

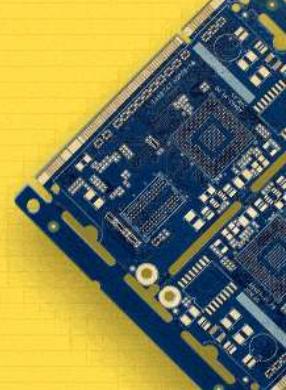
**NextPCB**

## Try Free PCB Prototype

**0\$ for 1-4 layer PCB Prototype**

- 1-4 layers, green mask, Fr-4
- Fast lead time as 24 hours

[Order Now](#)

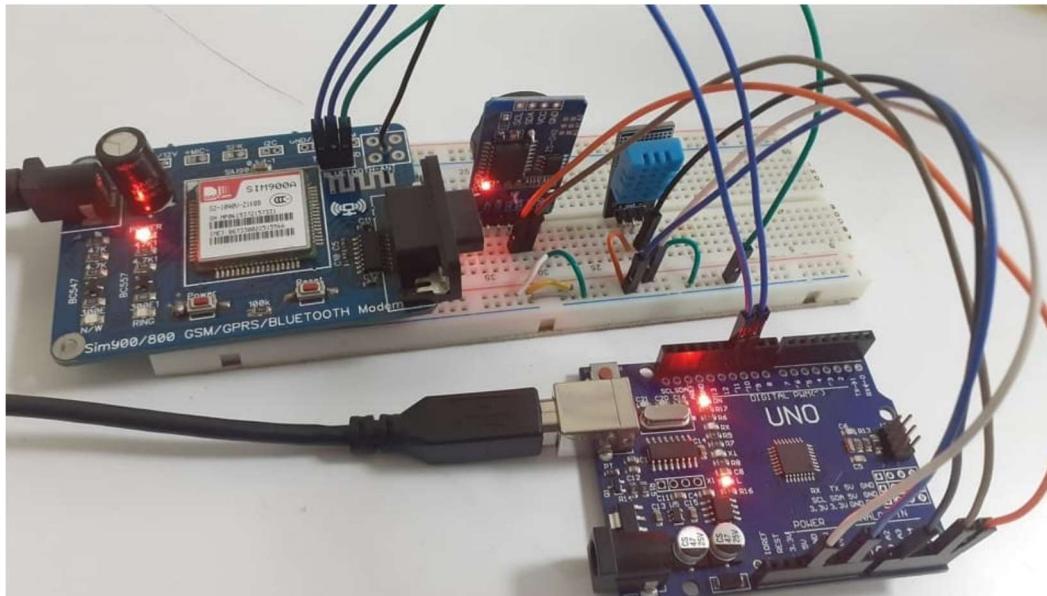


Arduino Projects   IoT Projects

# SIM900/800 HTTP Post Request in JSON Format with Arduino



Admin   Last Updated: August 21, 2022   4 comments   20,767 views   4 minutes read



In this post we will learn how to use SIM800 or SIM900 with Arduino & make HTTP Post Request in JSON format to any API Server.

Table of Contents [[hide](#)]

**EE Times**

The latest technology and engineering news in the electronics industry

[5 Hardware Setup](#)[6 Arduino JSON Library](#)[7 Source Code: SIM900/800 HTTP post request in JSON Format with Arduino](#)[8 Result & Responses](#)

## Overview

Apart from Wifi Module or Ethernet Module, the IoT Projects can also be done through SIM800/900 GSM GPRS Module. The main advantage of Cellular IoT compared to Wifi IoT is the availability of network and signal everywhere. In one of my earlier posts, I explained how you can send the sensor data wirelessly using GSM Module to Thingspeak Server. You can check the post here: [Send SIM800/900 GPRS Data to Thingspeak with Arduino](#)

But today we will interface the SIM800/900 GSM GPRS Module with Arduino and develop a code to send the data to any web address or web server using API. We need to send the data in JSON format as direct sending of a string data looks difficult. The whole process and the final code is explained in this post.

## Bill of Materials

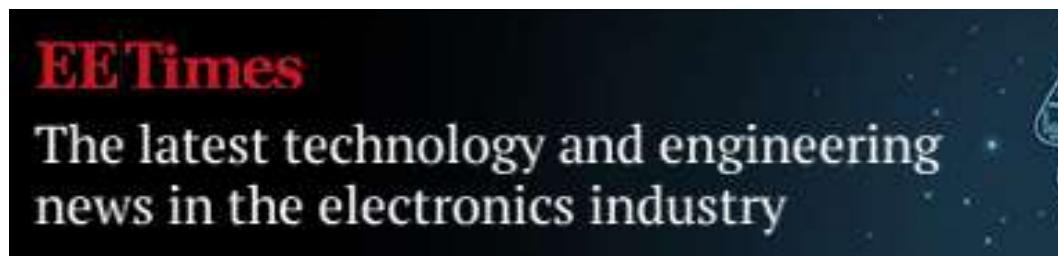
Following are the components required to learn and get practical experience of this tutorial. All the components can be easily





1	Arduino UNO	Arduino UNO R3 Development Board	1	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>
2	GSM Module	SIM800/SIM900 GSM GPRS Module	1	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>
3	DHT11 Sensor	DHT11 Humidity Temperature Sensor	1	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>

4	RTC Module	DS3231 Real Time Clock Module	1	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>
5	12/9V Power Supply	-	1	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>
6	Connecting Wires	Jumper Wires	10	<a href="https://amzn.to/3LcPQWz">https://amzn.to/3LcPQWz</a>



The latest technology and engineering news in the electronics industry

## Technology & Electronics

Learn From Industry Experts About N  
The Future Of The Electronics Indust

EE Times

### What is API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message or check the weather on your phone, you're using an API.

***Example of an API:*** When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is – all of this happens via API.

### What is JSON Format?

EE Times

# The latest technology and engineering news in the electronics industry

For example, the following is an example of a simple User object serialized to XML:

```
<xml>
  <user>
    <firstName>Jason</firstName>
    <middleName>Alexander</middleName>
    <lastName>Smith</lastName>
    <address>
      <street1>1234 Someplace Avenue</street1>
      <street2>Apt. 302</street2>
      <city>Anytown</city>
      <state>NY</state>
      <postalCode>12345</postalCode>
      <country>US</country>
    </address>
  </user>
</xml>
```

As you can see, the same data represented in JSON is far more efficient, while retaining all of its human-readability:

```
{
  "firstName" : "Jason",
  "middleName" : "Alexander",
  "lastName" : "Smith",
  "address" : {
    "street1" : "1234 Someplace Avenue",
    "street2" : "Apt. 302",
    "city" : "Anytown",
    "state" : "NY",
    "postalCode" : "12345",
    "country" : "US"
  }
}
```



The latest technology and engineering news in the electronics industry

In JSON, data is structured in a specific way. JSON uses symbols like { } , : " " [ ] and it has the following syntax:

*Data is represented in key/value pairs*

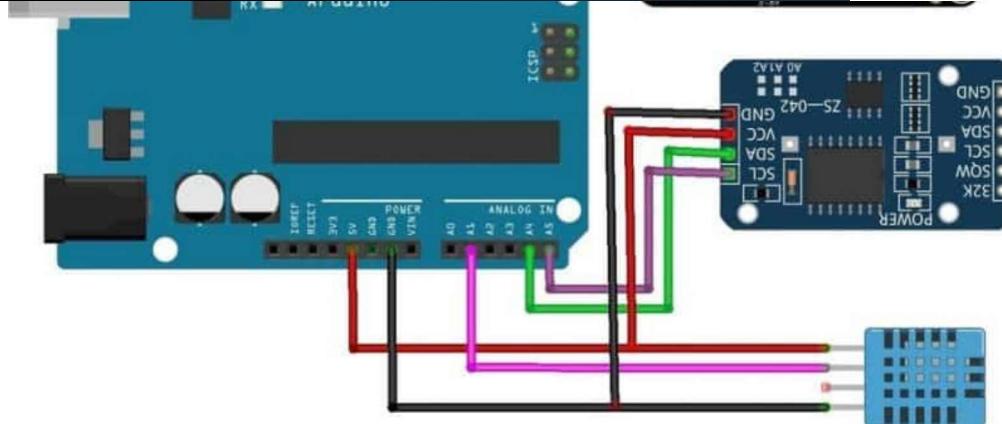
1. *The colon (:) assigns a value to key*
2. *key/value pairs are separated with commas (,)*
3. *Curly brackets hold objects ({ })*
4. *Square brackets hold arrays ([ ])*

## Hardware Setup

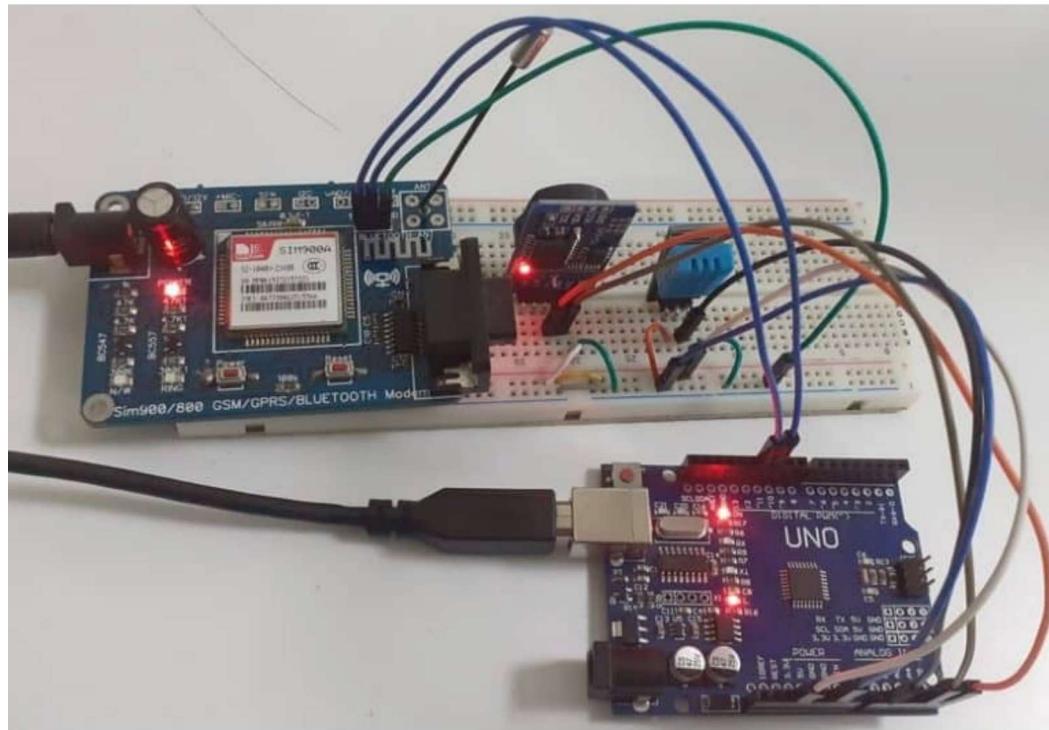
Let us do some hardware setup now. I am using 2 modules whose data is to be sent to the server via API in JSON format. I am using DHT11 Humidity & Temperature Sensor as I want to send Humidity & Temperature Data. Similarly, I am also using the DS3231 Real Time Clock (RTC) Module to check the time and send it to Server.

Following is the connection diagram below assemble the circuit as shown in the figure.

**EE Times**  
The latest technology and engineering news in the electronics industry



So I made the same connection here as shown below. So I have powered GSM Module with 12V and Arduino with 5V from computer USB port.



## Arduino JSON Library



The latest technology and engineering news in the electronics industry

Efficient memory management for Arduino. [Read more](#)

JSON library for Arduino and IoT (Internet Of Things).

## Features

1. *JSON decoding (comments are supported)*
2. *JSON encoding (with optional indentation)*
3. *Elegant API, very easy to use*
4. *Fixed memory allocation (zero mallocs)*
5. *No data duplication (zero-copy)*
6. *Portable (written in C++98)*
7. *Self-contained (no external dependency)*
8. *Small footprint*
9. *Header-only library*

---

## Technology & Electr News

Learn From Industry Experts Abc  
Innovations & The Future Of The  
Industry

EE Times

---

## Source Code: SIM900/800 HTTP post request in JSON Format with Arduino



following link and add to the Arduino IDE.

1. RTC Lib for DS3231: [Download](#)
2. DHT Library for DHT11 Sensor: [Download](#)
3. Arduino JSON Library: [Download](#)

In the following code, make changes to the APN. I used Airtel APN.

Check your Operator APN and make changes. Also make changes to the webserver or add your api webserver address where you want to send the data.

```
#include <RTClib.h> // Download the library from https://github.com/mkocabas/RTClib
#include <Wire.h> // Library for I2C Communication with DS3231
#include <DHT.h>
#include <SoftwareSerial.h>
SoftwareSerial myserial(10, 11); // RX: 10, TX:11
#include <ArduinoJson.h>
StaticJsonBuffer<200> jsonBuffer;

#define DHTPIN A1
#define DHTTYPE DHT11

RTC_DS3231 rtc;
DHT dht(DHTPIN, DHTTYPE);

char t[32];
char deviceID[12] = "MYTEST56";

void setup()
{
    myserial.begin(9600);           // the GPRS baud rate
```

**EE Times**

The latest technology and engineering news in the electronics industry

```
if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
}
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
//rtc.adjust(DateTime(2020, 02, 29, 17, 50, 40));
delay(5000);

}

void loop()
{
    Serial.println("");
    Serial.println("*****");
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

    DateTime now = rtc.now();

    sprintf(t, "%02d:%02d:%02d %02d/%02d/%02d", now.hour(), now.minute(), now.second(), now.month(), now.year(), now.day());

    Serial.print("Device ID: ");
    Serial.println(deviceID);
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.println(" °C");
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");
    Serial.print(F("Time/Date: "));
    Serial.println(t);
    delay(1000);

    ****GSM Communication Starts****

    if (myserial.available())
        Serial.write(myserial.read());
```

**EE Times**

The latest technology and engineering news in the electronics industry

```
ShowSerialData();

myserial.println("AT+SAPBR=3,1,\"APN\",\"airtelgprs.com\"");
delay(6000);
ShowSerialData();

myserial.println("AT+SAPBR=1,1");
delay(6000);
ShowSerialData();

myserial.println("AT+SAPBR=2,1");
delay(6000);
ShowSerialData();

myserial.println("AT+HTTPINIT");
delay(6000);
ShowSerialData();

myserial.println("AT+HTTPPARA=\"CID\",1");
delay(6000);
ShowSerialData();

StaticJsonBuffer<200> jsonBuffer;
JsonObject& object = jsonBuffer.createObject();

object.set("deviceID", deviceID);
object.set("humidity", humidity);
object.set("temperature", temperature);
object.set("timedate", t);

object.printTo(Serial);
Serial.println(" ");
String sendtoserver;
object.prettyPrintTo(sendtoserver);
delay(4000);

myserial.println("AT+HTTPPARA=\"URL\",\"http://192.xxxxxxxx")
```

**EE Times**

The latest technology and engineering news in the electronics industry

```
myserial.println("AT+HTTPDATA=" + String(sendtoserver.length));
Serial.println(sendtoserver);
delay(6000);
ShowSerialData();

myserial.println(sendtoserver);
delay(6000);
ShowSerialData;

myserial.println("AT+HTTPACTION=1");
delay(6000);
ShowSerialData();

myserial.println("AT+HTTPREAD");
delay(6000);
ShowSerialData();

myserial.println("AT+HTTPTERM");
delay(10000);
ShowSerialData;

/*******************GSM Communication Stops******************/

}

void ShowSerialData()
{
    while (myserial.available() != 0)
        Serial.write(myserial.read());
    delay(1000);

}
```



## Result & Responses

Once you upload the code to the Arduino and Power on the module, the Module will try connecting to the internet and will send data in JSON format.

Sometimes you might miss data and get some errors. This can be reduced by increasing delays after step by step in the above code. Or if the network connectivity and signal strength is good you can get the data most of the times.

Open your serial monitor and you should get the following results.



```

OK
AT

OK
AT+SAPBR=3,1,"Contype","GPRS"

OK
AT+SAPBR=3,1,"APN","airtelgprs.com"

OK
AT+SAPBR=1,1

ERROR
AT+SAPBR=2,1

+SAPBR: 1,1,"10.86.200.94"

OK
AT+HTTPINIT

OK
AT+HTTPPARA="CID",1

 Autoscroll  Show timestamp Newline 115200 baud Clear output

```

COM12

```

AT+HTTPPARA="CID",1

OK
{"deviceID": "MYTEST56", "humidity": 39.00, "temperature": 23.00, "timedate": "17:03:07 10/03/2020"},  

,"http://120.████████:8080/████████service1.php"

OK
AT+HTTPPARA="CONTENT", "application/json"

OK
{
  "deviceID": "MYTEST56",
  "humidity": 39.00,
  "temperature": 23.00,
  "timedate": "17:03:07 10/03/2020"
}
AT+HTTPDATA=115,100000

DOWNLOAD

OK
AT+HTTPACTION=1

OK

+HTTPACTION:1,200,184
b>22</b><br />
("deviceID": "MYTEST56", "status": "SUCCESS")
OK

 Autoscroll  Show timestamp Newline 115200 baud Clear output

```

So this was all about how to make to use SIM900/800 Module & make HTTP post request in JSON Format with Arduino.



## Electronics News & Analysis

Ad EE Times

## Engineering Articles

Ad Power Electronics News

## Genge & Thoma Ag potentiometer - Felicity Control Limited

Ad felicitycontrol.com

## LoRa

Ad fou

## IC Thermal Testing

Ad eldrotec LTD

## 4 Comments





The latest technology and engineering news in the electronics industry

EE Times

Loading...

Reply

---



### Eugene

October 1, 2020 at 4:00 PM

Hello, I would like to ask, why did you define three jsonbuffers

Loading...

Reply

---



### Vasekar

September 2, 2021 at 8:49 PM

Thanks, this is amazing, Relay you have done the extensive efforts for creating step by step connection and data sending commands. so many times serial monitor displays OK. I could make the HTTP POST request to my own website api using this program.

A small request, there is compilation error for Json Object. It is necessary to upgrade for ver. 6. For my work insted of creating JSON data string using JSON object, I have assigned it directly. However If possible, please try to upgrade the code.

Loading...

Reply

---



### Bobby

April 15, 2022 at 4:21 PM

Reply



