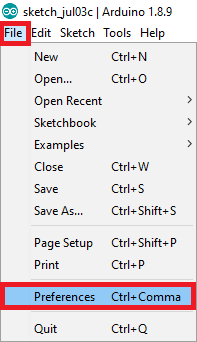
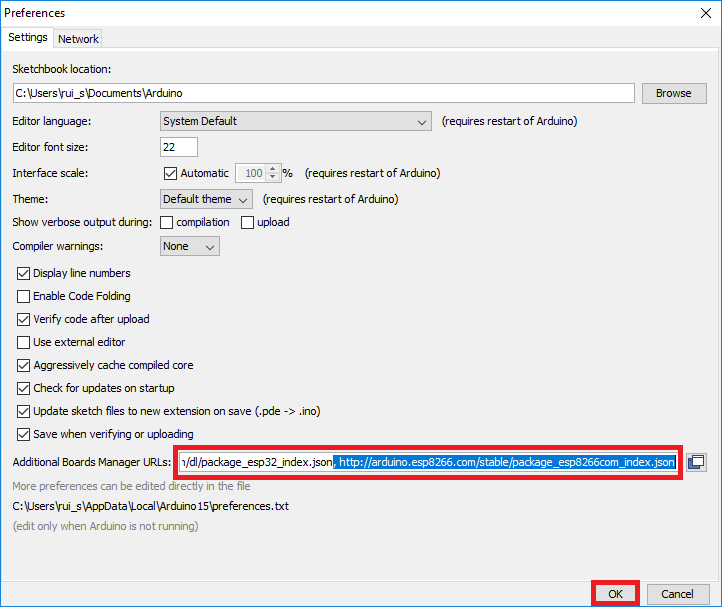
**Install ESP8266 Add-on in Arduino IDE**

To install the ESP8266 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to **File**> **Preferences**



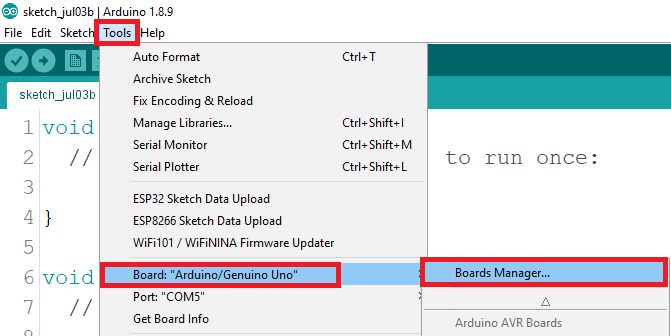
1. Enter **http://arduino.esp8266.com/stable/package\_esp8266com\_index.json** into the “Additional Boards Manager URLs” field as shown in the figure below. Then, click the “OK” button:



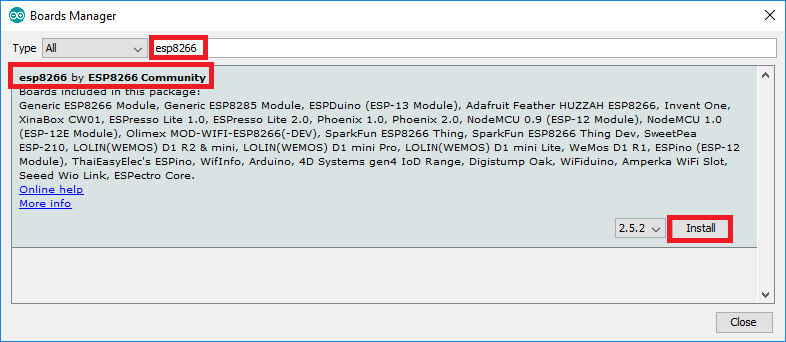
**Note:** if you already have the ESP32 boards URL, you can separate the URLs with a comma as follows:

https://dl.espressif.com/dl/package\_esp32\_index.json, http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

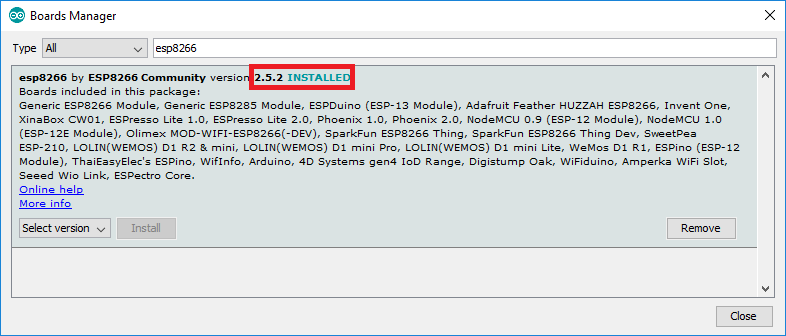
1. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**



1. Search for **ESP8266** and press install button for the “**ESP8266 by ESP8266 Community**“:



1. That’s it. It should be installed after a few seconds.



Error Missing DHT.h

Sketch -> Include Library > Manage Libraries

Library Manager -> Search for Adafruit dht

A screenshot of a computer

Description automatically generated

Code

#include <ESP8266HTTPClient.h>

#include <ESP8266WiFi.h>

#include "DHT.h"

// Uncomment one of the lines below for whatever DHT sensor type you're using!

#define DHTTYPE DHT11   // DHT 11

//#define DHTTYPE DHT21   // DHT 21 (AM2301)

//#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

// Replace with your network details

const char\* ssid = "XXXX";

const char\* password = "XXX";

#define SERVER\_IP "192.168.0.XXX"

#define PORT "XXX"

const String SENSOR\_ID = "3";

// DHT Sensor

// For ESP01s set DHTPin = 2

const int DHTPin = 2;

// Initialize DHT sensor.

DHT dht(DHTPin, DHTTYPE);

void setup() {

  //Serial.begin(115200); //Serial connection

  delay(10);

  dht.begin();

  // Connecting to WiFi network

  //Serial.println();

  //Serial.print("Connecting to ");

  //Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {  //Wait for the WiFI connection completion

    delay(500);

    //Serial.println("Waiting for connection");

  }

  //Serial.println("");

  //Serial.println("WiFi connected");

}

void loop() {

  if (WiFi.status() == WL\_CONNECTED) { //Check WiFi connection status

    WiFiClient client;

    HTTPClient http;    //Declare object of class HTTPClient

    //Serial.print("[HTTP] begin...\n");

    // configure traged server and url

    //http.begin(client, "http://" SERVER\_IP ":" PORT "/api/data/"); //HTTP

    http.begin(client, "http://" SERVER\_IP "/api/data/"); //HTTP

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

    String celsiusTemp;

    String fahrenheitTemp;

    String humidity;

    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

    float h = dht.readHumidity();

    // Read temperature as Celsius (the default)

    float t = dht.readTemperature();

    // Read temperature as Fahrenheit (isFahrenheit = true)

    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {

      Serial.println("Failed to read from DHT sensor!");

      celsiusTemp = "";

      fahrenheitTemp = "";

      humidity = "";

    } else {

      float hic = dht.computeHeatIndex(t, h, false);

      celsiusTemp = String(hic);

      float hif = dht.computeHeatIndex(f, h);

      fahrenheitTemp = String(hif);

      humidity = String(h);

    }

    //Serial.print("[HTTP] POST...\n");

    // start connection and send HTTP header and body

    String response = "{\"sensor\": " + SENSOR\_ID + ", \"temp\_c\": " + celsiusTemp + ", \"temp\_f\": " + fahrenheitTemp + ", \"humidity\" : " + humidity + "}";

    int httpCode = http.POST(response);

    //Get the response payload

    const String& payload = http.getString();

    //Serial.println(httpCode);   //Print HTTP return code

    //Serial.println(payload);    //Print request response payload

    http.end();  //Close connection

  } else {

    //Serial.println("Error in WiFi connection");

  }

  delay(600000);  //Send a request every 600 seconds

}

**Circuit Setup**

Breadboard Schematic Diagram shown below are set up to flash the ESP-01S module. Arduino IDE is used to upload the sketch and flash the ESP-01S module.

