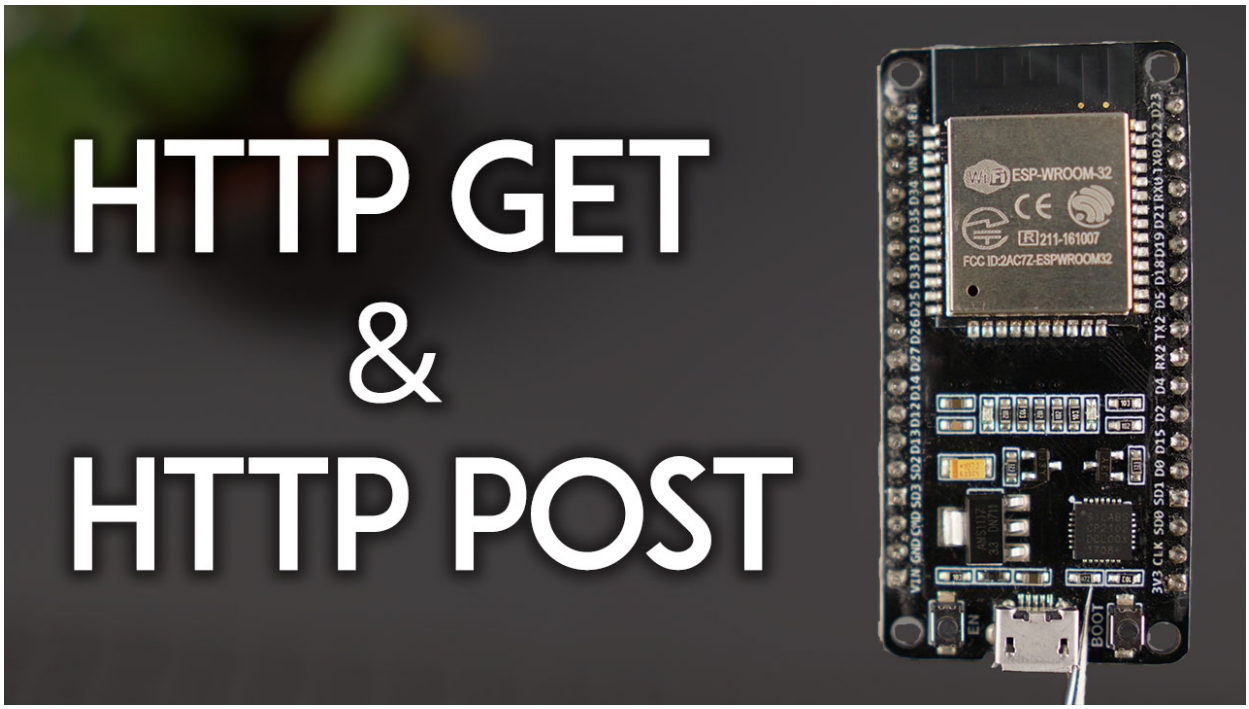
ESP32 HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text)

In this guide, you'll learn how to make HTTP GET and HTTP POST requests with the ESP32 board with Arduino IDE. We'll cover examples on how to get values, post JSON objects, URL encoded requests, and more.



Recommended: ESP8266 NodeMCU HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text)

HTTP Request Methods: GET vs POST

The Hypertext Transfer Protocol (HTTP) works as a request-response protocol between a client and server. Here's an example:

- The ESP32 (client) submits an HTTP request to a Raspberry Pi running Node-RED (server);
- The server returns a response to the ESP32 (client);
- Finally, the response contains status information about the request and may

HTTP GET

GET is used to request data from a specified resource. It is often used to get values from APIs.

For example, you can have:

```
GET /update-sensor?temperature=value1
```

Note that the query string (*name* = *temperature* and *value* = *value1*) is sent in the URL of the HTTP GET request.

Or you can use a simple request to return a value or JSON object, for example:

```
GET /get-sensor
```

(With HTTP GET, data is visible to everyone in the URL request.)

HTTP POST

POST is used to send data to a server to create/update a resource. For example, publish sensor readings to a server.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /update-sensor HTTP/1.1
Host: example.com
api_key=api&sensor_name=name&temperature=value1&humidity=value2&pressure=v
alue3
Content-Type: application/x-www-form-urlencoded
```

In the body request, you can also send a JSON object:

```
POST /update-sensor HTTP/1.1
Host: example.com
```

```
    value2, pressure: value3}
    Content-Type: application/json
```

(With HTTP POST, data is not visible in the URL request. However, if it's not encrypted, it's still visible in the request body.)

HTTP GET/POST with ESP32

In this guide, we'll explore the following scenarios:

1. ESP32 HTTP GET: Value or Query in URL
2. ESP32 HTTP GET: JSON Data Object or Plain Text
3. ESP32 HTTP POST: URL Encoded, JSON Data Object, Plain Text

Prerequisites

Before proceeding with this tutorial, make sure you complete the following prerequisites.

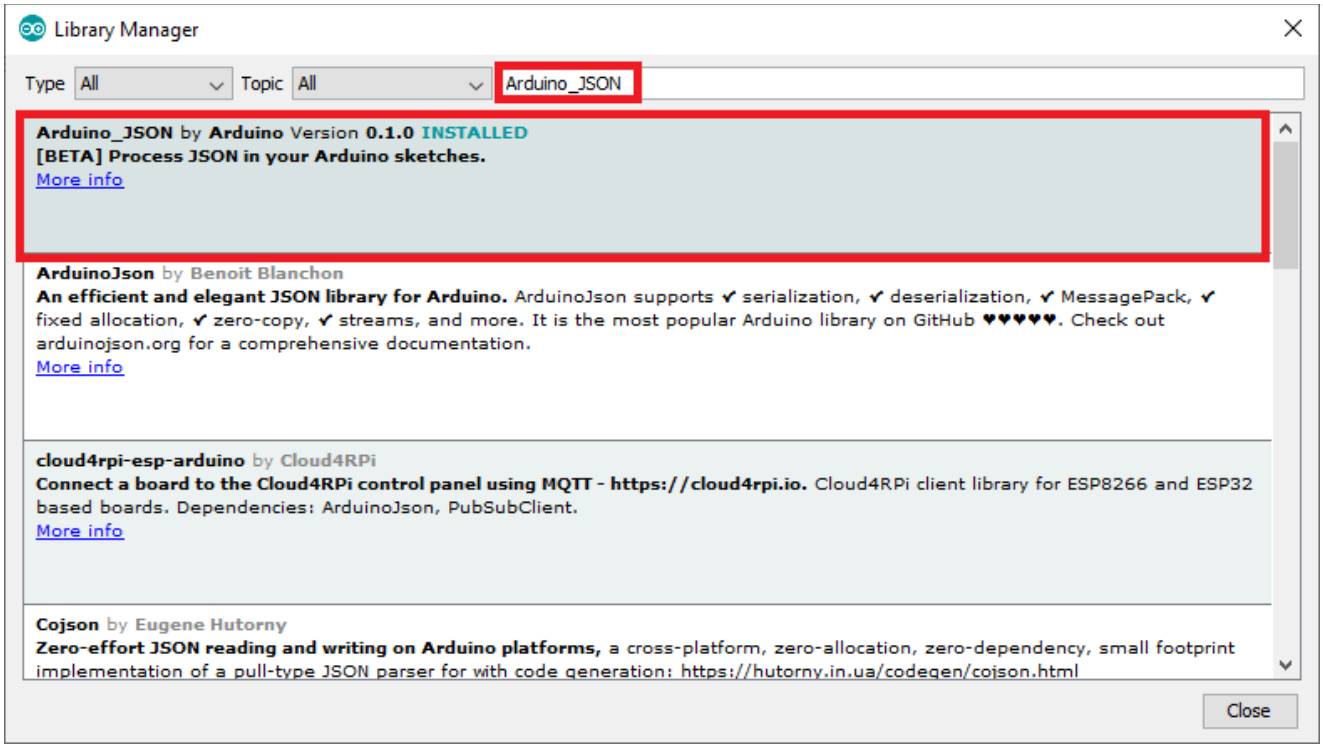
Arduino IDE

We'll program the ESP32 using Arduino IDE, so make sure you have the ESP32 add-on installed.

- Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux)

Arduino_JSON Library

You also need to install the Arduino_JSON library. You can install this library in the Arduino IDE Library Manager. Just go to **Sketch** > **Include Library** > **Manage Libraries** and search for the library name as follows:



Parts Required

For this tutorial you need the following parts:

- ESP32 (read Best ESP32 development boards)
- Raspberry Pi board (read Best Raspberry Pi Starter Kits)
- MicroSD Card – 16GB Class10
- Raspberry Pi Power Supply (5V 2.5A)
- Jumper wires
- Breadboard

You can use the preceding links or go directly to MakerAdvisor.com/tools to find all the parts for your projects at the best price!

Preparing Node-RED (optional)

As an example, we'll create a web service with a Raspberry Pi and Node-RED to act as a web service (like an API). Basically, you'll make HTTP GET and HTTP POST requests to your Raspberry Pi to get values or update them. You can use any other web service

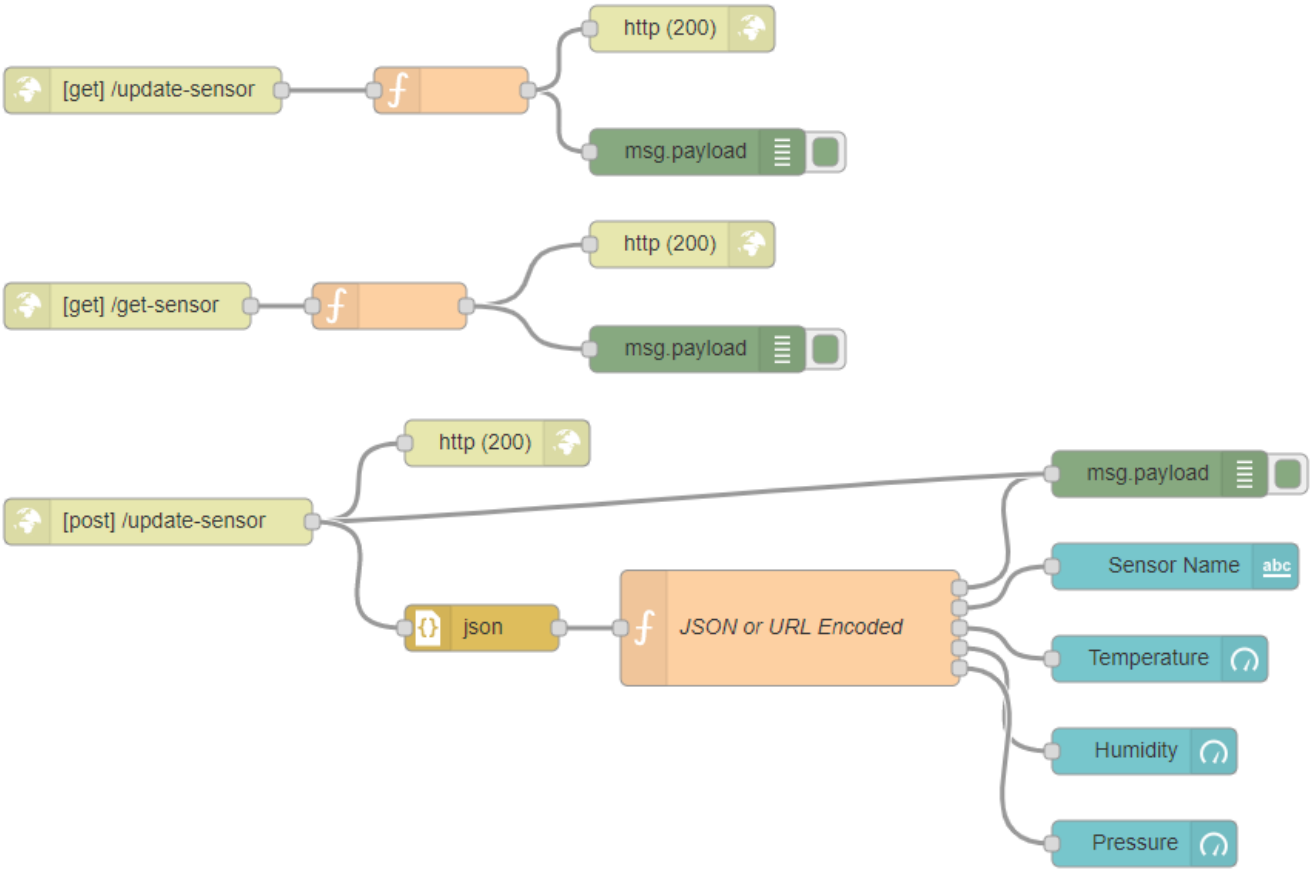
If you don't have Node-RED installed, follow the next tutorials:

- Getting Started with Node-RED on Raspberry Pi
- Installing and Getting Started with Node-RED Dashboard

Having Node-RED running on your Raspberry Pi, go to your Raspberry Pi IP address followed by :1880.

```
http://raspberry-pi-ip-address:1880
```

The Node-RED interface should open. You can simply import the final flow:



Go to **Menu** > **Import** and copy the following to your Clipboard to create your Node-RED flow.

```
[{"id":"599740b7.efde9","type":"http response","z":"b01416d3.f69f
```

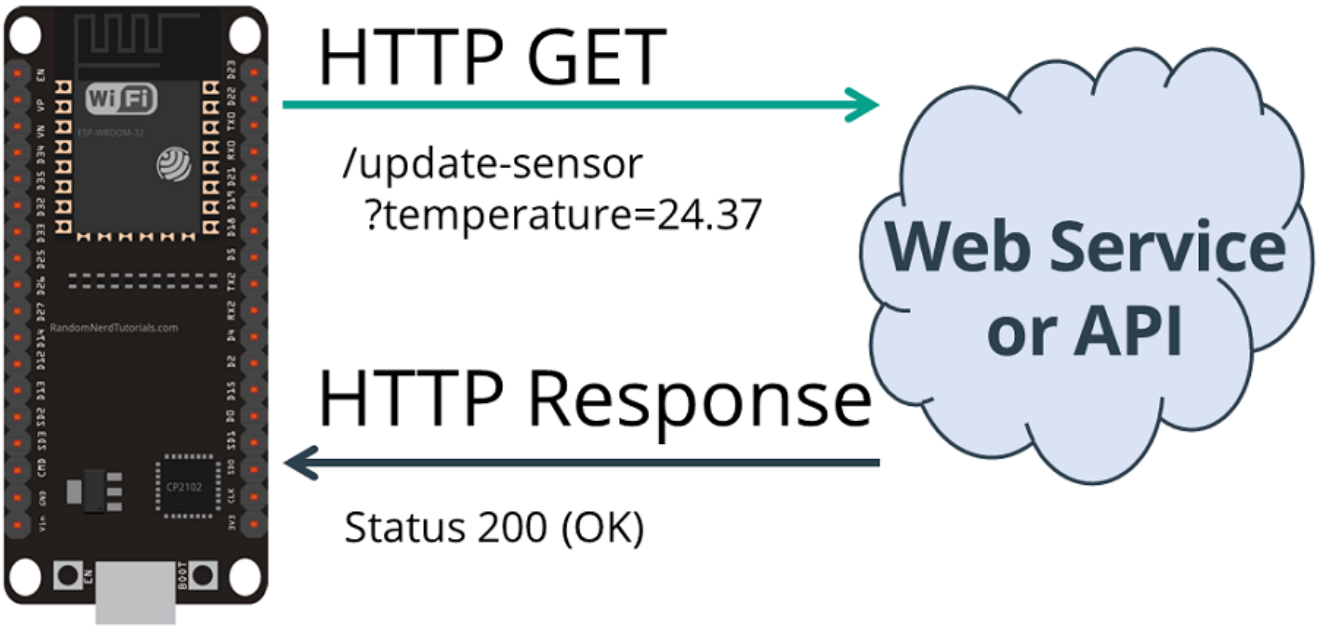
Other Web Services or APIs

In this guide, the ESP32 performs HTTP requests to Node-RED, but you can use these examples with other services like ThingSpeak, IFTTT.com (WebHooks service), OpenWeatherMap.org, PHP server, etc… All examples presented in this guide will also work with other APIs.

In summary, to make this guide compatible with any service, you need to search for the service API documentation. Then, you need the server name (URL or IP address), and parameters to send in the request (URL path or request body). Finally, modify our examples to integrate with any API you want to use.

1. ESP32 HTTP GET: Value or Query in URL

In the first example, the ESP32 will make an HTTP GET request to update a reading in a service. This type of request could also be used to filter a value, request a value, or return a JSON object.



Code ESP32 HTTP GET with Arduino IDE

After installing the necessary board add-ons and libraries, copy the following code to your Arduino IDE, but don't upload it yet. You need to make some changes to make it work for you.

```
/*
```

```
  Complete project details at Complete project details at https

  Permission is hereby granted, free of charge, to any person o
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall b
  copies or substantial portions of the Software.
*/

#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

//Your Domain name with URL path or IP address with path
String serverName = "http://192.168.1.106:1880/update-sensor";

// the following variables are unsigned longs because the time,
// milliseconds, will quickly become a bigger number than can b
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
```

View raw code

Setting your network credentials

Modify the next lines with your network credentials: SSID and password. The code is well commented on where you should make the changes.

```
// Replace with your network credentials
const char* ssid     = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

You also need to type your domain name or Node-RED IP address, so the ESP publishes the readings to your own server.

```
String serverName = "http://192.168.1.106:1880/update-sensor";
```

Now, upload the code to your board and it should work straight away.

Read the next section, if you want to learn how to make the HTTP GET request.

HTTP GET Request

In the `loop()` is where you actually make the HTTP GET request every 5 seconds with sample data:

```
String serverPath = serverName + "?temperature=24.37";

// Your Domain name with URL path or IP address with path
http.begin(serverPath.c_str());

// If your need Node-RED/server authentication, insert user and p
//http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_

// Send HTTP GET request
int httpResponseCode = http.GET();
```

Note: if Node-RED requires authentication, uncomment the following line and insert the Node-RED username and password.

```
// If you need Node-RED/server authentication, insert user and pa
//http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_
```

The ESP32 makes a new request in the following URL to update the sensor field

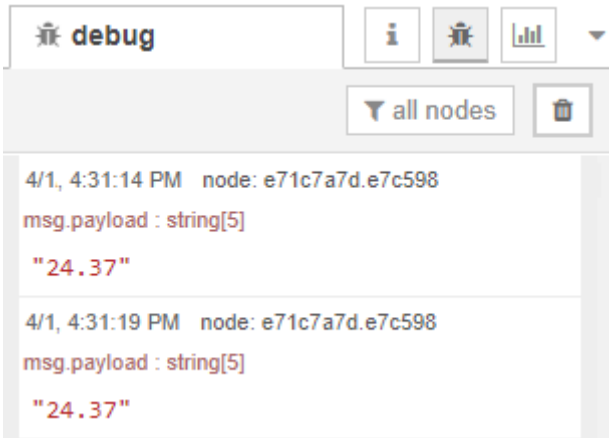
```
http://192.168.1.106:1880/update-sensor?temperature=24.37
```

Then, the following lines of code save the HTTP response from the server.

```
if (httpResponseCode>0) {
  Serial.print("HTTP Response code: ");
  Serial.println(httpResponseCode);
  String payload = http.getString();
  Serial.println(payload);
}
else {
  Serial.print("Error code: ");
  Serial.println(httpResponseCode);
}
```

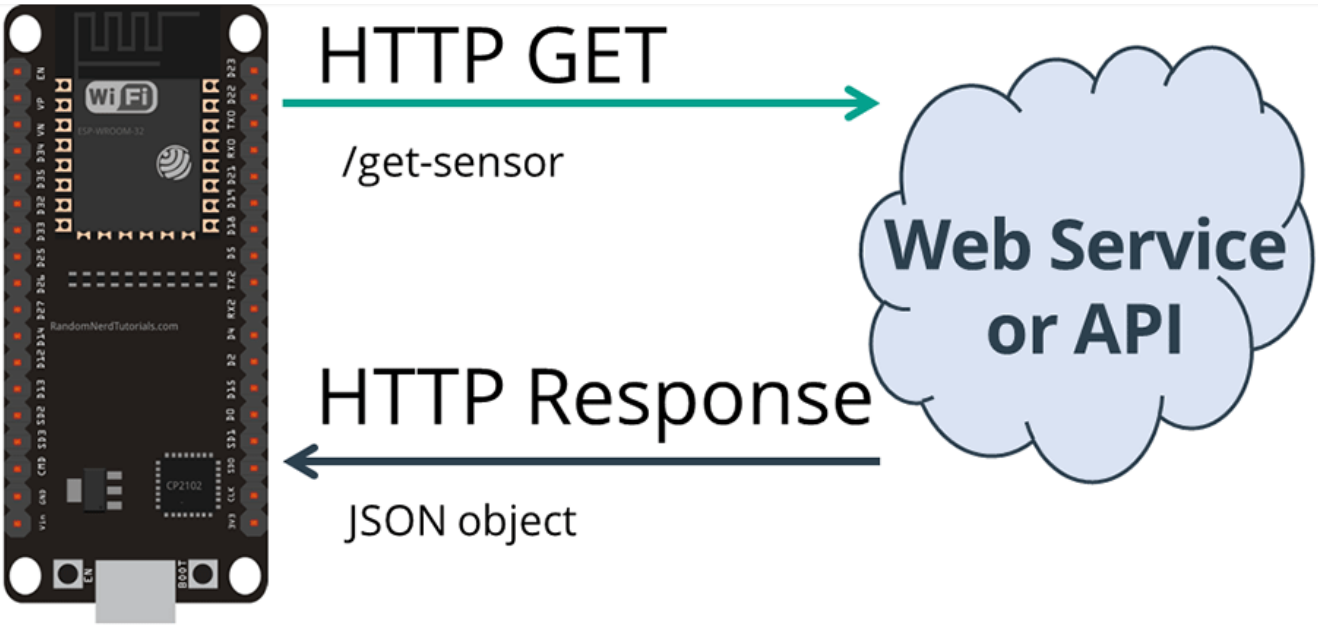
Demonstration

With your board running the new sketch, open the Node-RED debug window. You'll see that the sample values are being printed successfully (24.37).



2. ESP32 HTTP GET: JSON Data Object or Plain Text

This next example shows how to make an HTTP GET request to get a JSON object and decode it with the ESP32. Many APIs return data in JSON format.



Copy the next sketch to your Arduino IDE (type your SSID and password):

```
/*
  Rui Santos
  Complete project details at Complete project details at https

  Permission is hereby granted, free of charge, to any person o
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall b
  copies or substantial portions of the Software.
*/

#include <WiFi.h>
#include <HTTPClient.h>
#include <Arduino_JSON.h>

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

//Your Domain name with URL path or IP address with path
const char* serverName = "http://192.168.1.106:1880/get-sensor"

// the following variables are unsigned longs because the time
```

```
unsigned long lastTime = 0;
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
```

View raw code

Setting your serverName

Enter your domain name or Node-RED IP address, so the ESP requests the sensor readings that will be retrieved in a JSON object.

```
String serverName = "http://192.168.1.106:1880/get-sensor";
```

Now, upload the code to your board.

HTTP GET Request (JSON Object)

In the `loop()`, call the `httpGETRequest()` function to make the HTTP GET request:

```
sensorReadings = httpGETRequest(serverName);
```

The `httpGETRequest()` function makes a request to Node-RED address *http://192.168.1.106:1880/get-sensor* and it retrieves a string with a JSON object.

```
String httpGETRequest(const char* serverName) {
  HTTPClient http;

  // Your IP address with path or Domain name with URL path
  http.begin(serverName);

  // If you need Node-RED/server authentication, insert user and
  //http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLAC
```

```
    int httpResponseCode = http.GET();

    String payload = "{}";

    if (httpResponseCode>0) {
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
      payload = http.getString();
    }
    else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
    // Free resources
    http.end();

    return payload;
}
```

Note: if Node-RED requires authentication, uncomment the following line and insert the Node-RED username and password.

```
// If you need Node-RED/server authentication, insert user and pa
//http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_
```

Decoding JSON Object

To get access to the values, decode the JSON object and store all values in the `sensorReadingsArr` array.

```
JSONVar myObject = JSON.parse(sensorReadings);

// JSON.typeof(jsonVar) can be used to get the type of the var
```

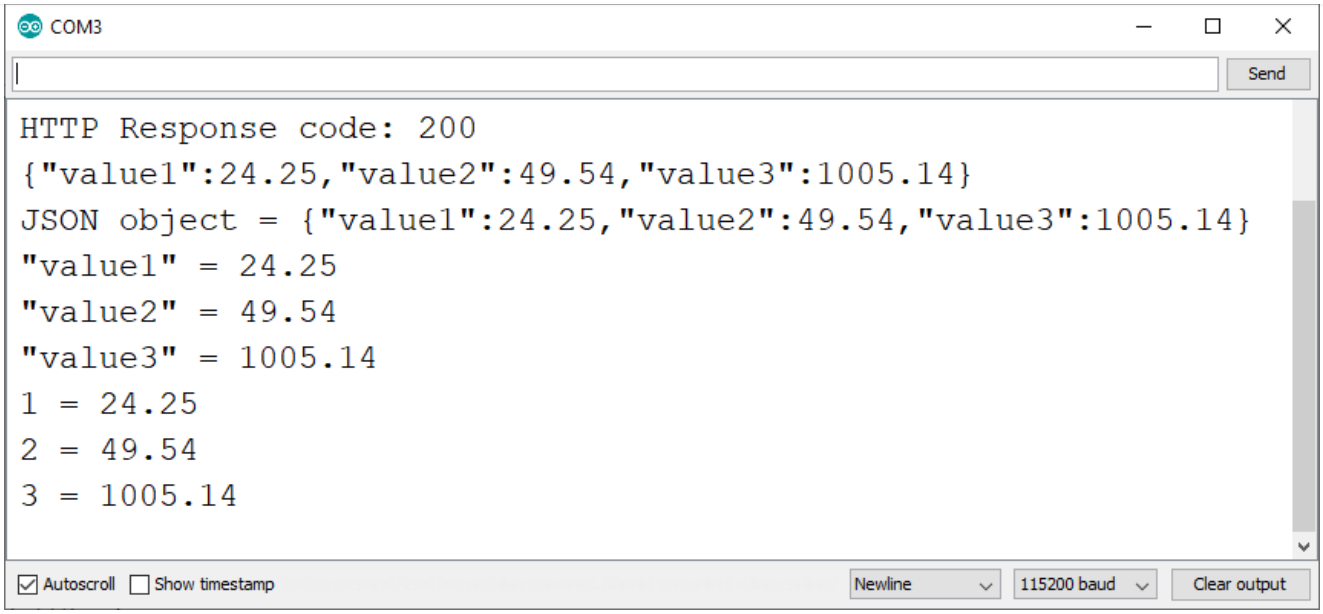
```
    return;
  }

  Serial.print("JSON object = ");
  Serial.println(myObject);

  // myObject.keys() can be used to get an array of all the keys ir
  JSONVar keys = myObject.keys();

  for (int i = 0; i < keys.length(); i++) {
    JSONVar value = myObject[keys[i]];
    Serial.print(keys[i]);
    Serial.print(" = ");
    Serial.println(value);
    sensorReadingsArr[i] = double(value);
  }
  Serial.print("1 = ");
  Serial.println(sensorReadingsArr[0]);
  Serial.print("2 = ");
  Serial.println(sensorReadingsArr[1]);
  Serial.print("3 = ");
  Serial.println(sensorReadingsArr[2]);
```

HTTP GET Demonstration

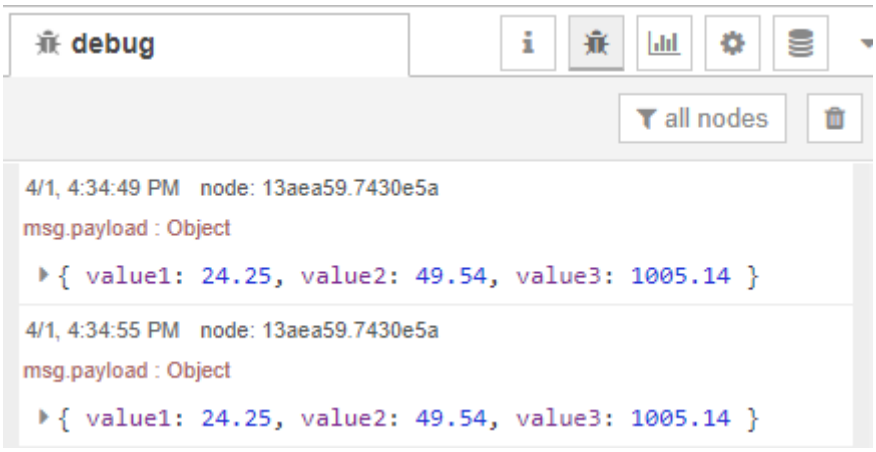
After uploading the code, open the Arduino IDE and you'll see that it's receiving the following JSON data:

```
{"value1":24.25,"value2":49.54,"value3":1005.14}
```

Then, you print the decoded JSON object in the Arduino IDE Serial Monitor.



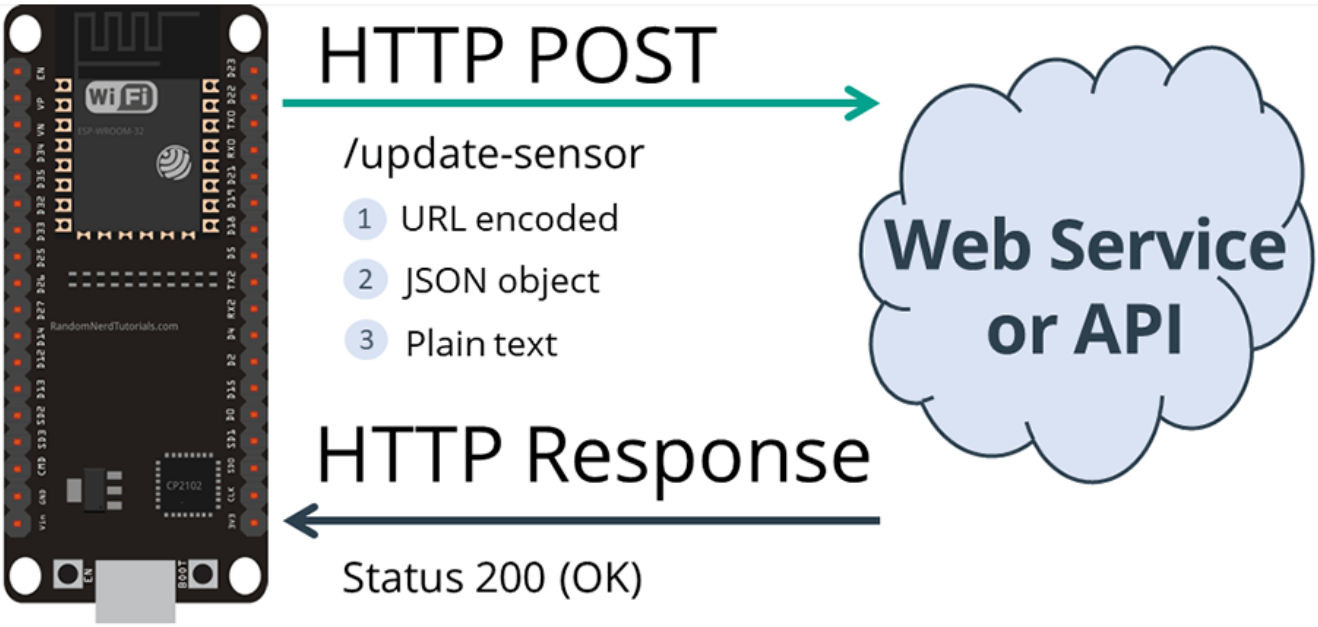
For debugging purposes, the requested information is also printed in the Node-RED debug window.



3. ESP32 HTTP POST: URL Encoded, JSON Data Object, Plain Text

Finally, you'll learn how to make an HTTP POST request with an ESP32.

With this example, your ESP32 can make HTTP POST requests using three different types of body requests: URL encoded, JSON object or plain text. These are the most common methods and should integrate with most APIs or web services.



Copy the next sketch to your Arduino IDE (type your SSID and password):

```
/*
  Rui Santos
  Complete project details at Complete project details at https

  Permission is hereby granted, free of charge, to any person o
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall b
  copies or substantial portions of the Software.
*/

#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

//Your Domain name with URL path or IP address with path
const char* serverName = "http://192.168.1.106:1880/update-sens

// the following variables are unsigned longs because the time,
// milliseconds will quickly become a bigger number than can b
```

```
// Timer set to 10 minutes (600000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
```

View raw code

Setting your serverName

Enter your domain name or Node-RED IP address, so the ESP posts sample sensor readings.

```
String serverName = "http://192.168.1.106:1880/update-sensor";
```

Now, upload the code to your board.

HTTP POST URL Encoded

To make an HTTP POST request of type URL encoded, like this

```
POST /update-sensor HTTP/1.1
Host: 192.168.1.106:1880
api_key=tPmAT5Ab3j7F9&sensor=BME280&value1=24.25&value2=49.54&value3=1005.
14
Content-Type: application/x-www-form-urlencoded
```

You need to run the following in your Arduino code:

```
// Your Domain name with URL path or IP address with path
http.begin(serverName);

// If you need Node-RED/server authentication, insert user and pa
//http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_

// Specify content-type header
http.addHeader("Content-Type", "application/x-www-form-urlencodec
```

```
String httpRequestData = "api_key=tPmAT5Ab3j7F9&sensor=BME280&val

// Send HTTP POST request
int httpResponseCode = http.POST(httpRequestData);
```

Note: if Node-RED requires authentication, uncomment the following line and insert the Node-RED username and password.

```
// If you need Node-RED/server authentication, insert user and pa
//http.setAuthorization("REPLACE_WITH_SERVER_USERNAME", "REPLACE_
```

HTTP POST JSON Object

Or if you prefer to make an HTTP POST request with a JSON object:

```
POST /update-sensor HTTP/1.1
Host: example.com
{api_key: "tPmAT5Ab3j7F9", sensor_name: "BME280", temperature: 24.25;
humidity: 49.54; pressure: 1005.14}
Content-Type: application/json
```

Use the next snippet:

```
http.addHeader("Content-Type", "application/json");

int httpResponseCode = http.POST("{\"api_key\":\"tPmAT5Ab3j7F9\",
```

HTTP Plain Text

If you want to send plain text or a value, use the following:

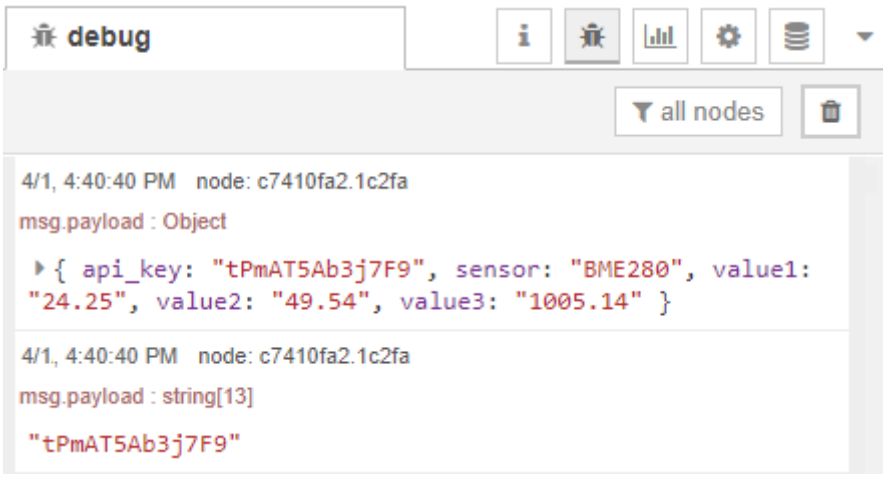
```
http.addHeader("Content-Type", "text/plain");
```

```
int httpResponseCode = http.POST("Hello, World!");
```

Note: the Node-RED flow we're using (web service) is not setup to receive plain text, but if the API that you plan to integrate only accepts plain text or a value, you can use the previous snippet.

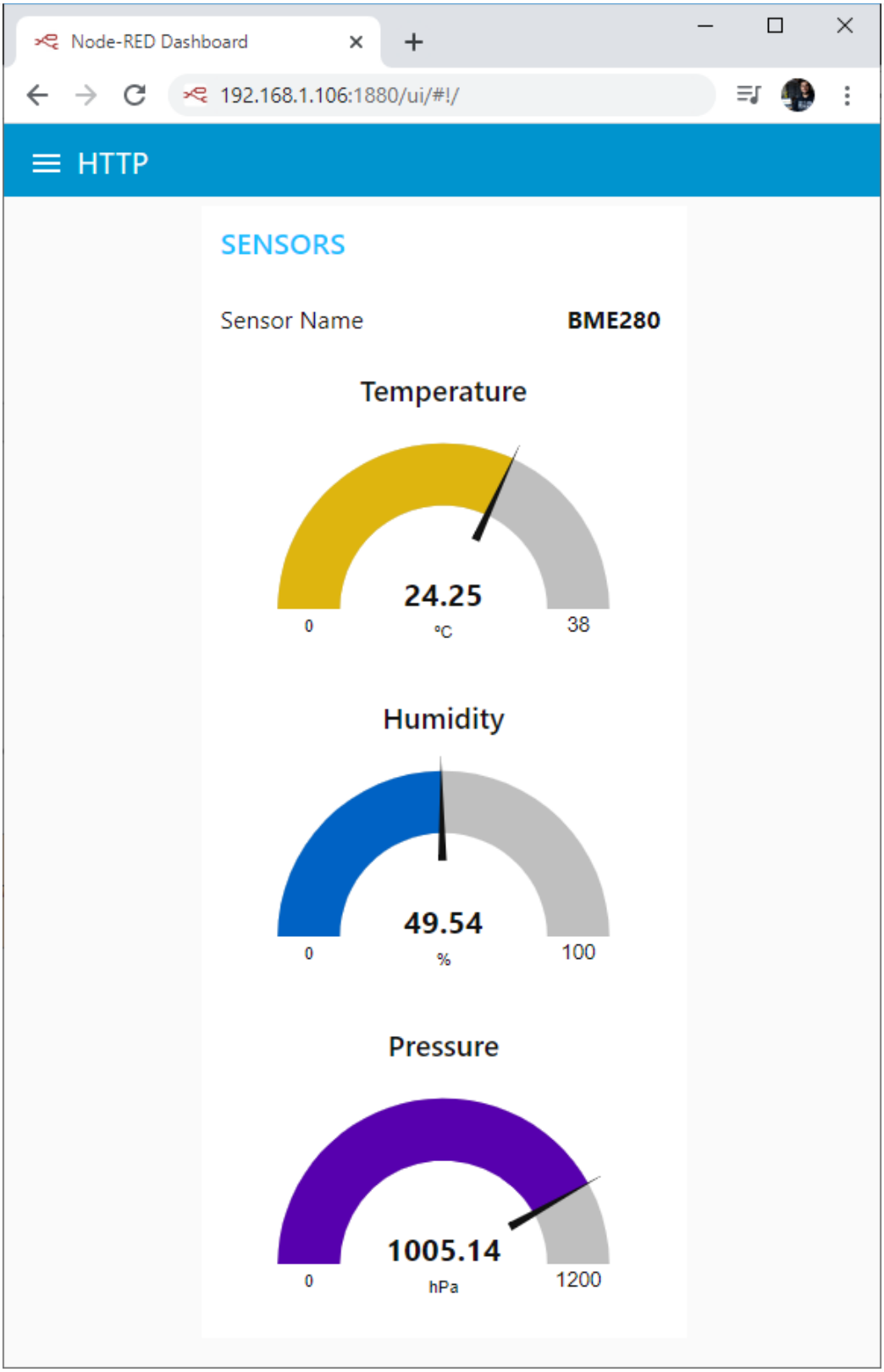
HTTP POST Demonstration

In the Node-RED debug window, you can view that your ESP is making an HTTP POST request every 5 seconds.



And in this example, those values are also sent to 3 Gauges and are displayed in Node-RED Dashboard:

```
http://raspberry-pi-ip-address:1880/ui
```



Wrapping Up

In this tutorial you've learned how to integrate your ESP32 with online services using HTTP GET and HTTP POST requests.

HTTP GET and HTTP POST are commonly used in most web services and APIs.

your own server; to request data from the internet or from your database, and much more.

If you're using an ESP8266 board, read: Guide for ESP8266 NodeMCU HTTP GET and HTTP Post Requests.

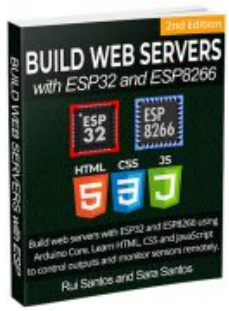
You might also like reading:

- **[Course] Learn ESP32 with Arduino IDE**
- ESP32/ESP8266 Send Email Notification using PHP Script
- Visualize Your Sensor Readings from Anywhere in the World (ESP32/ESP8266 + MySQL + PHP) using Charts
- ESP32 Relay Module Web Server
- MicroPython Programming with ESP32 and ESP8266

I hope you liked this project. If you have any questions, post a comment below and we'll try to get back to you.

If you like ESP32, you might consider enrolling in our course "Learn ESP32 with Arduino IDE". You can also access our free ESP32 resources here.

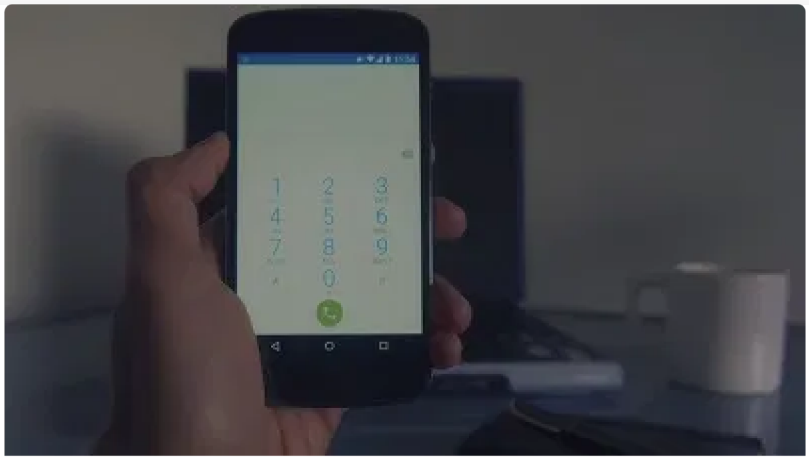
Thank you for reading.



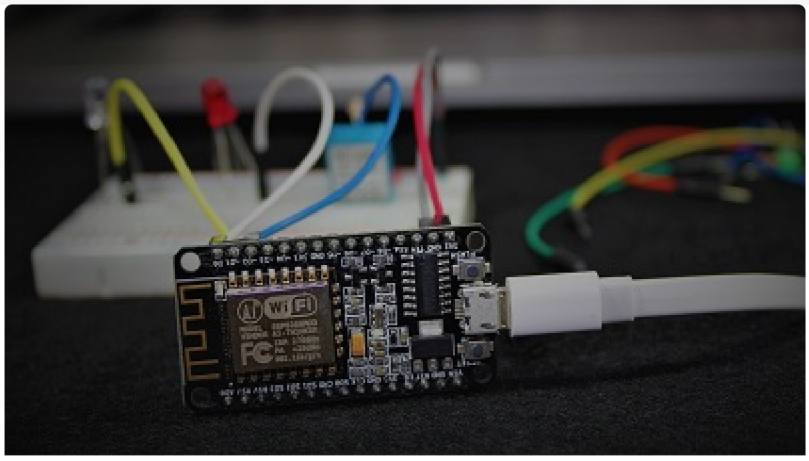
[eBook] Build Web Servers with ESP32 and ESP8266 (2nd Edition)

Build Web Server projects with the ESP32 and ESP8266 boards to control outputs and monitor sensors remotely. Learn HTML, CSS, JavaScript and client-server communication protocols **DOWNLOAD »**

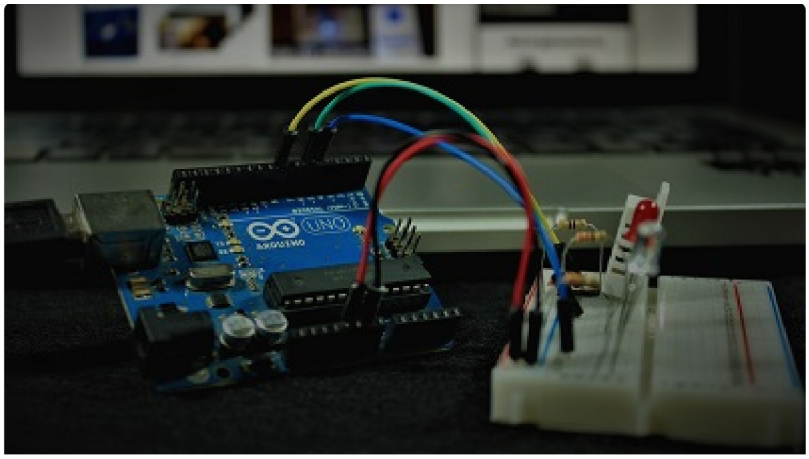
Recommended Resources



Build a Home Automation System from Scratch » With Raspberry Pi, ESP8266, Arduino, and Node-RED.



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



What to Read Next…

Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE

ESP8266 Voltage Regulator (LiPo and Li-ion Batteries)

ESP32/ESP8266: Firebase Authentication (Email and Password)

74 thoughts on “ESP32 HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text)”

Steph

April 8, 2020 at 10:47 am

Hey guys,

It's a very good thing that you're finally covering that part. The other day I was wondering why you didn't put more emphasis on the specifics of communications that use the HTTP[S] protocol.

In particular, the possibility of transporting structured data in JSON format. And I was going to talk about this in one of my next tutorials. But here it is, it's done.

And you could also take the opportunity to talk about the excellent Postman tool that greatly facilitates the development and introspection of HTTP APIs…

https://www.postman.com/

See you soon,
Steph

Reply

Sara Santos
April 9, 2020 at 9:42 am

Hi Steph.
You're right.
We should have published about this subject earlier.
Thank you for your suggestion.
Regards,
Sara

Reply

Wayne

October 31, 2022 at 9:24 am

Yes, I agree with this. Very little information examples with https to public servers.

Reply

miguel alexandre wisintainer

April 8, 2020 at 11:41 am

Hi, excelent blog

Dont forget too the DELETE, CONNECT

Reply

Sara Santos

April 9, 2020 at 9:46 am

Hi Miguel.
There are also other HTTP methods: like HTTP PUT, HTTP DELETE, HTTP PATCH
But the ones we talked about in this tutorial are the most relevant for the ESP32.
Thank you for your comment.

Reply

RayB

April 9, 2020 at 1:59 am

Another excellent tutorial BUT could you confirm all of this equally applies to ESP8266

Reply

Sara Santos

April 9, 2020 at 2:30 pm

Hi.
We've just published a similar tutorial for the ESP8266: https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/
Regards,
Sara

Reply

miguel alexandre wisintainer

April 9, 2020 at 6:49 pm

Hi, some bugs on BLOG

JSON format, need quotes and node-red JSON name fields that dont

Reply

Rui Santos

April 10, 2020 at 10:58 am

Hello Miguel,
The guide is correct, for that Node-RED the HTTP POST request is the following (api_key, sensor, value1, value2, value3):
https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/ESP32/HTTP/ESP32_HTTP_POST.ino

```
// Specify content-type header
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
// Data to send with HTTP POST
String httpRequestData =
"api_key=tPmAT5Ab3j7F9&sensor=BME280&value1=24.25&value2=49.
54&value3=1005.14";
// Send HTTP POST request
int httpResponseCode = http.POST(httpRequestData);
```

Or in JSON

```
// If you need an HTTP request with a content type: application/json, use
the following:
http.addHeader("Content-Type", "application/json");
int httpResponseCode =
http.POST("{\"api_key\":\"tPmAT5Ab3j7F9\",\"sensor\":\"BME280\",\"value
1\":\"24.25\",\"value2\":\"49.54\",\"value3\":\"1005.14\"}");
```

So, the Node-RED flow needs to be:

```
var msg0 = { payload: msg.payload.api_key };
var msg1 = { payload: msg.payload.sensor };
var msg2 = { payload: msg.payload.value1 };
```

return [msg0, msg1, msg2, msg3, msg4];

Reply

Tapas

April 19, 2020 at 4:49 am

Thanks For this excellent tutorial.

How to pass variable in JSON format .

http.POST("{\"api_key\":\"tPmAT5Ab3j7F9\",\"sensor\":\"BME280\",\"value1\":variable,\"value2\":variable,\"value3\":variable}");

Reply

behzad

April 13, 2020 at 10:49 am

THIS IS VERY AWESOME !
Thank you sir.. sooo very much

Reply

Sara Santos

April 14, 2020 at 2:48 pm

Thanks 😀

Michael

June 5, 2020 at 1:36 am

Does anyone know how I can solve this problem?
'JSONVar' was not declared in this scope
JSONVar myObject = JSON.parse(sensorReadings);

Reply

Sara Santos

June 5, 2020 at 9:31 am

Hi.
You need to install the Arduino_JSON Library with the "_" on the name.
Regards,
Sara

Reply

Ling

August 5, 2020 at 3:32 am

hi thank you for making this tutorial, its very helpful.

but i was wondering would i be able to get a response from web server without raspberry pi, can i replace that with Arduino?

Reply

Ling

August 5, 2020 at 3:41 am

also, instead of getting a response from NODE_RED, can i use a web server that i built following this tutorial??
https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/

i tried this, but failed, in my arduino serial monotor, it shows the folloeing msg
Error code: -1
13:41:28.031 -> {}
13:41:28.031 -> JSON object = {}
13:41:28.031 -> 1 = 0.00
13:41:28.031 -> 2 = 0.00
13:41:28.031 -> 3 = 0.00
13:41:33.036 -> Error code: -1
13:41:33.036 -> {}
13:41:33.036 -> JSON object = {}
13:41:33.036 -> 1 = 0.00
13:41:33.036 -> 2 = 0.00
13:41:33.036 -> 3 = 0.00

Reply

Ling

August 5, 2020 at 5:29 am

I was wondering if I can replace raspberry pi with Arduino or even only using esp32?
and if I can use my Arduino built web server which is what I built following this tutorial https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/, instead of using node_red??

Reply

nia1286
October 9, 2020 at 12:57 pm

can it be used without raspberry pi?

Reply

Sara Santos
October 10, 2020 at 10:05 am

Hi.
Yes. These are just examples that show how to make a request to the Raspberry Pi.
You can make requests to any other resources or paths.
Regards,
Sara

Reply

Johan

October 11, 2020 at 1:43 pm

This tutorial works great for a POST and GET, but DELETE is not mentioned. But it is also not supported by the HTTPClient. When I replace the http.GET() with http.DELETE() it gives an error:

error: ‘class HTTPClient’ has no member named ‘DELETE’

I searched the internet, but all I got was how to delete files and stuff, while I am trying to send a DELETE request. Any idea how to fix this?

Reply

Paulo

November 11, 2020 at 10:43 pm

Hi, congrats for the great tutorial.

How about if I need to post an image stored on the local SD card to the web?
Or an image just taken with ESP32-CAM?
What kind of post should I use?

Thanks in advance
Paulo

Reply

Sara Santos

Hi.
Maybe these tutorials might be useful:
https://randomnerdtutorials.com/esp32-cam-http-post-php-arduino/
https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/
Regards,
Sara

Reply

Martin Rice
November 16, 2020 at 10:44 am

Hi,

Is there an equivalent tutorial for microPython? Specifically, how do I send POST requests to ThingSpeak. (I know how to do this with Raspberry Pi and Python.)

Thanks

Reply

Sara Santos
November 17, 2020 at 3:56 pm

Hi.
Unfortunately, at the moment, we don't have any tutorials about that subject.
Regards,

Reply

Lorenzo Innocenti

November 20, 2020 at 3:57 pm

Hi! Great tutorial! Do you have anything similar for Ethernet?
I've tried using the same HTTPClient library to reach an https host over Ethernet, but after a successful client.begin() the ESP crashes, most likely due to the fact that the HTTPClient library is calling functions from the WiFiClientSecure library, which apparently as the name says is only good for Wi-FI 🙂

Thanks!

Reply

sergio

January 10, 2023 at 4:49 am

do you already have solution for this, cause i am still looking for solution, especiall i am using esp32 with w5500 lite

Reply

Vicente L. C. Rubino

November 22, 2020 at 2:18 pm

What if the json code from the server is something like:

{
“code”:”A10000″,
“data”:{
“ask”:”100438.41″,
“bid”:”100087.96″,
“high24h”:”101971.13″,
“lastPrice”:”100500.08″,
“timestamp”:1606017474423
},
“message”:”Success”
}

How to deserialize the internal data segment and read the field lastPrice, for example?

Thanks!

Reply

Sara Santos
November 23, 2020 at 3:34 pm

Hi.
For that, I recommend using a JSON library.
The Arduino_JSON is one of the easiest to work with:
https://github.com/arduino-libraries/Arduino_JSON

Save the server response on a JSONVar, for example:
JSONVar myObject = JSON.parse(RESPONSE FROM THE SERVER);

Then, access the lastPrice as follows:
String lastPrice = myObject[“lastPrice”];

I hope this helps.
Regards,
Sara

Reply

Vicente L. C. Rubino
December 1, 2020 at 2:55 pm

Thank you for your attention. The problem was solved!

Reply

MGI
February 17, 2021 at 12:34 am

Thank you for this tutorial. But it does not show how a JSON object can be POSTed. Instead, it explains how to send a string containing JSON syntax. You should have mentioned the use of the PrintTo method from JSON objects, that is needed to POST a JSON object without having to encode manually the data

Reply

Sara Santos
February 17, 2021 at 11:11 am

Thanks.

Regards,
Sara

Reply

maul

March 22, 2021 at 7:05 am

hai, i have an error with the header library, is that correct? or it should be
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
in case I missing some understood?

Reply

Sara Santos

March 23, 2021 at 11:46 am

Hi.
What is exactly the error that you get?
Regards,
Sara

Reply

Flipside

June 19, 2021 at 8:28 pm

Hello!

First of all, thank you for the great tutorials!
I would like to ask a question about the connection between a smartphone and ESP32.
Suppose that I have only the hotspot feature of my smartphone on (no wifi, no mobile data).
Is there a way to establish a connection between them, so the ESP32 can send numbers (this could be a temperature value or anything else) to the smartphone?
The general idea is to have an offline webpage in my smartphone, so i can see the values that the ESP32 is sending, for example to auto refresh every 1 minute.

Regards.

Reply

Jeroen

June 21, 2021 at 8:08 pm

Hi, get the error “HTTPClient.h no such file or directory”. I’m missing the correct library I guess? I searched for it but I cant find it anywhere. Can you help me out?

Thank you for the great tutorials!

Reply

Justin

hi i am trying the http post part but i am unable to get the esp32 to connect to the node-red server and i am unsure why

Reply

Sara Santos

June 23, 2021 at 10:45 pm

Hi.
Can you provide more details abou the issue?
Regards,
Sara

Reply

Justin

June 29, 2021 at 3:19 am

in the serial monitor it says software cause connection abort

Reply

tamer

July 11, 2021 at 12:52 pm

what's mean
HTTP Response code: 400

Reply

Sara Santos

July 12, 2021 at 10:11 am

Hi.
That means something is wrong with your request.
“The HyperText Transfer Protocol (HTTP) 400 Bad Request response status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error”
Regards,
Sara

Reply

Muhammad

July 22, 2021 at 12:58 am

Does the http post work with https?

Reply

Juan de Jong

July 23, 2021 at 2:58 pm

Hi Rui !, I need to read a value in Thinkspeak
“https://api.thingspeak.com/channels/1450703/fields/1.json?results=2” can

you help me with this, I tried the code you use and it gives me 400 the status of the petition, already very grateful.

Reply

Sara Santos

July 23, 2021 at 3:05 pm

Hi.
Try the ThingSpeak library for the Arduino IDE.
Go to Sketch > Include Library > Manage Libraries… and search for “ThingSpeak” in the Library Manager. Install the ThingSpeak library by MathWorks.
Then, experiment with this example, it shows how to read a value: https://github.com/mathworks/thingspeak-arduino/blob/master/examples/ESP32/ReadField/ReadField.ino
Don’t forget to insert your API key and other details.
I hope this helps.
Regards,
Sara

Reply

Juan de Jong

July 23, 2021 at 3:10 pm

thanks!!!!! i try it

Reply

Hydroen

August 21, 2021 at 11:37 am

Hello,

Could we make sure to send a file located on an SD card?

because I have a little trouble finding a solution to my problem …

Thank you in advance for your answer

Reply

frogg1ven

September 14, 2021 at 11:33 am

Hi,
I have problem with httpclient library, im working with platformio and I can't find exact lib for this enviroment, and I'm wondering if there is no lib or this HTTPClient.h is part of bigger library?

Reply

John

September 22, 2021 at 12:07 am

Hello,

http.POST(“{\”api_key\”:\”tPmAT5Ab3j7F9\”,\”sensor\”:\”BME280\”,\”value1\”:\”24.25\”,\”value2\”:\”49.54\”,\”value3\”:\”1005.14\”}”);

I would like to know how to replace values with variables, for example assign value1 to a variable?

Thank you

Reply

Sara Santos

September 22, 2021 at 9:36 pm

Hi.
Create your variable. It must be a String.
Then, just replace “value1” with the name of the variable.
Regards,
Sara

Reply

Paulo Borges

September 25, 2021 at 10:38 pm

Hi, I am trying to submit a web picture to Google Vision using the provided principle to post a json request but something is not working as I always get httpResponseCode = -5 in return.
Making the same request with the same image from nodered or from the try yourself on the Google Vision page
https://cloud.google.com/vision/docs/ocr?

TION%22%7D%5D%2C%22image%22%3A%7B%22source%22%3A%7B%22imageUri%22%3A%22http%3A%2F%2Fwww.newdesignfile.com%2Fpostpic%2F2010%2F05%2Ffree-stock-photos-people_102217.jpg%22%7D%7D%7D%5D%7D%7D#try_it

works as well.
Any sugestion of what may be wrong?
Thanks

Reply

Jim Allison
December 17, 2021 at 2:03 pm

Can you update this guide, or respond to this comment, to show how to send a plain text POST request using the AsyncTCP and ESPAsyncWebServer libraries? I purchased the "Build_Web_Servers_ESP32_ESP8266_V1" and it references this guide, but the libraries are different here, and so I need some help.

Reply

Sara Santos
December 17, 2021 at 2:55 pm

Hi.
I think this tutorial explains it pretty well for the Asyncwebserver:
techtutorialsx.com/2018/02/03/esp32-arduino-async-server-controlling-http-methods-allowed/
Regards,
Sara

Reply

Jim Allison

December 17, 2021 at 3:11 pm

Thank you! This is helpful.

Reply

UweR7

December 17, 2021 at 11:22 pm

Really excellent.
But on part is missing completely: The server side in C#, .NET 5.0.
Regarding swagger and Postman works my “ASP.NET Core Web Api” works fine.
But I am not able to call the C# code from a ESP32 via GET nor POST (“plain text” or “JSON” formatted).

Any ideas, hints? (content-type is well set, and as shown by swagger.

Reply

Trung

June 18, 2022 at 10:32 am

Have you solved the problem yet? Can you talk to me?

Yugo
January 18, 2022 at 5:04 am

Hi ,thanks for the awesome tutorial. I want to publish the json data with mqtt, can you give me any information about that, I have no idea to do by myself.

Reply

Yugo
January 19, 2022 at 10:00 am

And I have the problem with this tutorial. Esp32 always receives “the error code:-1”, I think it caused by my url path. I use nodered and its ip is 127.0.0.1, Does it matter?

Reply

David W
January 29, 2022 at 6:17 pm

Good morning,
Following your example I have:

```
#include <WiFi.h>
#include <HTTPClient.h>
```

```
HTTPClient http;
Serial.println("[HTTP] begin…\n");
http.begin("https://data.tankutility.com/api/getToken");
http.setAuthorization("ZGd3ZXRlQGdtYWlsLmNvbTpQb2lrYTRUYW5r");
Serial.println("[HTTP] GET…\n");
```

The above code works from within Arduino IDE but if I attempt the same in PlatformIO, using the same http library, the methods are different. There is no "http.begin", only http.beginrequest which doesn't take any parameters. I checked the versions of HTTPClient.h in both cases and they appear to be the same. Can you give me any help?
Thank you

Reply

David
January 29, 2022 at 7:23 pm

Sorry, again but I figured it out. The PlatformIO http client I should use is an ESP32 built-in library. I found this statement on the internet: "…you globally installed the erronous amcewen/HttpClient in PlatformIO. This will prevent the Arduino-ESP32's internal HTTPClient.h library to be picked up."
This now works in PlatformIO as well.

Thank you, again

Reply

Yaroslav
February 3, 2022 at 1:26 pm

Mi servidor host tiene una dirección https y respuesta servidor “Connected to WiFi network with IP Address: 192.168.0.16
14:19:46.313 -> httpsRequestData: api_key=tPmAT5Ab3j7F9&sensor=BMP180&location=Office&value1=20.40&value2=95873
14:19:46.408 -> HTTP Response code: 400

Reply

umer

July 20, 2022 at 12:18 pm

Hi I am using post request but to send this to Nodejs and I am getting error.
HTTP Response code: -1

Reply

John

August 9, 2022 at 4:14 am

The get request responds with httpResponseCode:
int httpResponseCode = http.GET();

What are the meaning of the negative codes?
I’m seeing intermittenty a -11 response. What does that mean?

Reply

G Reddy

February 5, 2023 at 12:50 pm

Did you ever find an answer to this? I'm too am getting -11 responses, but server seems to think everything is okay.

Have you found a lookup table for the httpResponseCode values?

Reply

Renzo

August 21, 2022 at 3:38 pm

Hi Sara,
This is very interesting, but in the future can you go more deeply in explaing the syntax and the structure of Json objects with the connection to Firebase?
I.e. we find the results in the screen line by line:
Temp: 27,6
Humidy: 46.
Can we obtain more data in the same line:
Temp: 27,6 Humidity 46.
Or more a table too?

Thanks Renzo

Reply

Hi Sara and Rui,
Very very good post : your are definitively the best !
But what is the simplest way for an ESP32 (not a Raspberry PI !), connected to an unique sensor, to respond in a JSON format, to an HTTP GET/POST request for the sensor value ?
I understood how to perform it using a web page with the sensor value, but is there a simpler solution as an API web service response ?
Thanks.
NB : my scope is to realize, with an ESP32, a photovoltaic router working with the solar POTENTIAL production, estimated using a lux sensor (VEML7700 from Adafruit) connected to a 2nd ESP32. The 1rst ESP32 also uses data from an inverter API (Fronius GEN24), formatted in JSON as described in https://github.com/jlemaire06/Superviseur-solaire, a small monitoring project realized to test the API access, in view of the router.
The doc is in French but I can give you any explanations if you are interested.

Reply

LEMAIRE
November 6, 2022 at 10:50 am

Oh sorry but I found the response to my previous question in another post, https://randomnerdtutorials.com/esp32-client-server-wi-fi/, which uses an AsyncWebServer objects with callbacks to client http GET requests.
Your example works well when both client and server ESP32 are stations in a local network, with obvious changes when defining the Wifi connections.
And, this communication is compatible with other GETs to API, in the client side.
Congratulations !

Rudi

January 2, 2023 at 4:25 am

Great article. I used the method presented to POST a string to a homeassistant/NodeRed installation and works great. One issue I ran into is that the URL generated by HTTP-IN node in NodeRed is now relative to /endpoint which slightly changes the URL's listed. I also get consistently a -11 response out of http.POST but the data is received fine and can be used in NodeRed.

Reply

G Reddy

February 5, 2023 at 12:51 pm

Did you ever find an answer to this? I'm too am getting -11 responses, but server seems to think everything is okay.

Have you found a lookup table for the httpResponseCode values?

Reply

G Reddy

February 5, 2023 at 1:48 pm

https://stackoverflow.com/questions/69327002/arduino-http-post-returns-

HTTPC_ERROR_READ_TIMEOUT

Reply

Joe

February 5, 2023 at 8:36 pm

Here is what I have been using for requests and responses:
`
void sendRequest(String url){
HTTPClient http;
http.begin(url);
int httpResponseCode = http.GET();

```
Serial.print("HTTP Response: ");
Serial.println(httpResponseCode);
```

if (httpResponseCode>0) {
handlePayload(http.getString());
}
http.end();
}

`

Parsing JSON (with an array at the root):
`
void handlePayload(String payload){
Serial.println(payload);
deserializeJson(jsonDoc, payload);

```
String flow = jsonDoc[0]["actionName"];
Serial.println(flow);
```

if(flow == "MyFlowName" && jsonDoc[0]["output\values"]["mode"]

```
validatePackage();
}
}

`
```

Reply

João Paulo

February 16, 2023 at 6:28 pm

Is there a way to add a Chart to this project? I want to aquire audio through my ESP32 and send it to NODE-RED. Is this communication protocol fast enough?

Reply

Chris

March 11, 2023 at 7:02 pm

What might be the best way to receive an action from IFTTT and cause a specific action on a ESP8266 module ?
I have done this on a Particle Photon module and would like to learn how to migrate it to an ESP8266 12-E NodeMCU Kit module.
Here is the project: Using particle.io IDE, IFTTT and Webhooks; when a specific Twitter User has posted a Tweet, Send a message to call a function (Flashing LED) on the Photon module. This is currently working.

Please note, I am new to the webhooks and other methods of sending

I am looking for a similar project that may be doing this so I can learn how to do this for the NodeMCU.

Sorry if I missed any information above that solves this, like I said, new to all the http comms services and how they operate.
Happy to share the project in the future if I can get it solved.
Appreciate any guidance, thank you.

Reply

Sara Santos
March 14, 2023 at 10:36 am

Hi.
You can take a look at all our IFTTT projects:
https://randomnerdtutorials.com/?s=IFTTT
But unfortunately, we don't cover exactly what you're looking for.
Regards,
Sara

Reply

Steven
May 6, 2023 at 10:23 am

Hello,

thanks a lot for that tutorial. Due the fact, the JSON i requested is encoded as gzip I running into trouble using the response. How can i handle the request at a esp32?

Interface: http://s3.eu-central-1.amazonaws.com/app-prod-static.warnwetter.de/v16/forecast_mosmix_10487.json

Header of response:
Accept-Ranges bytes
Content-Encoding gzip
Content-Length 793
Content-Type application/json

…

Reply

Leave a Comment

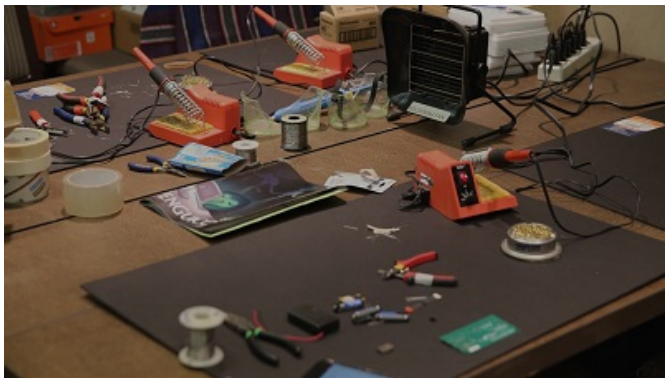
Name *

Email *

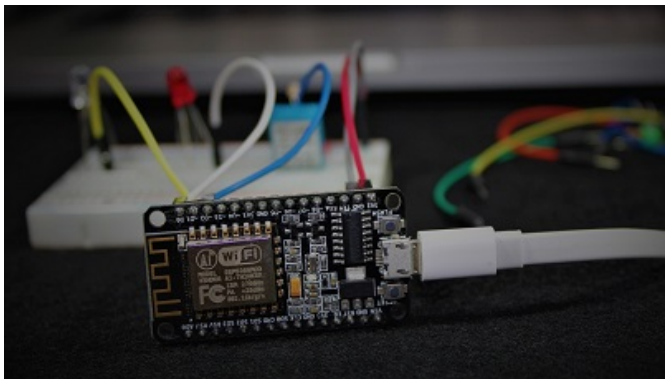
Website

☐ Notify me of follow-up comments by email.

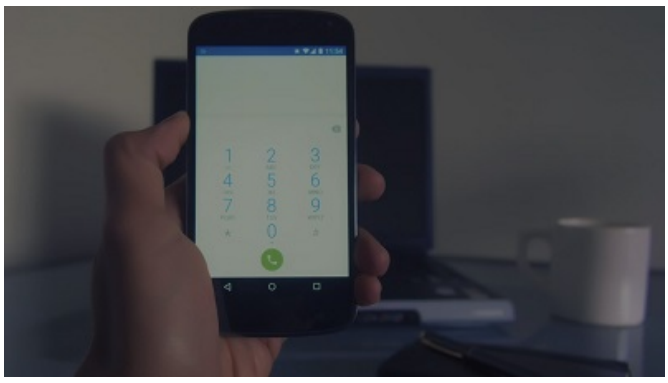
☐ Notify me of new posts by email.



Visit Maker Advisor – Tools and Gear for makers, hobbyists and DIYers »



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



Build Web Servers with ESP32 and ESP8266 » boards to control outputs and monitor sensors remotely.