

UNIVERSITY OF SCIENCE - VNUHCM

Faculty of Information Technology

INTERNET OF THINGS

3.3

ESP8266 IS A CLIENT



The Basic Client

GET CONTENT OF A WEBSITE

```
const char* host = "www.example.com";
const int port = 80;
const char* request = "/";

void sendRequest() {
    WiFiClient client;
    while(!client.connect(host, port)) {
        Serial.println("connection fail");
        delay(1000);
    }
    client.print(String("GET ") + request + " HTTP/1.1\r\n"
        + "Host: " + host + "\r\n"
        + "Connection: close\r\n\r\n");
    delay(500);

    while(client.available()) {
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }
}

void setup() {
    Serial.begin(9600);
    Serial.print("Connecting to WiFi");
    wifiConnect();
    sendRequest();
}
```



if  this then that

Want to build your own service? Build on the platform [↗](#)

Home Explore Create Learn ▼



Step 1: Sign up new account in **ifttt.com**

Step 2: Sign in

Step 3: Select menu Create

Create your own



If This

Add

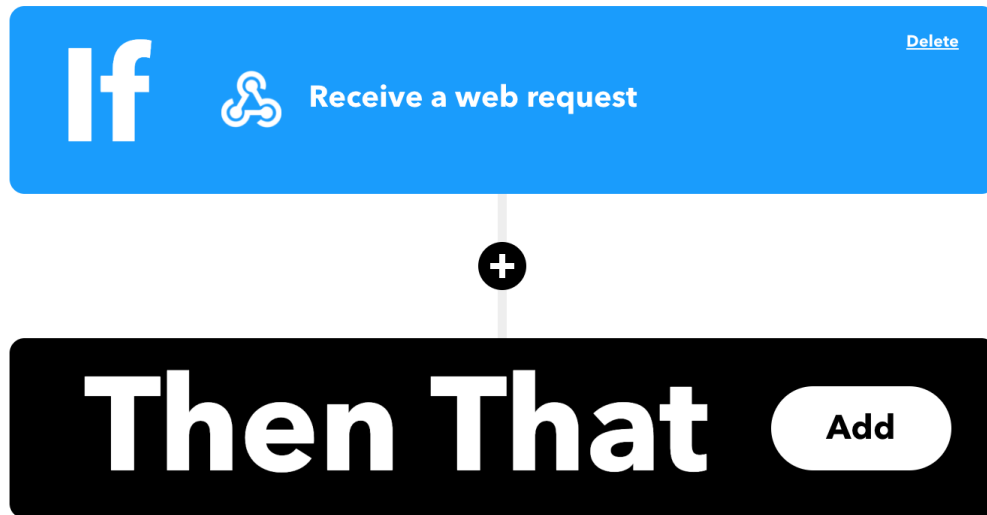
Then That

Step 4: Select button **Add (If this)**

Step 5: Search “**Webhooks**” > Select **WebHooks** in the result list

Step 6: Select **Receive a web request** >

Enter **Event Name** > Create trigger



Step 7: Select **button Add (Then That)**

Step 8: Search “**Notifications**” > Select **Notifications** in the result list

Step 9: Select **Send a notification from IFTTT app** > Change Message > Create action

Webhooks Settings

View activity log

Account Info

Edit

Connected as

cxnam

URL
Status

https://maker.ifttt.com/use/PuIEpkBl5flrMP51qo7fX
Active

↑
Host

↑
Key ID

Step 10: Select Avatar > My Services > WebHooks > Settings

Step 11: Copy host + key Id



Press button + Send notification

Hint:

Host: maker.ifttt.com

URL: `/trigger/{{event_name}}/with/key/{{your_key}}`



Press button +
Send Email

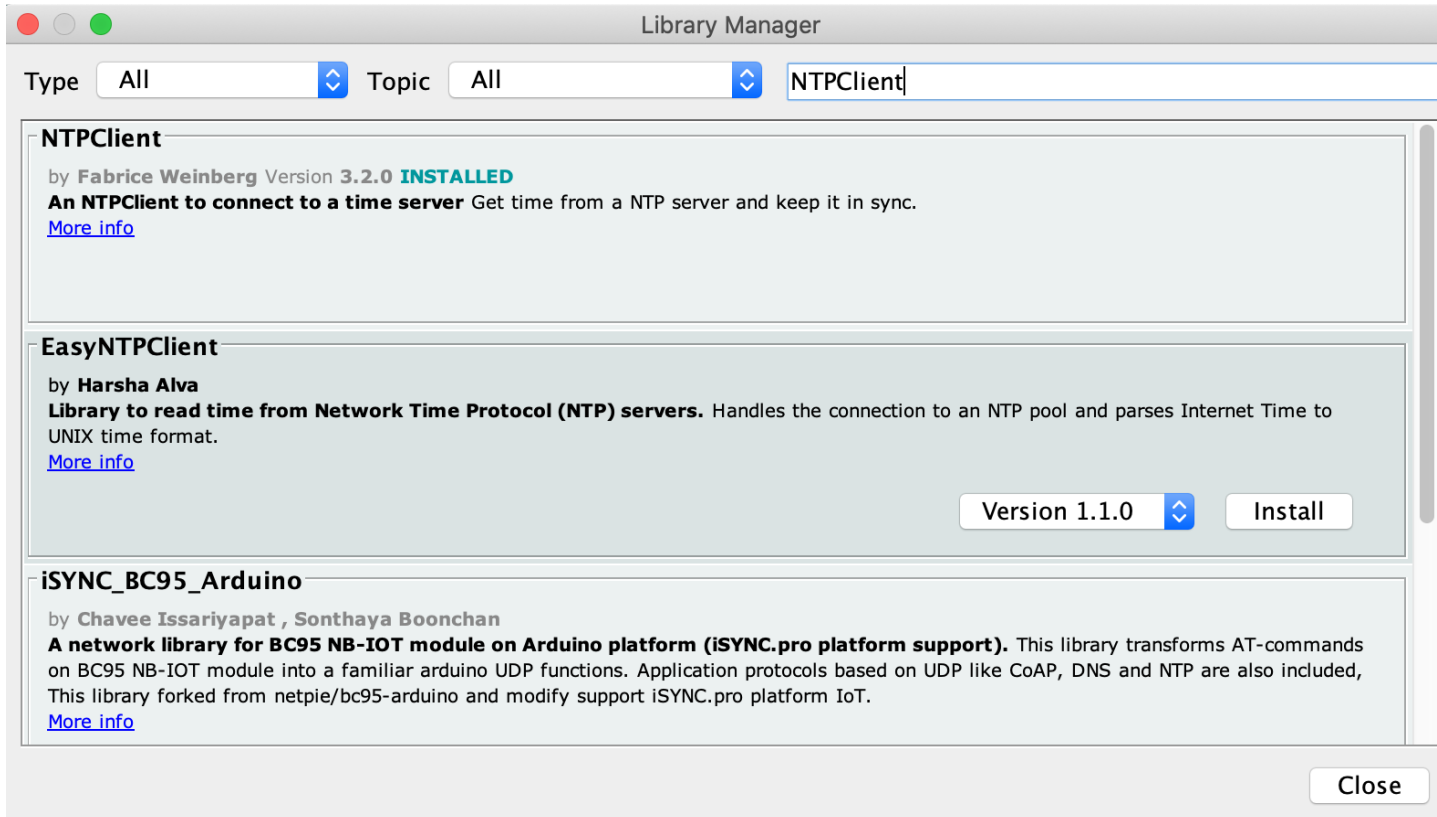


Press button +
Facebook Page

NTP Server

NETWORK TIME PROTOCOL

NTP stands for Network Time Protocol, and it is an Internet protocol used to synchronize the clocks of computers to some time reference



Step 1: Open *Sketch*
> *Include Library* >
Manage Libraries...

Step 2: Search
“*NTPClient*” and
install the latest
version

```
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

char *ssid      = "your wifi name";
char *password  = "your wifi password";

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

void setup() {
    //Connect to WiFi Network. DIY


    timeClient.begin();
}

void loop() {
    timeClient.update();
    Serial.println(timeClient.getFormattedTime());
    delay(1000);
}
```



Alarm Clock

WEATHER FORECAST STATION







Weather in your city


Get StartedAPIPricingMapsPartnersBlogMarketplacecxnamis▼Support

New ProductsServicesAPI keysBilling plansPaymentsBlock logsMy ordersMy profile

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name
	Default  
	CXNam  

Create key

API key name 

Step 1: Sign up new account

Step 2: Sign in

Step 3: Go to **API keys > Use Default Key or Generate new key**

Step 4: Open Google Maps > Search location that you want to get forecast info

Step 5: Get **latitude**, **longitude** of location.

<https://www.google.com/maps/@10.7571534,106.6880955,15z>

Step 6: Modify this URL

<http://api.openweathermap.org/data/2.5/weather?lat=xxxxx&lon=yyyyy&units=metric&appid=zzzzz>

```
{ "coord": { "lon": 106.69, "lat": 10.76 }, "weather":  
[ { "id": 803, "main": "Clouds", "description": "broken  
clouds", "icon": "04d" } ], "base": "stations", "main":  
{ "temp": 33, "feels_like": 36.59, "temp_min": 33, "temp_max": 33, "pressure": 1003, "humidity": 59 },  
"visibility": 10000, "wind": { "speed": 3.1, "deg": 260 }, "clouds":  
{ "all": 75 }, "dt": 1591944869, "sys":  
{ "type": 1, "id": 9314, "country": "VN", "sunrise": 1591914651, "sunset": 1591960532 }, "timezone": 2  
5200, "id": 1566083, "name": "Ho Chi Minh City", "cod": 200 }
```



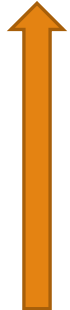
Get Temperature & Humidity using OpenWeatherMap

Hint:

Host = "api.openweathermap.org"

URL = "/data/2.5/weather?lat=xxxxx&lon=yyyyy&units=metric&appid=zzzzz"

```
String line = "";  
while (client.available()) {  
    line = client.readStringUntil('\n');  
}  
Serial.println(line);
```



Get last line of response

Step 1: Open *Sketch > Include Library > Manage Libraries...*

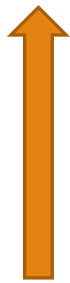
Step 2: Search “*ArduinoJson*” and install the latest version (v6.x)

Step 3: Include library in code

`#include <ArduinoJson.h>`

Step 4: Modify code to parse String to JSON

```
//create a json buffer where to store the json data
DynamicJsonDocument doc(1024);
DeserializationError error = deserializeJson(doc, line);
if (error)
    return;
int value = doc["main"]["temp"];
Serial.println(value);
```



Parse String to JSON Object



Get Temperature &
Humidity indoor
and outdoor

AJAX

```

void handleRoot() {
    String MAIN_page = R"=====(
    <!DOCTYPE html>
    <html>
        <head>
            <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
        </head>
        <body>
            <center>
                <div id="ext_temp"></div>
                <div id="ext_humidity"></div>
                <script>
                    $.ajax({url: "http://api.openweathermap.org/data/2.5/weather?lat=xxxxx&lon=yyyyy&units=metric&appid=zzzzz"
                    success: function(result){
                        $("#ext_temp").html(result.main.temp);
                        $("#ext_humidity").html(result.main.humidity);
                    }});
                </script>
            </center>
        </body>
    </html>
    )=====";

    server.send(200, "text/html", MAIN_page);
}

```