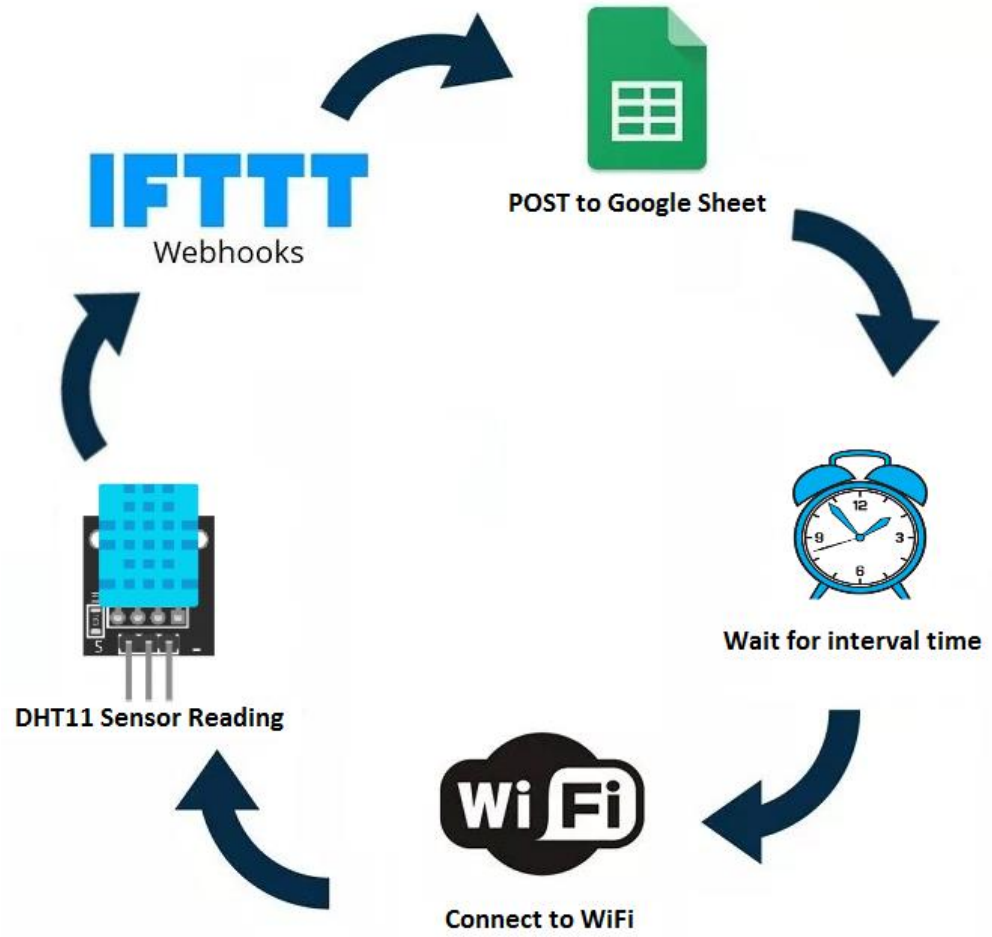
ESP8266 – IFTTT
Publish Sensor Readings to Google Sheets

We will see how to publish sensor readings to Google Sheets using ESP8266 board. As an example, we'll publish temperature C, temperature F and humidity readings using the DHT11 sensor to a Google Sheets spreadsheet every 1 minute – we'll be using IFTTT.



- First, the ESP connects to your Wi-Fi network;
- Then, it takes the temperature, humidity readings from sensor;
- The ESP8266 communicates with the IFTTT Webhooks service that publishes the readings to a spreadsheet on Google Sheets that is saved in your Google Drive's folder;
- After publishing the readings, the ESP waits for time interval to pass and repeats the process;

Creating Your IFTTT Account

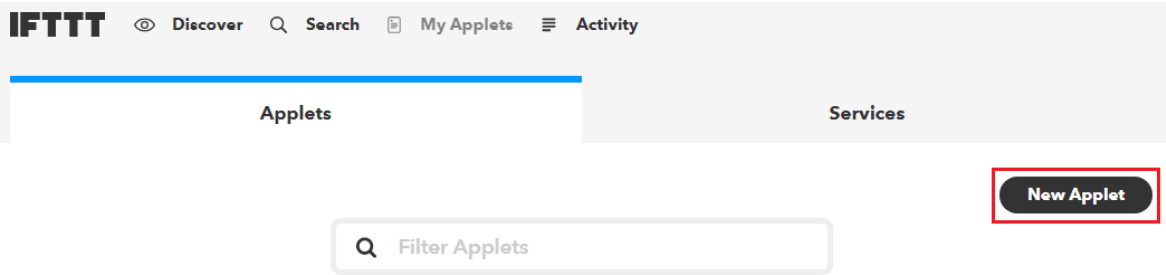
For this project we'll be using IFTTT to integrate with Google Sheets. So, the first step is creating an account on IFTTT if you don't have one. Creating an account on IFTTT is free!

Go the official site: ifttt.com and enter your email to get started.

Creating an Applet

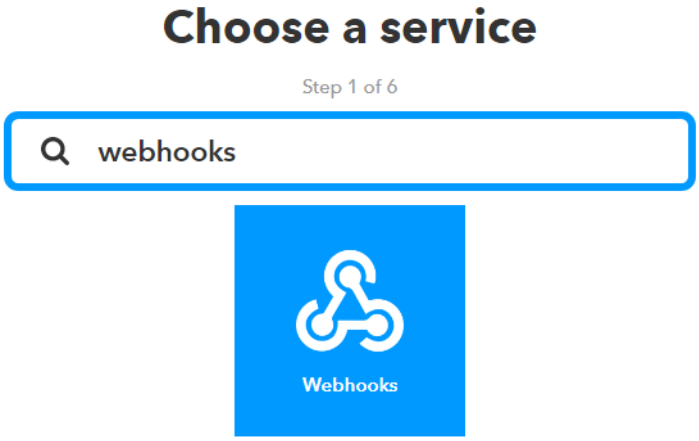
Next, you need to create a new applet. Follow the next steps to create a new applet:

1) Go to "My Applets" and create a new applet by clicking the "New Applet" button.

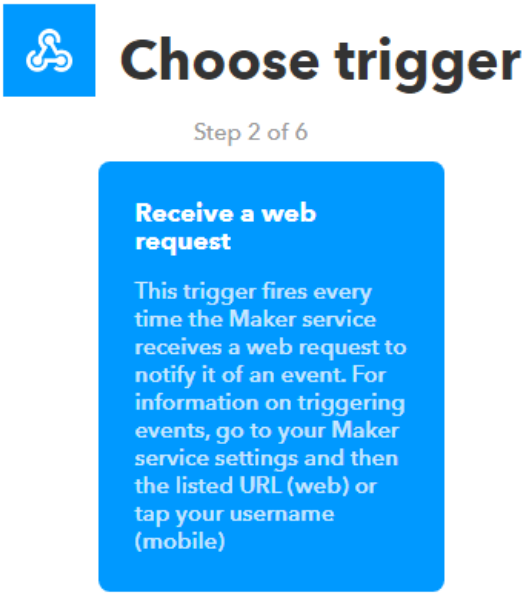


2) Click on the "this" word that is in a blue color – as highlighted in the figure below.

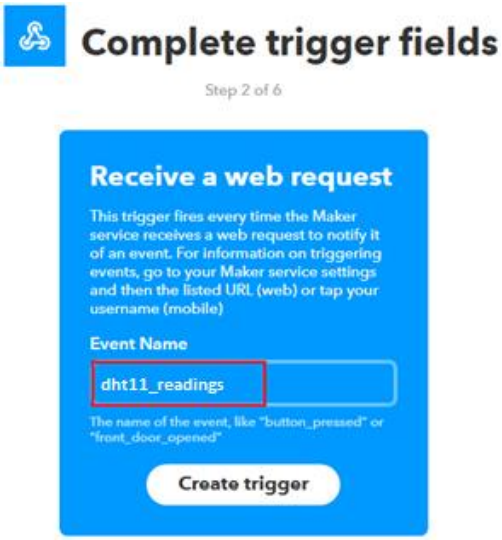
3) Search for the “Webhooks” service and select the Webhooks icon.



4) Choose the “Receive a web request” trigger.

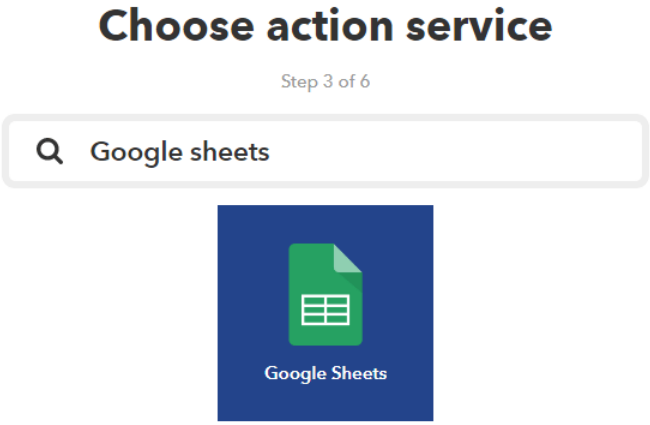


5) Give a name to the event. In this case “**dht11_readings**” as shown in the figure below. Then, click the “Create trigger” button.



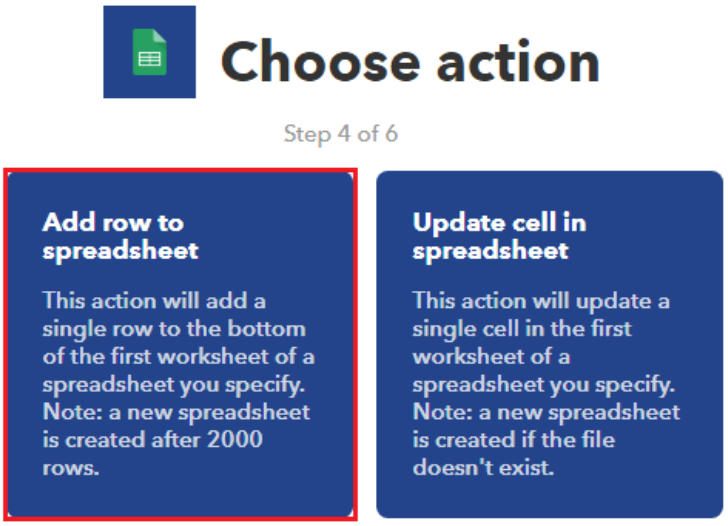
6) Click the “that” word to proceed.

7) Search for the “Google Sheets” service, and select the Google Sheets icon.

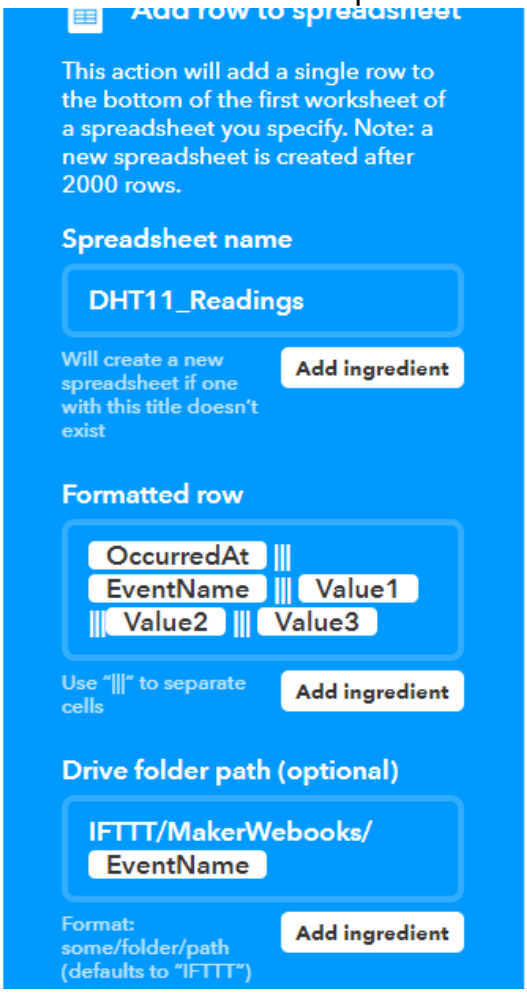


8) If you haven’t connected with the Google Sheets service yet, you need to click the “Connect” button.

9) Choose the “Add a row to spreadsheet” action.



10) Then, complete the action fields. Give the spreadsheet a name, leave the "Formatted row" field as default, and then, choose a Google Drive folder path. If you leave this field empty, IFTTT will create a folder called "IFTTT" in your Google Drive folder to save the spreadsheet. Finally, click the "Create action" button.

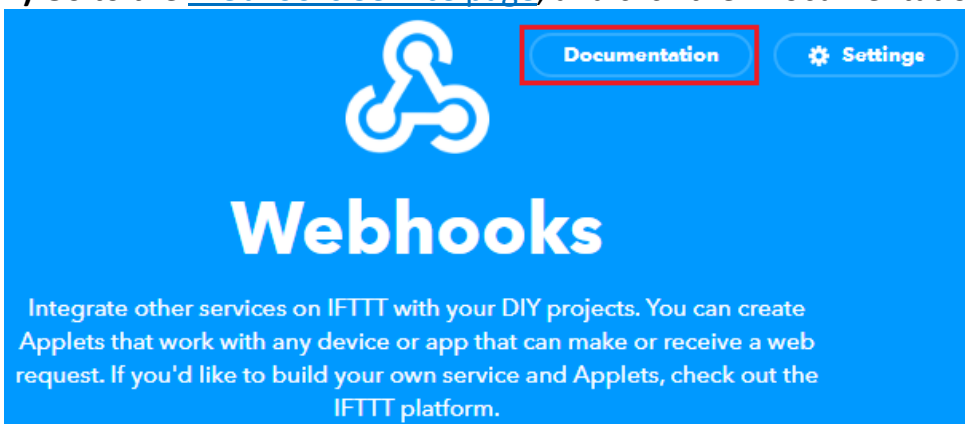


11) Your applet should be created after you press the "Finish" button.

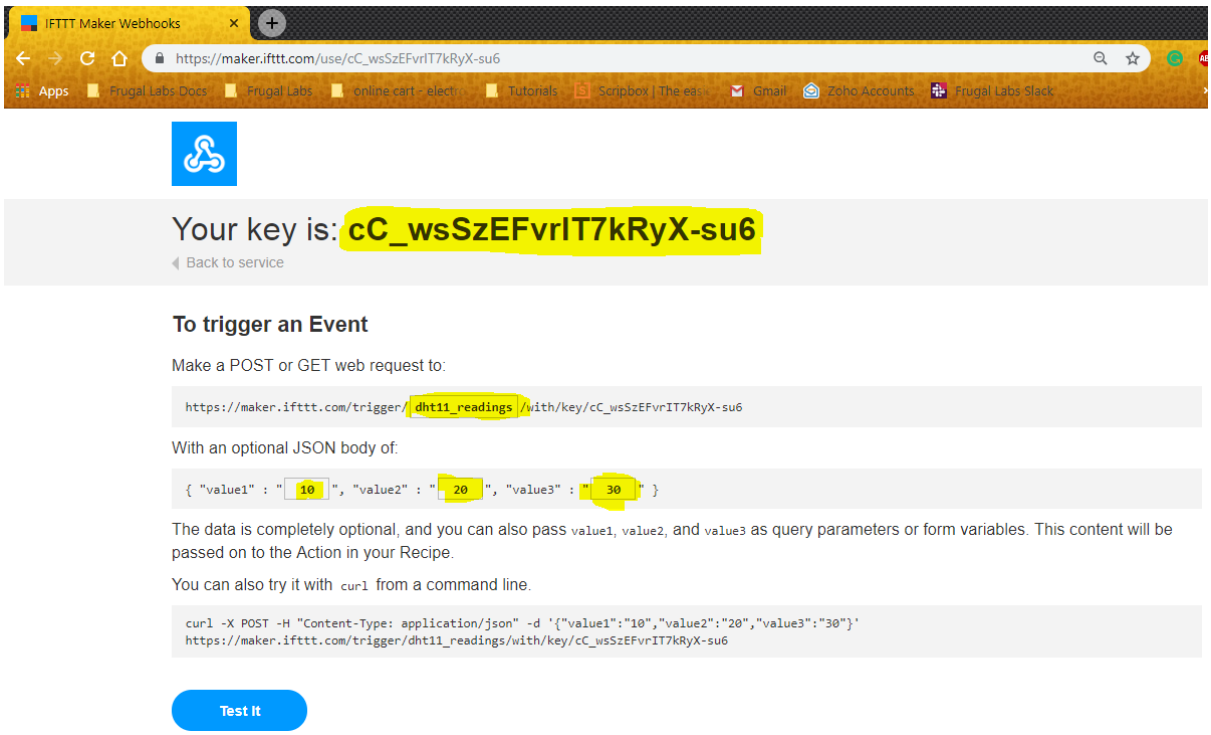
Testing Your Applet

Before proceeding with the project, it is very important to test your applet first. Follow the next steps to test your applet.

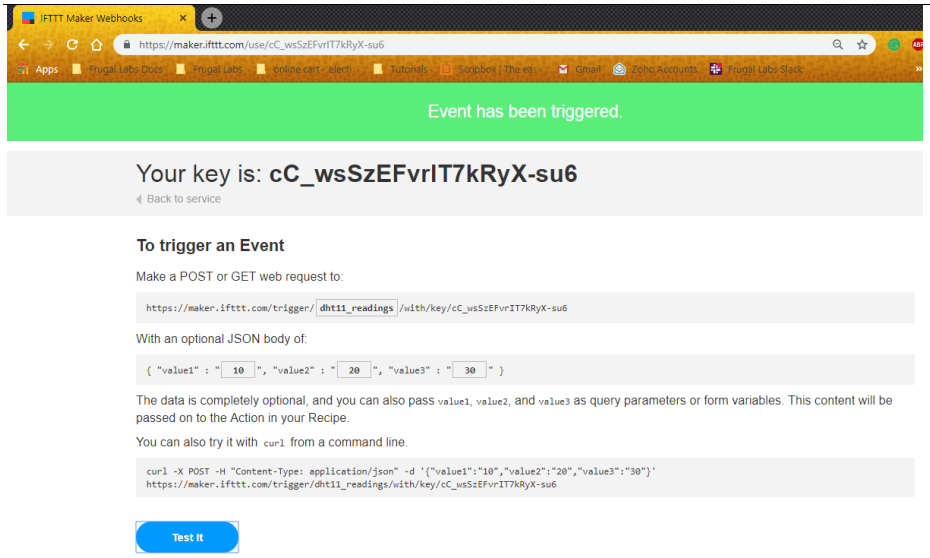
1) Go to the Webhooks Service page, and click the "Documentation" button.



2) A page as shown in the following figure will appear. The page shows your unique API key. You shouldn't share your unique API key with anyone.
Fill the "To trigger an Event" section as shown below – it is highlighted with red rectangles. Then, click the "Test it" button.

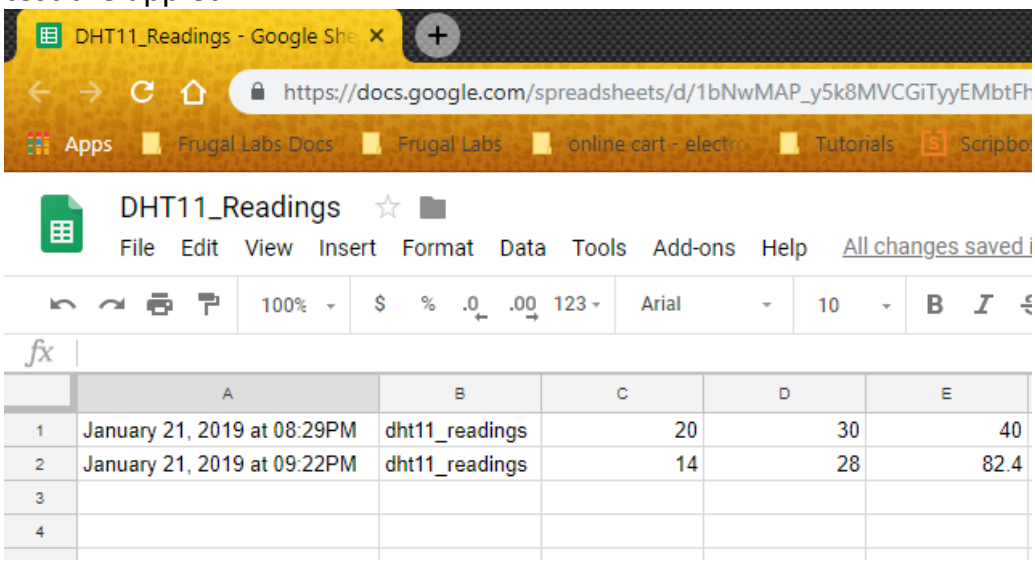


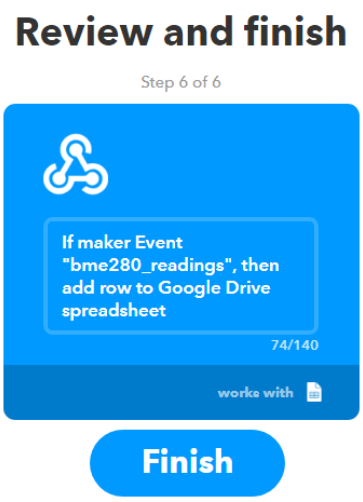
3) The event should be successfully triggered, and you'll get a green message as shown below saying "Event has been triggered".



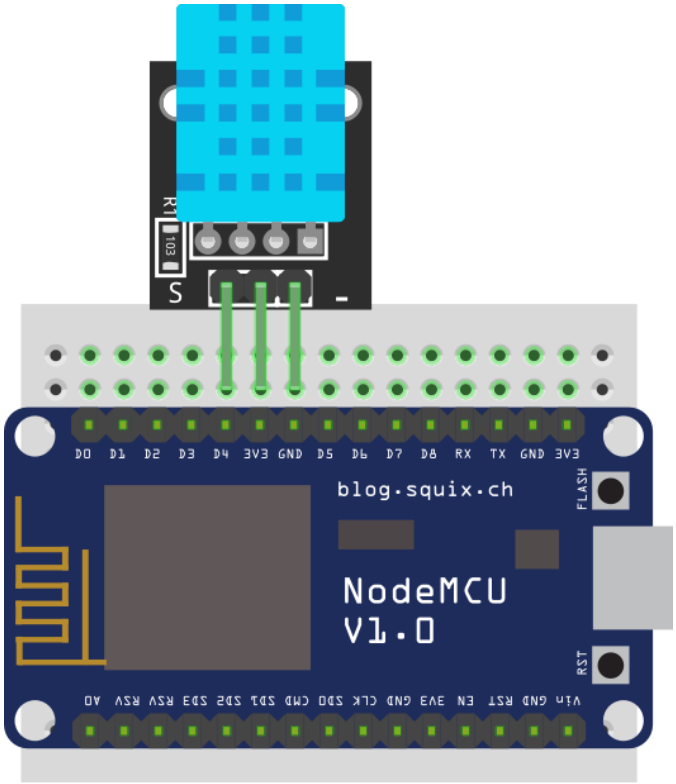
4) Go to your Google Drive. The IFTTT service should have created a folder called "IFTTT" with the "DHT11_Readings" spreadsheet inside.

5) Open the spreadsheet, and you should see the values you've filled previously to test the applet.





Rigging up Your Circuit




```
#include <ESP8266WiFi.h>
#include <DHT.h>
#define DHTTYPE DHT11 // DHT11 or DHT22
#define DHTPIN  D4 //D3 on nodemcu
DHT dht(DHTPIN, DHTTYPE, 11);

// Replace with your SSID and Password
const char* ssid     = "flip-test";
const char* password = "flip1234";

// Replace with your unique IFTTT URL resource
const char* resource = "/trigger/dht11_readings/with/key/cC_wsSzEFvrIT7kRyX-su6";

// Maker Webhooks IFTTT
const char* server = "maker.ifttt.com";

unsigned long nowTime = 0;
unsigned long lastTime = 0;
long interval = 60; // - 60 seconds between reports
float hum, tc, tf;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  delay(2000);
  dht.begin();
  delay(10);
  wifiConnect();
  //makeIFTTTRequest();
}

void loop()
{
  if (WiFi.status() == WL_CONNECTED)
```

```
  {
    nowTime = millis();
    unsigned long x = (nowTime - lastTime);
    Serial.println("--------------------------------");
    Serial.print(F("POSTING Time (sec) = "));
    Serial.println(interval);
    Serial.print(F("last POST (sec) = "));
    Serial.println(x / 1000);
    hum = dht.readHumidity();
    tc = dht.readTemperature();
    tf = dht.readTemperature(true);
    Serial.println("Humidity = " + String(hum));
    Serial.println("Temperature *C = " + String(tc));
    Serial.println("Temperature *F = " + String(tf));
    Serial.println("--------------------------------");
    if (x > (interval * 1000))
    {
      Serial.println("Time to post");
      lastTime = nowTime;
      makeIFTTTRequest();
    }
  }
  else
  {
    wifiConnect();
  }
  delay(2000);
}

// Establish a Wi-Fi connection with your router
void wifiConnect()
{
  Serial.print("Connecting to: ");
  Serial.print(ssid);
  WiFi.begin(ssid, password);

  int timeout = 20 * 4; // 20 seconds
  while (WiFi.status() != WL_CONNECTED  && (timeout-- > 0)) {
    delay(250);
    Serial.print(".");
  }
  Serial.println("");

  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Failed to connect, going back to sleep");
  }

  Serial.print("WiFi connected in: ");
  Serial.print(millis());
  Serial.print(", IP address: ");
  Serial.println(WiFi.localIP());
}

// Make an HTTP request to the IFTTT web service
void makeIFTTTRequest()
{
  Serial.print("Connecting to ");
  Serial.print(server);

  WiFiClient client;
  int retries = 5;
  while (!!!client.connect(server, 80) && (retries-- > 0)) {
    Serial.print(".");
  }
  Serial.println();
  if (!!!client.connected()) {
    Serial.println("Failed to connect...");
  }
```

```
  Serial.print("Request resource: ");
  Serial.println(resource);
  String jsonObject = String("{\"value1\":\"") + (hum) + "\",\"value2\":\"" + (tc) + "\",\"value3\":\"" + (tf) + "\"}";
  client.println(String("POST ") + resource + " HTTP/1.1");
  client.println(String("Host: ") + server);
  client.println("Connection: close\r\nContent-Type: application/json");
  client.print("Content-Length: ");
client.println(jsonObject.length());
  client.println();
  client.println(jsonObject);

  int timeout = 5 * 10; // 5 seconds
  while (!!!client.available() && (timeout-- > 0))
  {
    delay(100);
  }
  if (!!!client.available())
  {
    Serial.println("No response...");
  }
  while (client.available())
  {
    Serial.write(client.read());
  }

  Serial.println("\nclosing connection");
  client.stop();
}
```

Note: There is a time interval limit for using IFTTT service. If you exceed the number of triggers in a day for an applet, it will get blocked for that day.