

# 7.3 Web Communication with HTTP/HTTPS

## HTTP

HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the World Wide Web. It operates on a request-response model.

- Client (ESP32): The client make a request to the server for a resource, like a webpage or data.
- Server (A Web Server): The server takes your request, process it, and send the processed request back to you as a response.

## HTTPS

HTTPS is simply HTTP Secure. It adds a layer of SSL/TLS encryption to the conversation. This prevents anyone from eavesdropping on the data exchanged between the client and server, which is crucial for protecting sensitive information like passwords or personal data.

## Example

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

// --- Replace with your network credentials ---
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";
// -----

// The URL of the API we want to request data from
const char* api_url = "http://jsonplaceholder.typicode.com/posts/1";

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

```

    delay(500);
    Serial.print(".");
}
Serial.println("\nWiFi connected!");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
}

void loop() {
    // Check if we are connected to WiFi before proceeding
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        Serial.println("Making HTTP GET request...");
        http.begin(api_url); // Initialize the HTTP request

        int httpCode = http.GET(); // Send the GET request

        if (httpCode > 0) { // Check if the request was successful
            Serial.printf("HTTP Response code: %d\n", httpCode);

            if (httpCode == HTTP_CODE_OK) {
                String payload = http.getString(); // Get the response payload as a string

                // --- Parse the JSON response ---
                JsonDocument doc;
                DeserializationError error = deserializeJson(doc, payload);

                if (error) {
                    Serial.print("deserializeJson() failed: ");
                    Serial.println(error.c_str());
                }
                else {
                    // Extract the value associated with the "id" and "title" key

                    int postId = doc["id"];
                    const char* title = doc["title"];

                    Serial.printf("Post ID: %d\n", postId);
                    Serial.printf("Title: %s\n", title);
                }
            }
        }
    }
}

```

```
    }  
  } else {  
    Serial.printf("HTTP GET request failed, error: %s\n",  
http.errorToString(httpCode).c_str());  
  }  
  
  http.end(); // Free resources  
} else {  
  Serial.println("WiFi not connected.");  
}  
  
// Wait 30 seconds before the next request  
delay(30000);  
}
```

---

Revision #2

Created 2025-08-28 16:38:21 UTC by NS

Updated 2025-10-01 07:29:37 UTC by NS