

# NodeMCU: obteniendo datos desde una dirección HTTPS con un JSON. Conexiones seguras. (III)

[agrportfolioeducativo.blogspot.com/2019/07/nodemcu-obteniendo-datos-desde-una-2.html](http://agrportfolioeducativo.blogspot.com/2019/07/nodemcu-obteniendo-datos-desde-una-2.html)

Empezamos nuestro programa con el acceso a las librerías. Añadimos, respecto al paso anterior, una librería más: **WiFiClientSecure**

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h> // Incluimos la biblioteca WiFiClientSecure.h
```

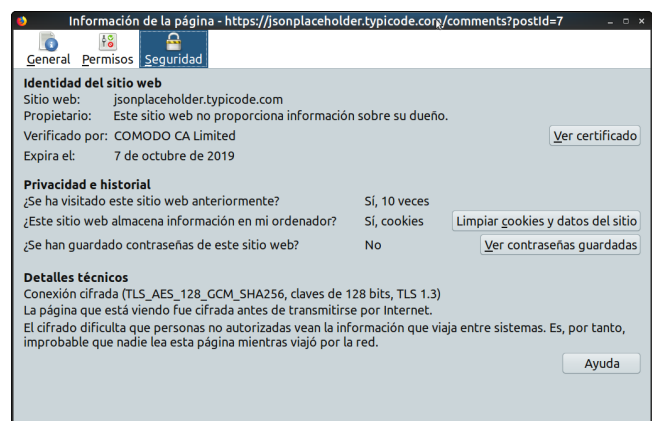
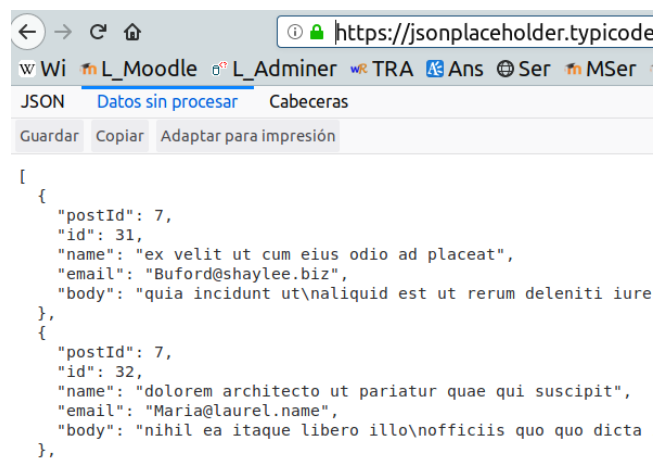
Lo primero que tenemos que hacer es comprobar que a nuestra petición JSON se accede a través de un navegador. En nuestro caso, nuestra URL es

**<https://jsonplaceholder.typicode.com/>** y le añadiremos la petición **comments?postId=7**. Al añadirlo al navegador comprobamos que, en efecto, funciona.

Tenemos que buscar ahora la huella o fingerprint SHA1 de esta petición. En mi caso, en firefox, busco el símbolo del candado al lado de la URL y pulso en CONEXIÓN SEGURA (flecha a la derecha), y en la siguiente ventana en **MÁS INFORMACIÓN**

Y aquí en **VER CERTIFICADO**.  
Buscamos y copiamos la huella SHA1.

Que en nuestro caso es



**"oE:81:AA:54:2C:1A:AC:BA:15:A8:92:AD:62:32:59:1B:B2:E8:0D:9E"**

== == == == ==

Seguimos con el programa en el IDE de ARDUINO...

```

constchar* ssid = "miSSID";// Rellena con el nombre de tu red WiFi
constchar* password = "miCONTRASEÑA";// Rellena con la contraseña de tu red WiFi

constchar*host = "jsonplaceholder.typicode.com";// incluimos el host
constint httpsPort = 443;//HTTPS= 443 and HTTP = 80

const char fingerprint[] PROGMEM = "0E 81 AA 54 2C 1A AC BA 15 A8 92 AD 62 32 59
1B B2 E8 0D 9E";

```

Creamos tres constantes de tipo carácter. Una con el nombre del **host**, otra que contenga el número del puerto **httpsPort** (HTTPS es **443**) y otra que almacene la huella. El array que almacena la huella (**fingerprint**) se escribirá en la memoria flash (**PROGMEM**)

= = = = =

## SETUP()

---

Análogamente al programa anterior, consigo la conexión a la red Wifi desde SETUP

```

void setup()
{
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.println("Conectando...");
  }
}

```

= = = = =

## LOOP()

---

Y si consigo la conexión se ejecuta el programa...

```

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

```

.....

```

    }
    delay(10000);
}

```

¿Cómo me conecto a la información por HTTPS? Dentro del if, incluyo el siguiente código:

### A) Declaro objeto del tipo WiFiClientSecure

---

```
WiFiClientSecure httpsClient; //Declare object of class WiFiClient
```

```

Serial.println(host);

    Serial.printf("Using fingerprint '%s'\n", fingerprint);
httpsClient.setFingerprint(fingerprint);
httpsClient.setTimeout(15000); // 15 Seconds
delay(1000);

```

Declaro el objeto httpsClient de la clase WiFiClientSecure. Establezco su huella con el método setFingerPrint y establezco en 15 segundos su TimeOut. No estoy muy seguro qué significa exactamente este parámetro. Supongo que a los 15 segundos rompe la conexión automáticamente. ¿?

### B) Intentando conectarme al cliente HTTPS

---

```

Serial.print("HTTPS Connecting");
int r=0; //retry counter
while(!httpsClient.connect(host, httpsPort)) && (r < 30)){
    delay(100);
    Serial.print("#");
    r++;
}

```

Intenta, con un retardo de 100 ms, conectarse al cliente en el host y con el puerto 443. Lo repite 30 veces.

### C) Si r es menor que 30...

---

Si tengo conexión, o sea, no he llegado a 30...

```

if(r==30) {
    Serial.println("Fallo de conexión");
}
else {
    Serial.println("¡¡Conectado a la web!!");
}

```

.....

}

#### **D) Continuo en los puntos suspensivos. Construyo la cadena GET y envío la petición.**

---

```
//GET Data
Link = "/comments?postId=7";

Serial.print("requesting URL: ");
Serial.println(host+Link);

httpsClient.print(String("GET ") Link " HTTP/1.1\r\n"
    "Host: " host "\r\n"
    "Connection: close\r\n\r\n");

Serial.println("enviada petición");
```

#### **E) Leo la cabecera y las ignoro.**

---

```
// Recibiendo las cabeceras
while (httpsClient.connected()) {
    line = httpsClient.readStringUntil('\n');
    if (line == "\r") {
        Serial.println("cabeceras recibidas");
        break;
    }
}
```

#### **F) Leo la información y la imprimo por puerto en serie**

---

```
// Recibiendo la información
Serial.println("La respuesta fue:");
Serial.println("=====");

while(httpsClient.available()){
    line = httpsClient.readStringUntil('\n');//Read Line by Line
    Serial.println(line);//Print response
}

Serial.println("=====");
Serial.println("cerrando conexión");
```

Y cierro la conexión.

= = = = =

## Programa completo.

---

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h> // Incluimos la biblioteca WiFiClientSecure.h
// #include <ArduinoJson.h>

// https://circuits4you.com/2019/01/11/nodemcu-esp8266-arduino-json-parsing-
// example/
// https://circuits4you.com/2019/01/10/esp8266-nodemcu-https-secured-get-request/ -
-> https

constchar* ssid = "miSSID"; // Rellena con el nombre de tu red WiFi
constchar* password = "miCONTRASEÑA"; // Rellena con la contraseña de tu red WiFi

constchar* host = "jsonplaceholder.typicode.com"; // incluimos el host
constint httpsPort = 443; // HTTPS = 443 and HTTP = 80

int n = 1;

// https://jsonplaceholder.typicode.com/comments?postId=7
// Huella--> 0E:81:AA:54:2C:1A:AC:BA:15:A8:92:AD:62:32:59:1B:B2:E8:0D:9E

constchar fingerprint[] PROGMEM = "0E 81 AA 54 2C 1A AC BA 15 A8 92 AD 62 32 59 1B
B2 E8 0D 9E";

void setup()
{
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while(WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.println("Conectando...");
  }
}
```

```

void loop()
{

if(WiFi.status()== WL_CONNECTED)
{

    WiFiClientSecure httpsClient;//Declare object of class WiFiClient


    Serial.println(host);


    Serial.printf("Usando la huella %s'\n", fingerprint);
    httpsClient.setFingerprint(fingerprint);
    httpsClient.setTimeout(15000);// 15 Seconds
    delay(1000);


    Serial.print("HTTPS Conectando...");
    int r=0;//retry counter
    while((!httpsClient.connect(host, httpsPort))&&(r <30)){
        delay(100);
        Serial.print("#");
        r++;
    }


    if(r==30){
        Serial.println("Fallo de conexión");
    }
    else{
        Serial.println("¡¡Conectado a la web!!");
        String Link,line;

        //GET Data
        Link = "/comments?postId="

        String(n);

        Serial.print("Preguntamos por la URL: ");
        Serial.println(host

```

Link);

```
httpsClient.print(String("GET ") Link " HTTP/1.1\r\n"
    "Host: " host "\r\n"
```

"Connection: close\r\n\r\n");

Serial.println("petición enviada");

```
// Recibiendo las cabeceras
while(httpsClient.connected()){
    line = httpsClient.readStringUntil('\n');
    if(line == "\r"){
        Serial.println("cabeceras recibidas");
        break;
    }
}
```

```
// Recibiendo la información
Serial.println("Respuesta:");
Serial.println("=====");
```

```
while(httpsClient.available()){
    line = httpsClient.readStringUntil('\n');//Read Line by Line
    Serial.println(line);//Print response
}

Serial.println("=====");
Serial.println("Cerrando conexión");
}
```

}

delay(10000);

```
n = (n1)*(n<7)1*(n>=7);
}
```

El próximo paso es trabajar con la información tipo JSON y extraer datos de la misma.

**NOTA: el programa está levemente modificado para que obtenga en secuencia varias series de datos cambiando la variable postID.**

Direcciones útiles:

<https://circuits4you.com/2019/01/10/esp8266-nodemcu-https-secured-get-request/>

<https://circuits4you.com/2019/01/11/nodemcu-esp8266-arduino-json-parsing-example/>

=====