OpenWeather
- NodeMcu -

OpenWeather

❖ ex05/app.ino

```
#include <WifiMiniCom.h>
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>

const char *ssid = "Campus7_Room4_2.4GHz";
// const char *password = "12345678";
const char *password = NULL;

WifiMiniCom com;

String host = "api.openweathermap.org";
String url = "/data/2.5/weather?q=Seoul&APPID=93ec9acd67f5e6d7fd08ff43c857eeac&units=metric";

// 디폴트는 켈빈 온도
// 섭씨 온도 = 캘빈 온도 - 273.15
```

OpenWeather

❖ ex05/app.ino

```
bool request(WiFiClient &client) {
    const int port = 80;
    if (!client.connect(host, port))
    {
        Serial.println("connection failed");
        return false;
    }

    Serial.print("Requesting URL: ");
    Serial.println(url);

    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");
    return true;
}
```

OpenWeather

❖ ex05/app.ino

```
bool request(WiFiClient &client) {
    const int port = 80;
    if (!client.connect(host, port))
    {
        Serial.println("connection failed");
        return false;
    }

    Serial.print("Requesting URL: ");
    Serial.println(url);

    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");
    return true;
}
```

OpenWeather

❖ ex05/app.ino

```
void deserialize(String line) {
    // DynamicJsonDocument doc(800);
    StaticJsonDocument<800> doc;

    auto error = deserializeJson(doc, line);
    if (error) {
        Serial.print("deserializeJson() failed with code ");
        Serial.println(error.c_str());
        return;
    }

    String w_main = doc["weather"][0]["main"];
    double temp = doc["main"]["temp"].as<double>();
    double humi = doc["main"]["humidity"].as<double>();
    String msg = "weather: " + w_main;
    com.print(0, msg.c_str(), true);
    msg = String("T:") + temp + ", H:" + humi;
    com.print(1, msg.c_str(), true);
}
```

OpenWeather

❖ ex05/app.ino

```
String response(WiFiClient &client) {

    int timeout = millis() + 5000;

    while (client.available() == 0)  {
        if (timeout - millis() < 0)  {
            Serial.println(">>> Client Timeout !");
            client.stop();
            return "";
        }
    }

    bool isBody = false;
    String body ="";
```

OpenWeather

❖ ex05/app.ino

```
    while(client.available()) {
        String line = client.readStringUntil('\r');
        line.trim();
        if(line == "") {
            isBody = true;
            continue;
        }
        Serial.println(line);
        if(isBody) {
            body = line;
            break;
        }
    }
    Serial.println();
    Serial.println("closing connection");
    return body;
}
```

OpenWeather

❖ ex05/app.ino

```
void setup() {
    com.init(ssid, password);


}

void loop(){
    WiFiClient client;
    if(request(client)) {
        String body = response(client);
        deserialize(body);
    }
    delay(100000);
}
```