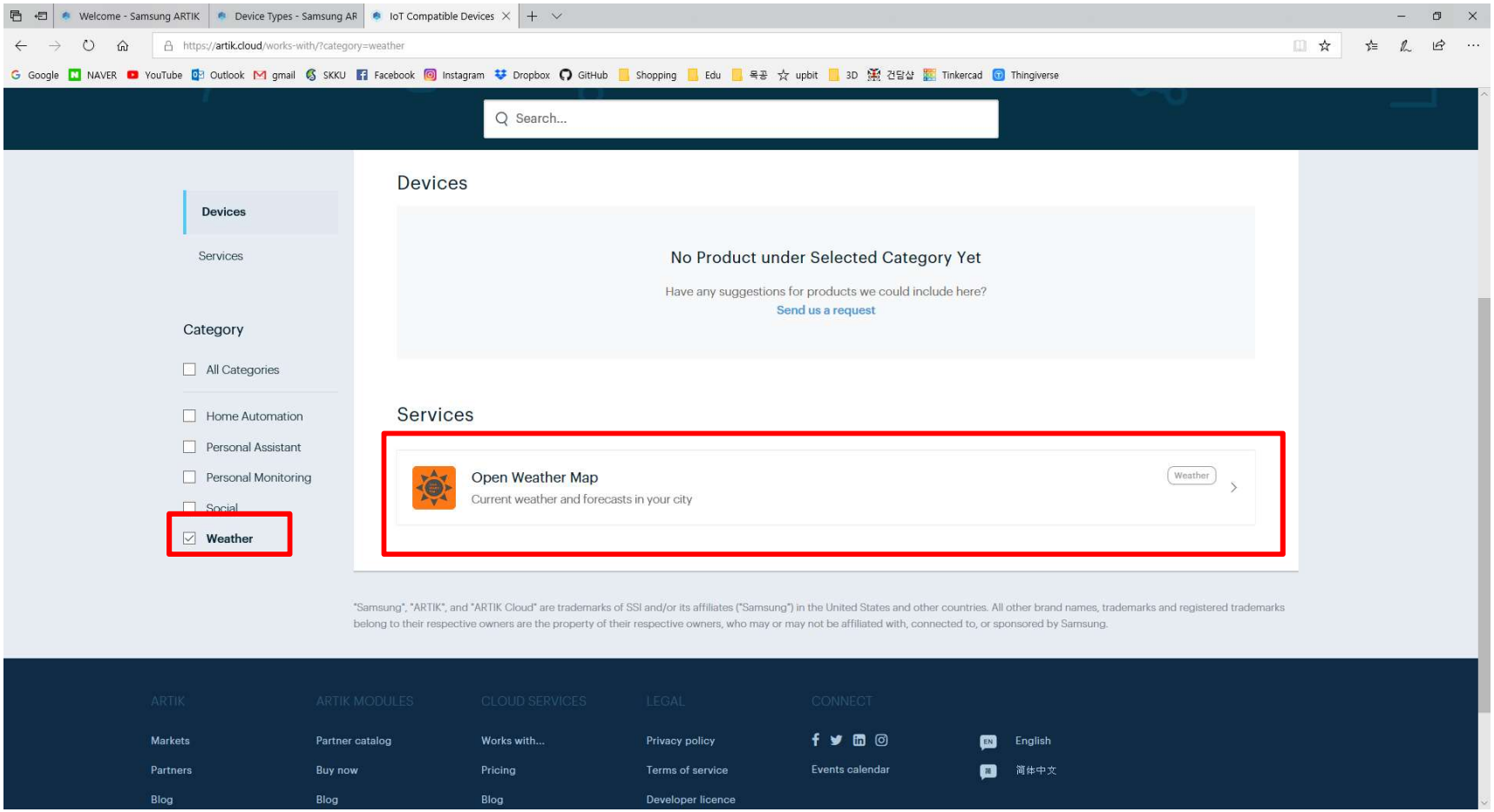
➢ ARTIK Cloud 맨 아래에 [Work with...] 클릭



Login to Open Weather Map (2)

❖ ARTIK Cloud에서 지원하는 서비스

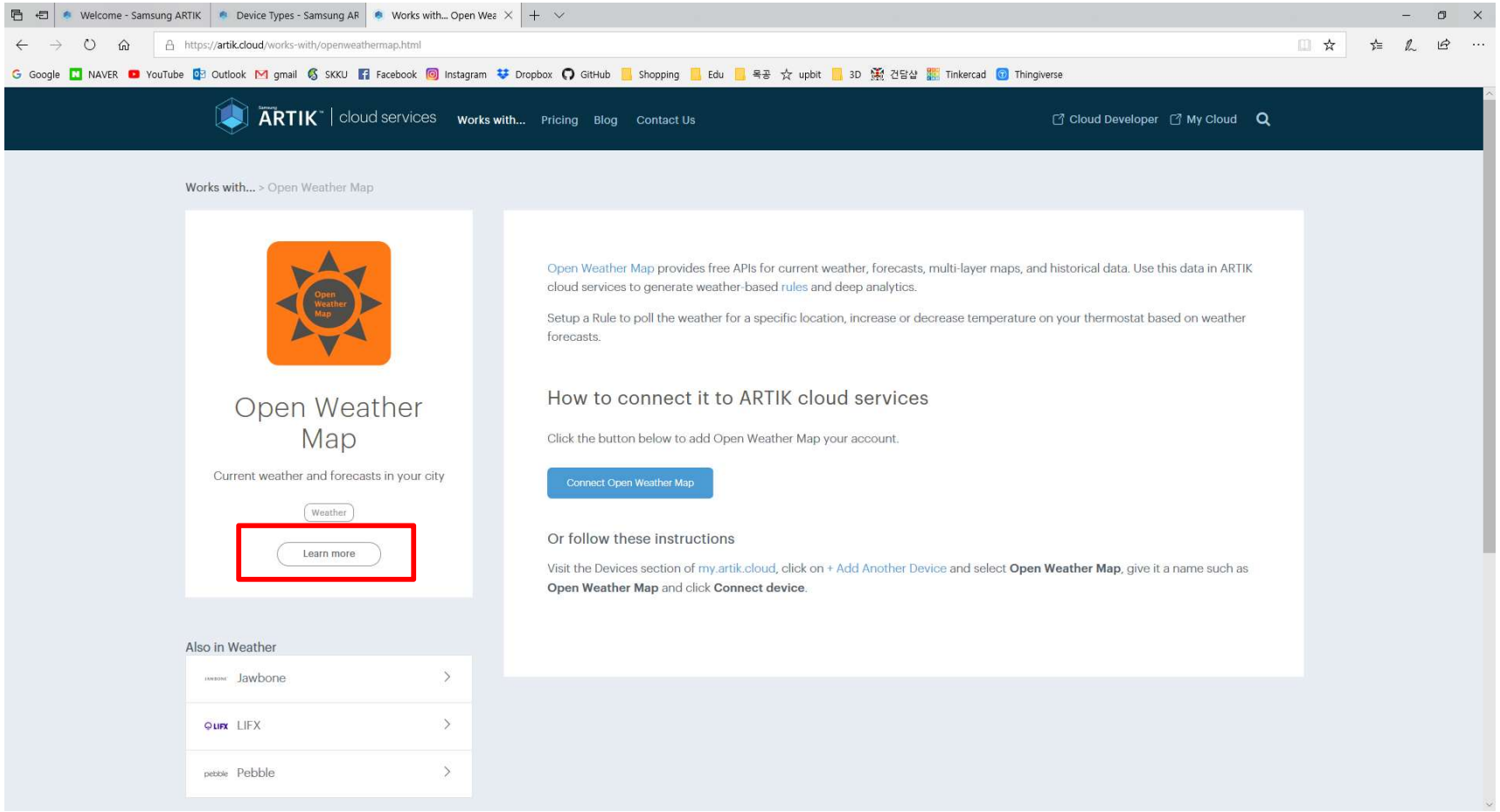
➢ 다양한 서비스 중 [Weather]를 선택하고 Services 중 Open Weather Map 클릭



Login to Open Weather Map (3)

❖ Open Weather Map에 접속

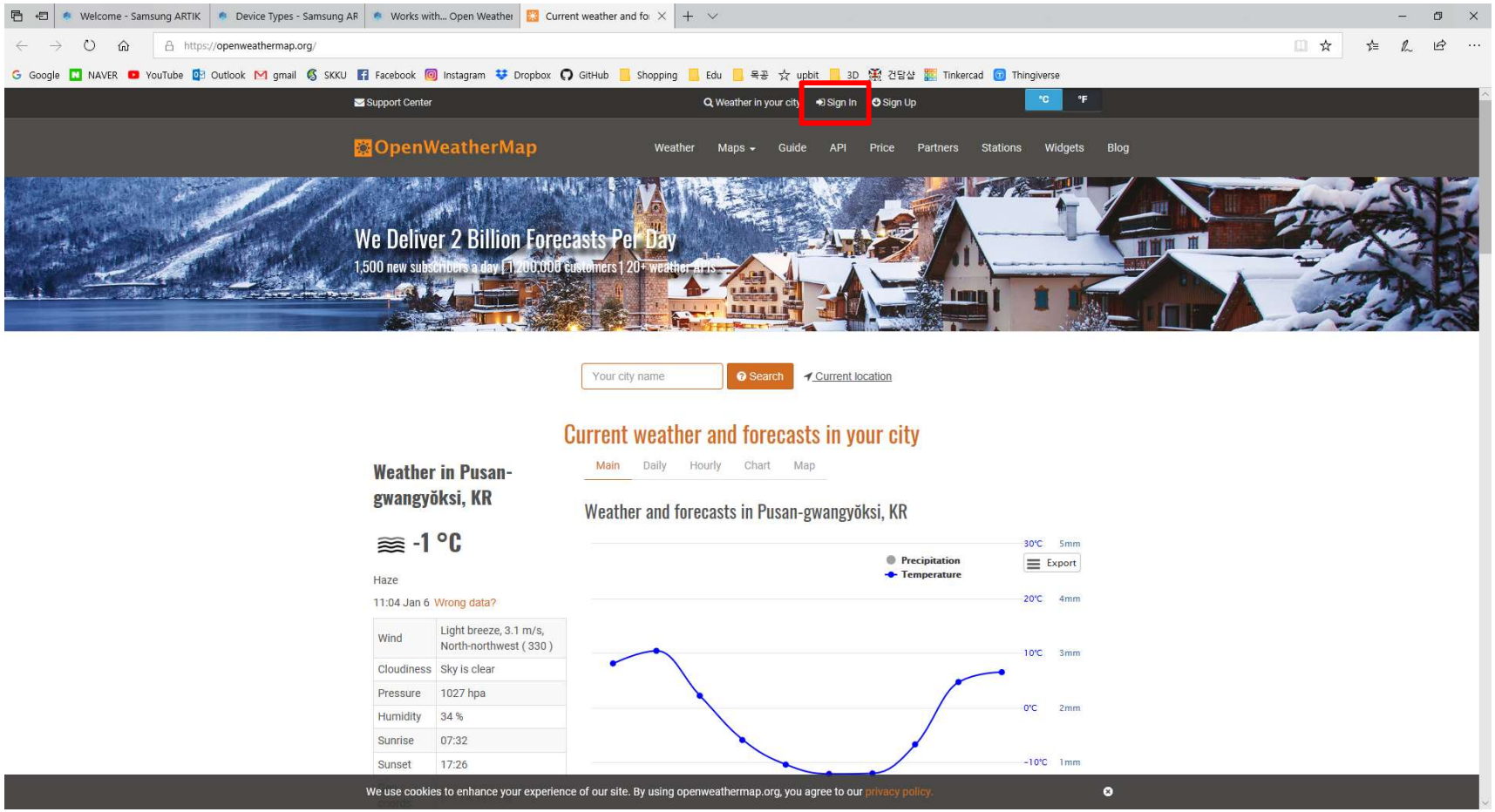
➢ [learn more]을 클릭하여 Open Weather Map 사이트에 접속



Login to Open Weather Map (4)

❖ Open Weather Map에 회원가입

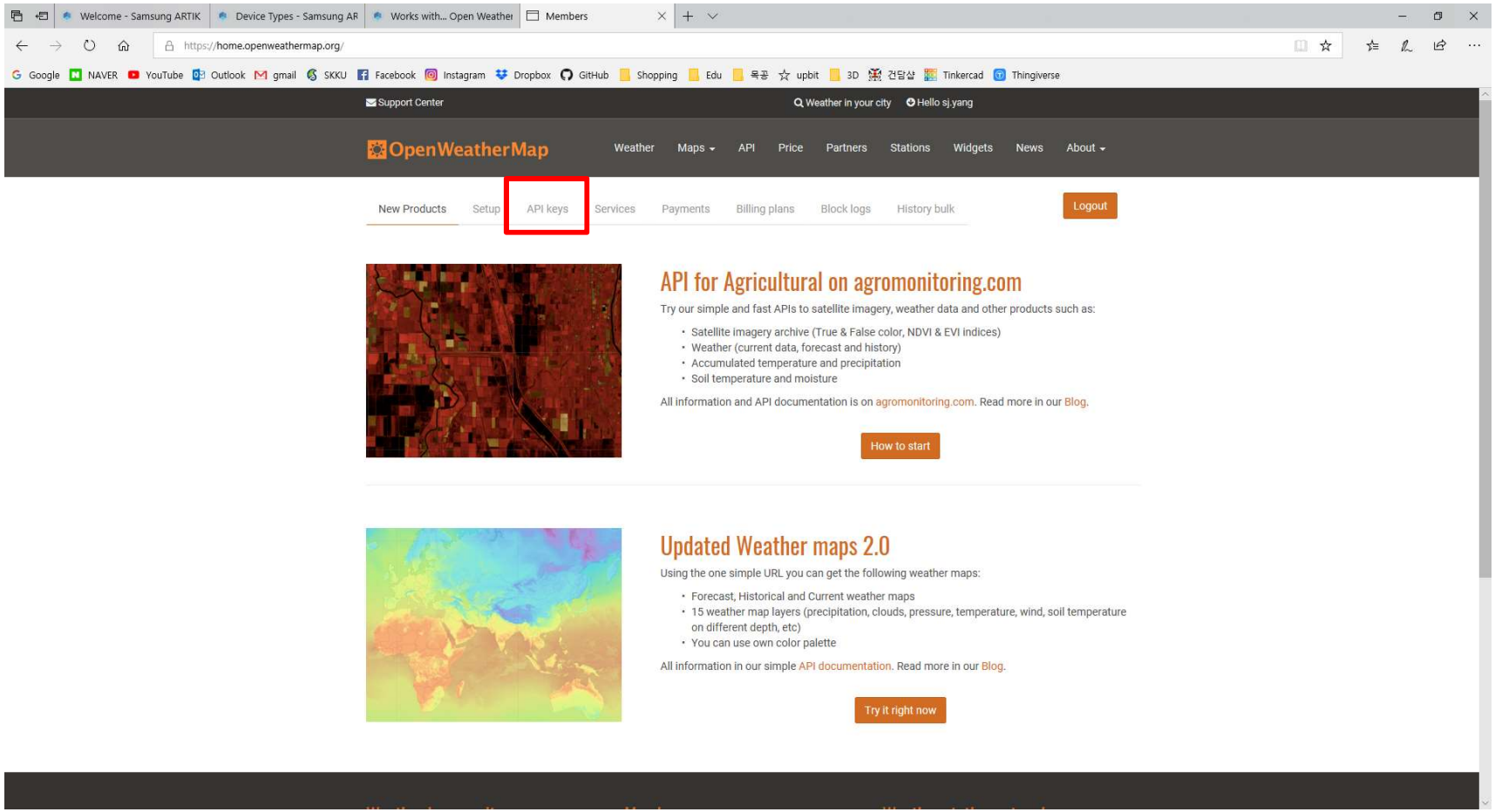
➢ sign-in을 눌러 회원가입 후 로그인 (기존 회원일 경우 스킵)



Login to Open Weather Map (5)

❖ login 후 API Key 받기

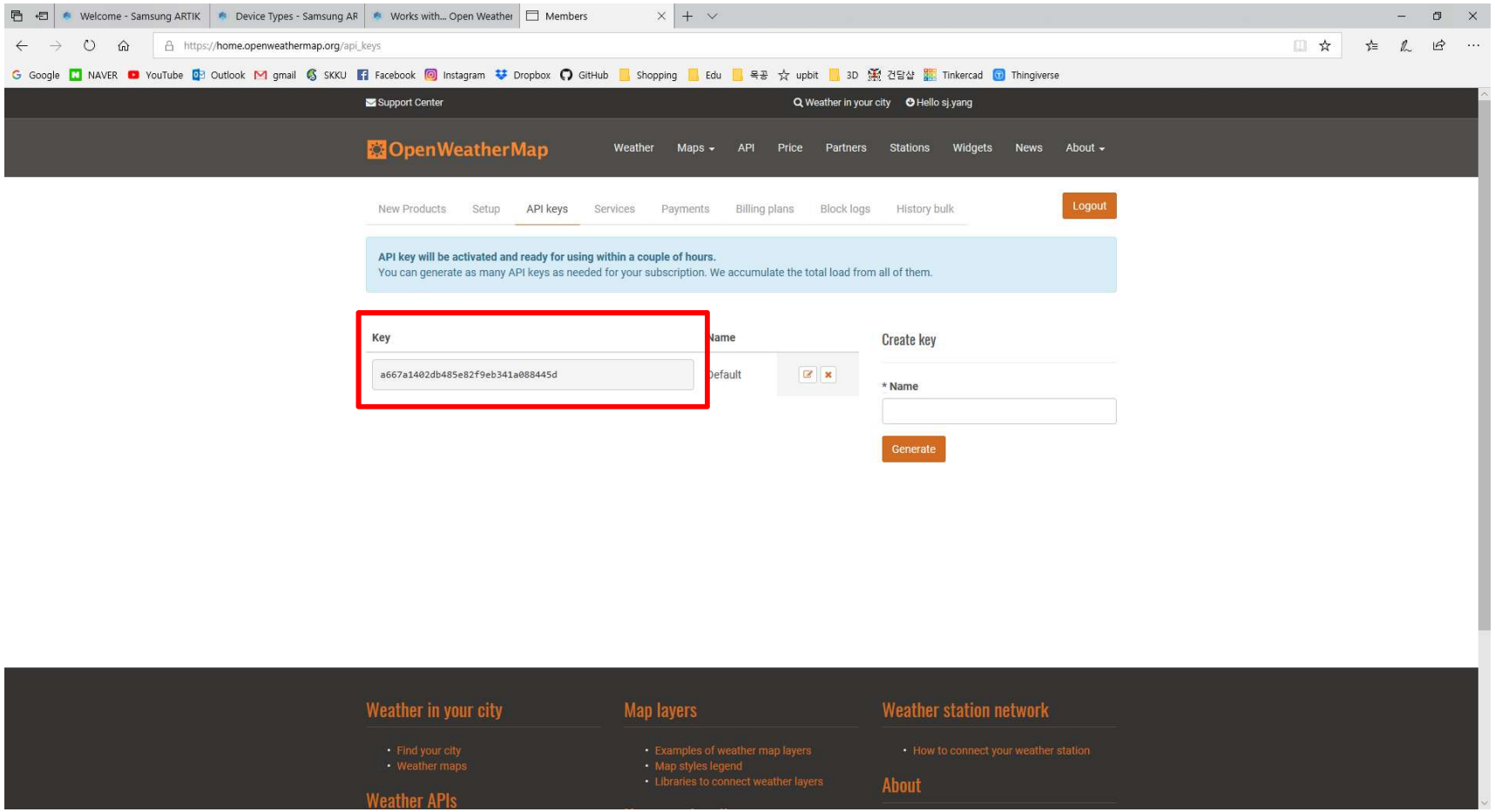
➢ 메뉴의 [API Keys]를 클릭하여 API Key 확인



Login to Open Weather Map (6)

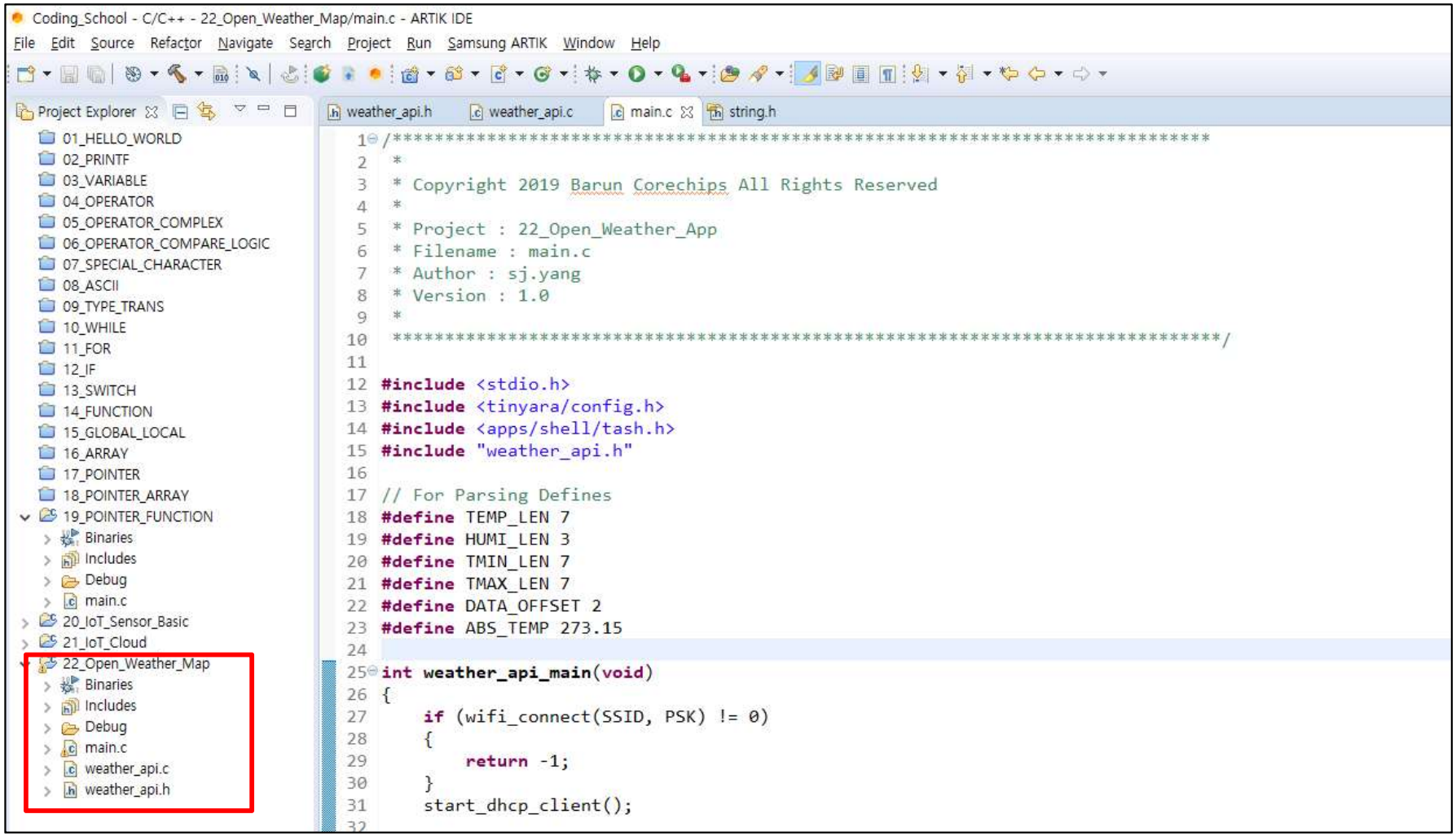
❖ API Key 받기

➢ 몇 분에서 몇시간 정도 지난 후 사용 가능



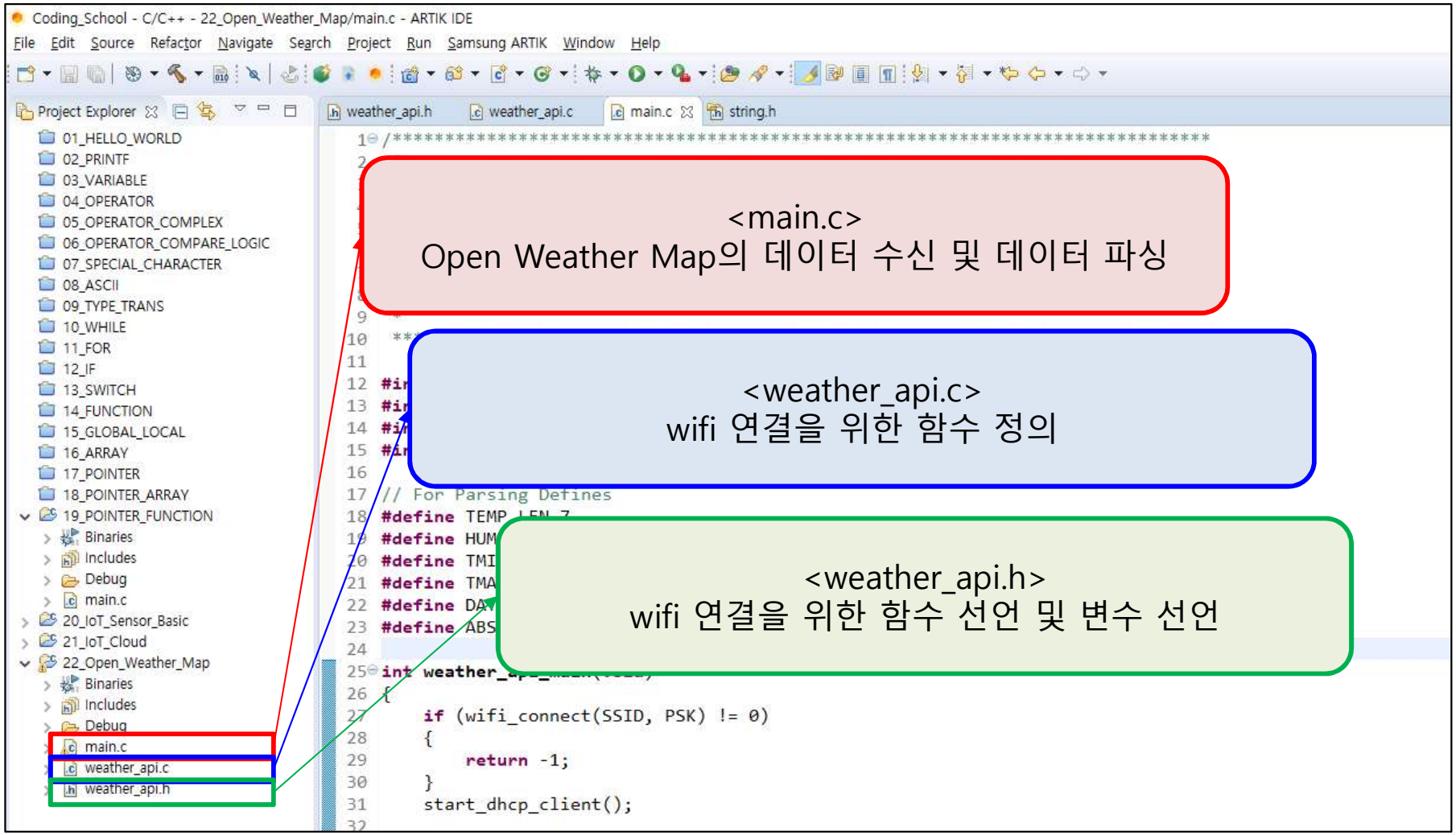
Source Code Flash

❖ 22_Open_Weather_Map 프로젝트를 ARTIK에 flash



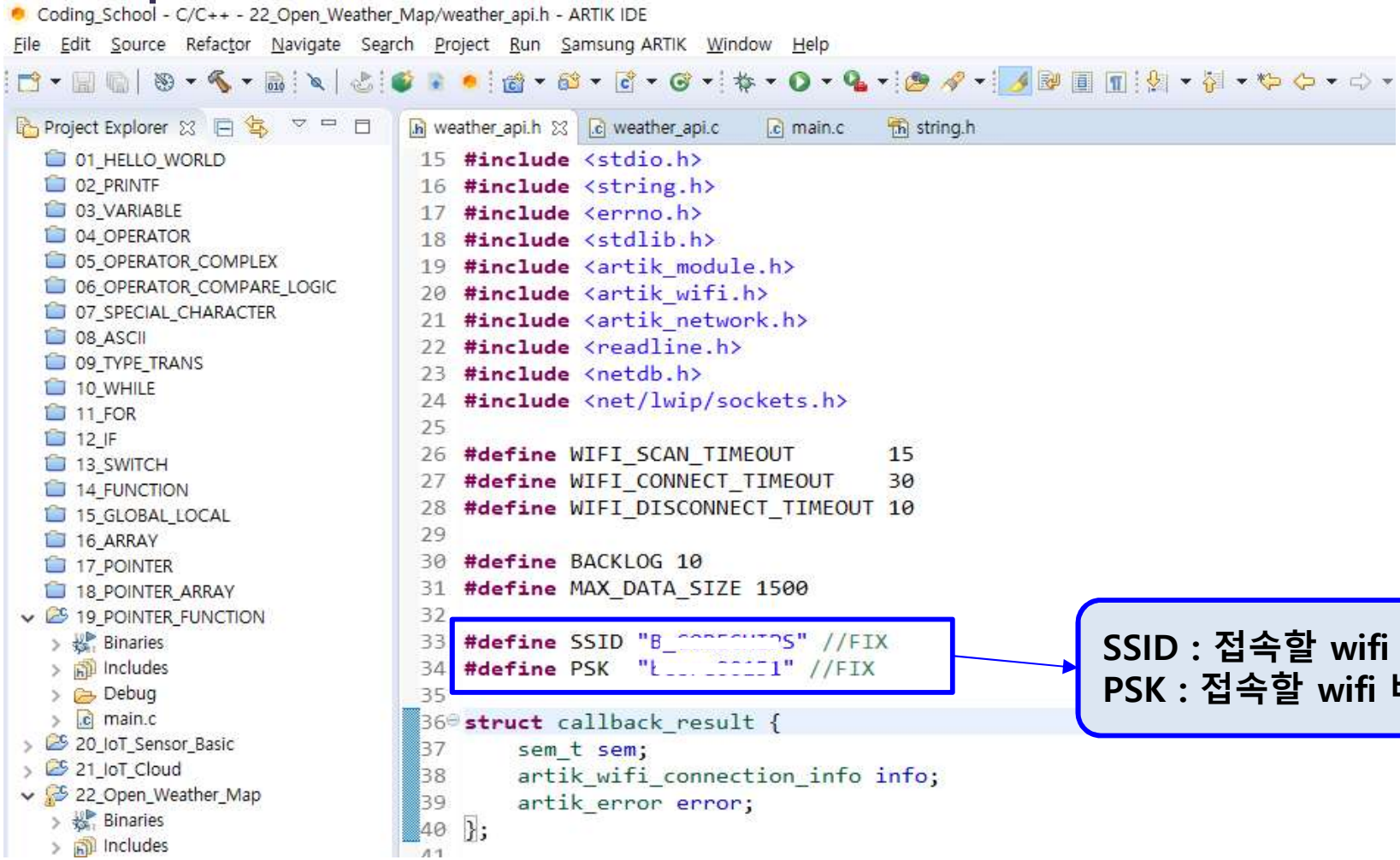
Source Code Flash

❖ Source code files



Source Code 설정

❖ weather_api.h



Source Code 설정

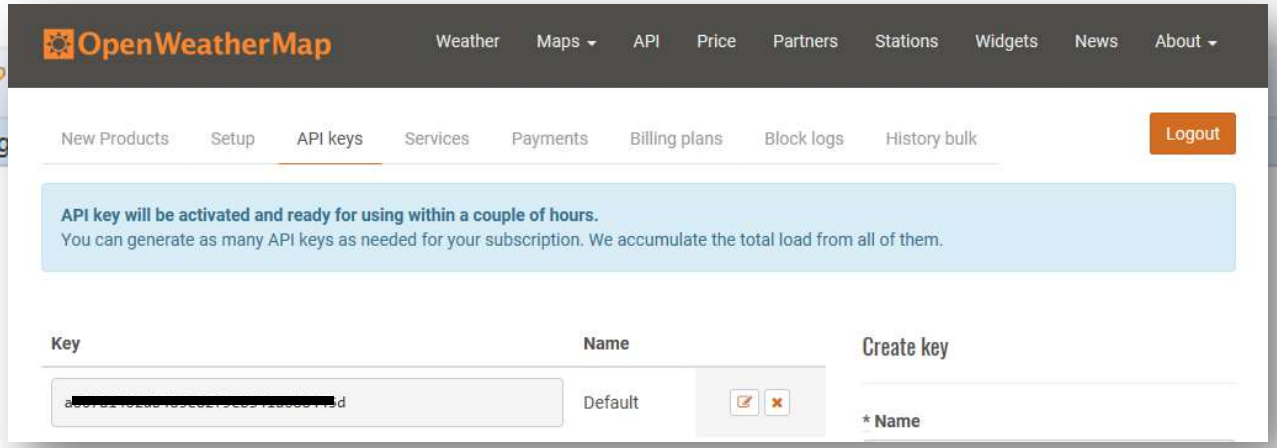
❖ main.c

ap/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=a6[redacted]45d HTTP/1.0\r\n\r\n";
40     char *host = "api.openweathermap.org";  //weather api
```

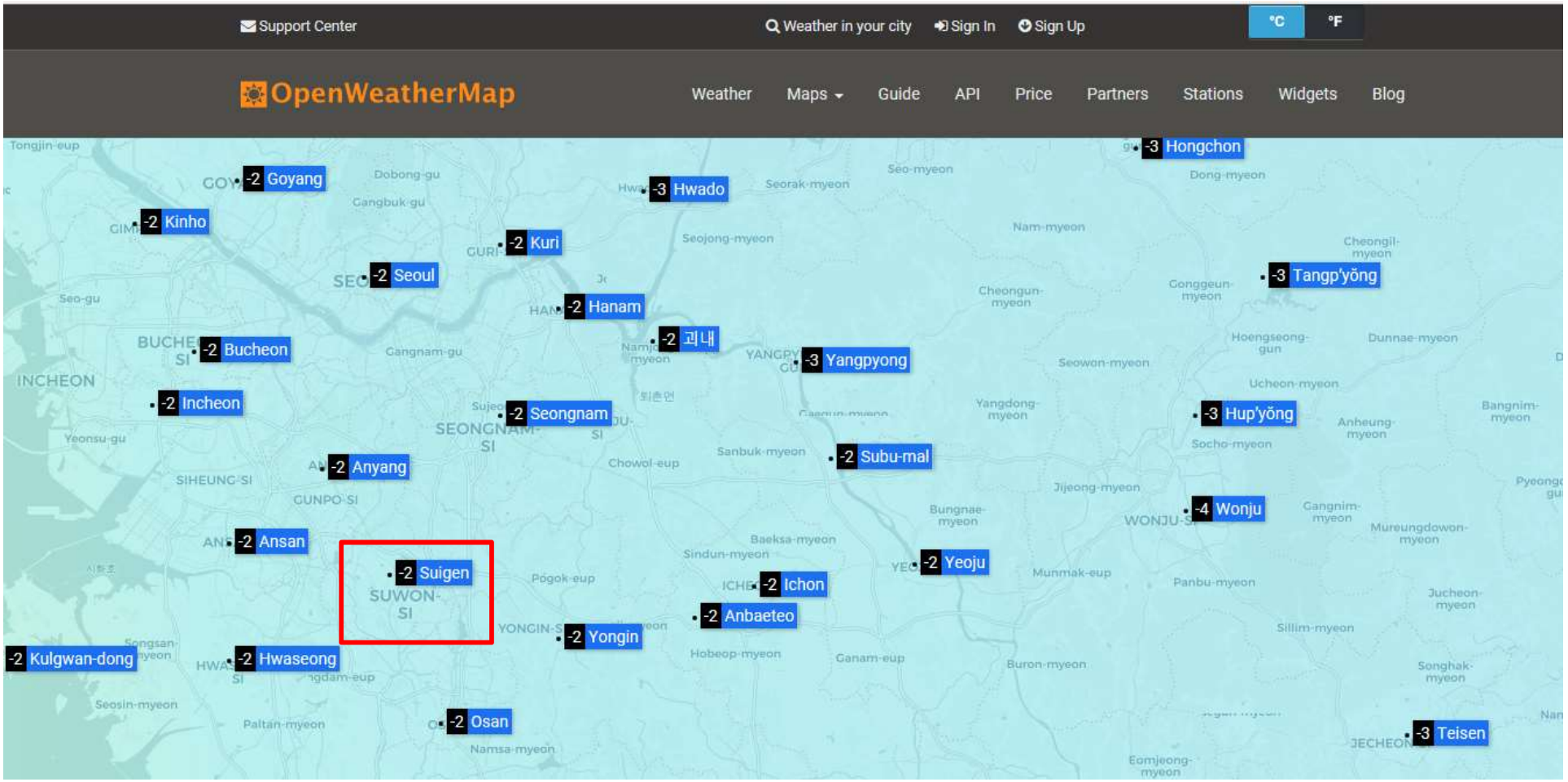


도시 이름(수원 = suigen)

앞에서 확인한 API Key 입력

지역에 따른 지역명

❖ Open Weather Map에 각 지역별로 지역이름 확인 가능



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","main":{"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":1546763349},"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?

IT용어사전

파싱

[parsing , 文章-分析]

컴퓨터에서 컴파일러 또는 번역기가 원시 부호를 기계어로 번역하는 과정의 한 단계로, 각 문장의 문법적인 구성 또는 구문을 분석하는 과정. 즉 원시 프로그램에서 나타난 토큰(token)의 열을 받아들여 이를 그 언어의 문법에 맞게 구문 분석 트리(parse tree)로 구성해 내는 일이다. 파싱은 크게 하향식 파싱과 상향식 파싱으로 나눌 수 있다.

[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

```
weather_api.h  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;
00          humi_arr[i] = *(humi_offset+i);
01      }
02
03      tmin_offset = strstr(str,"temp_min") + strlen("temp_min") + DATA_OFFSET;
04      for(i = 0 ; i < TEMP_LEN-1 ; i++)
05      {
06          if (*(tmin_offset+(TEMP_LEN-1))==",") break;
07          tmin_arr[i] = *(tmin_offset+i);
08      }
09
10      tmax_offset = strstr(str,"temp_max") + strlen("temp_max") + DATA_OFFSET;
11      for(i = 0 ; i < TEMP_LEN-1 ; i++)
12      {
13          if (*(tmax_offset+(TEMP_LEN-1))==",") break;
14          tmax_arr[i] = *(tmax_offset+i);
15      }
16      printf("\n*** The Weather of Suwon(suigen) ***\n");
17      printf("| Current Temp. : %.2f degree\n",atof(temp_arr)-ABS_TEMP);
18      printf("| Maxinum Temp. : %.2f degree\n",atof(tmax_arr)-ABS_TEMP);
19      printf("| Minimum Temp. : %.2f degree\n",atof(tmin_arr)-ABS_TEMP);
20      printf("|   Humidity    : %.2f%% \n",atof(humi_arr));
21      printf("************************************\n");
```

필요한 날씨정보 파싱

❖ main.c

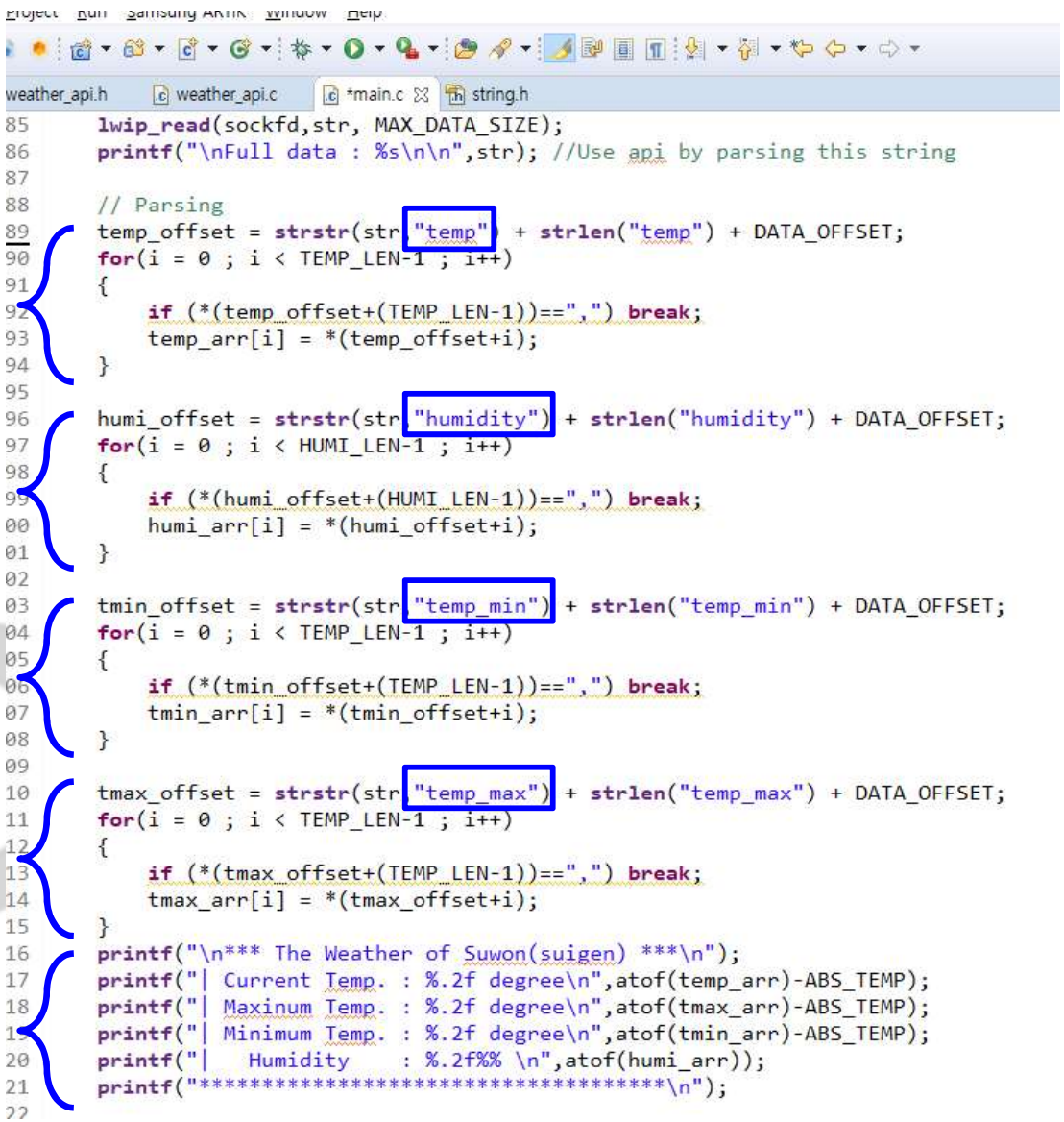
현재 기온 (temp)의 값을 배열(temp_arr)에 저장

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




```
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;
00          humi_arr[i] = *(humi_offset+i);
01      }
02
03      tmin_offset = strstr(str,"temp_min") + strlen("temp_min") + DATA_OFFSET;
04      for(i = 0 ; i < TEMP_LEN-1 ; i++)
05      {
06          if (*(tmin_offset+(TEMP_LEN-1))==",") break;
07          tmin_arr[i] = *(tmin_offset+i);
08      }
09
10      tmax_offset = strstr(str,"temp_max") + strlen("temp_max") + DATA_OFFSET;
11      for(i = 0 ; i < TEMP_LEN-1 ; i++)
12      {
13          if (*(tmax_offset+(TEMP_LEN-1))==",") break;
14          tmax_arr[i] = *(tmax_offset+i);
15      }
16      printf("\n*** The Weather of Suwon(suigen) ***\n");
17      printf("| Current Temp. : %.2f degree\n",atof(temp_arr)-ABS_TEMP);
18      printf("| Maxinum Temp. : %.2f degree\n",atof(tmax_arr)-ABS_TEMP);
19      printf("| Minimum Temp. : %.2f degree\n",atof(tmin_arr)-ABS_TEMP);
20      printf("|   Humidity    : %.2f%% \n",atof(humi_arr));
21      printf("************************************\n");
```

필요한 날씨정보 파싱

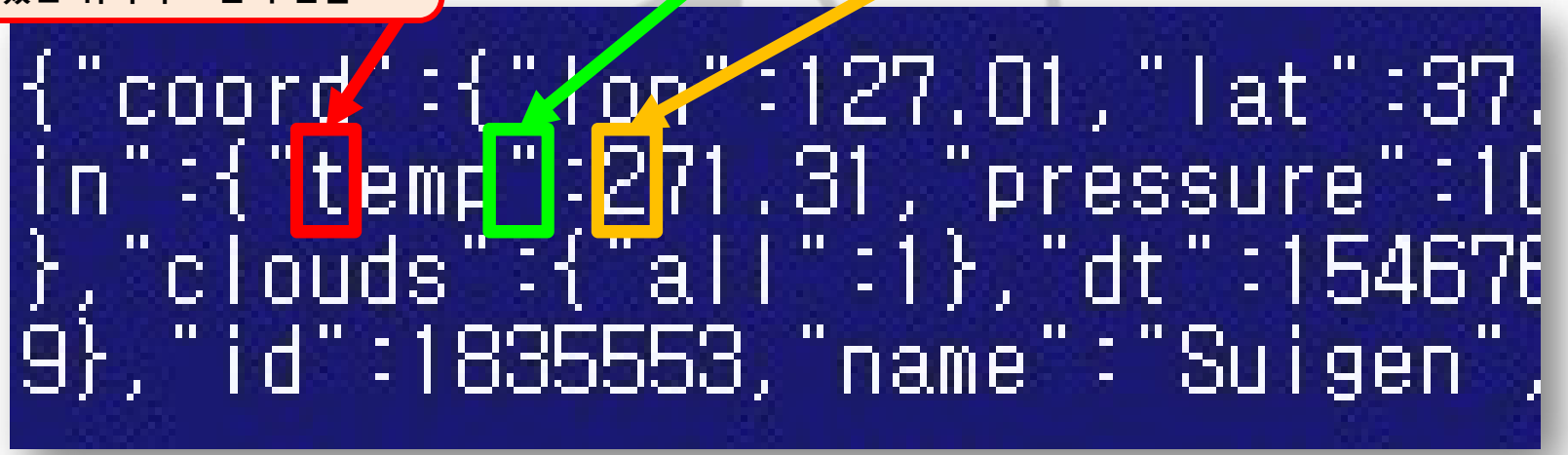
❖ parsing algorithm(1)

```
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)



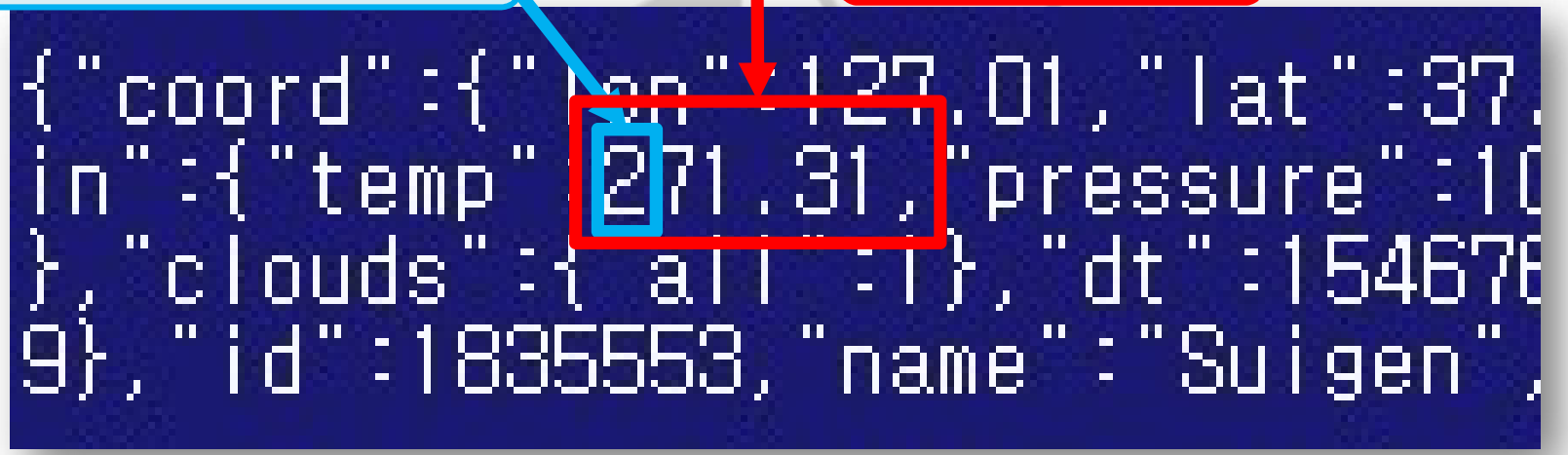
필요한 날씨정보 파싱

❖ 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



Reference

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

Thank you