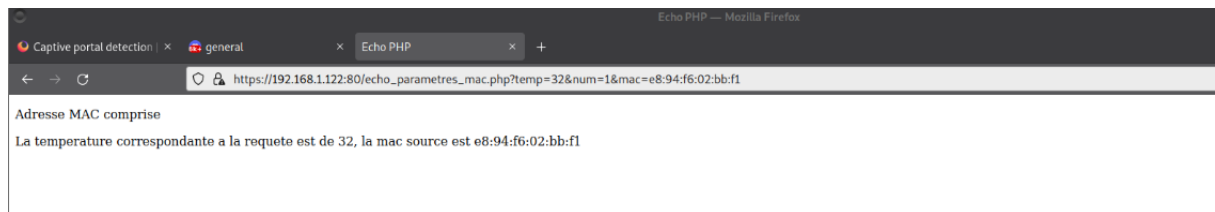


Jalon 3 :

- Copie d'écran de l'URL en https et de sa réponse :



- ⇒ On peut voir la requête :
https://192.168.1.122:80/echo_parametres_mac.php?temp=32&num=1&mac=e8:94:f6:02:bb:f1
 - ⇒ La réponse affichée utilise donc les paramètres de l'URL.
-

- Copie du fichier exemple.txt contenant plusieurs valeurs du string initial

```
tp@rt:~/Téléchargements$ cat exemple1.txt
4
5
3
9
```

- ⇒ On peut voir le contenu du fichier exemple1.txt, qui contient les valeurs du string initial.
-

- Copie du fichier exemple.txt contenant température et adresse mac

```
tp@rt:~/Téléchargements$ cat exemple2.txt
temp= 5 ,mac= 34:17:EB:9D:53:A2
temp= 5 ,mac= 34:17:EB:9D:53:A2
temp= 32 ,mac= E8:DB:84:95:DF:49
temp= 12 ,mac= E8:DB:84:95:DF:49
temp= 16 ,mac= E8:DB:84:95:DF:49
```

- ⇒ On peut voir le contenu du fichier exemple2.txt, qui contient les valeurs de température « temp= » et de l'adresse mac « mac= ».

- Copie du fichier exemple.txt contenant température, adresse mac et numéro de requête

```
tp@rt:~/Téléchargements$ cat exemple3.txt
temp= 32,num=2 ,mac= E8:DB:84:95:DF:49
temp= 32,num=3 ,mac= E8:DB:84:95:DF:49
temp= 32,num=4 ,mac= E8:DB:84:95:DF:49
temp= 32,num=5 ,mac= E8:DB:84:95:DF:49
temp= 32,num=6 ,mac= E8:DB:84:95:DF:49
temp= 32,num=7 ,mac= E8:DB:84:95:DF:49
temp= 32,num=8 ,mac= E8:DB:84:95:DF:49
temp= 32,num=9 ,mac= E8:DB:84:95:DF:49
temp= 32,num=10 ,mac= E8:DB:84:95:DF:49
temp= 32,num=11 ,mac= E8:DB:84:95:DF:49
temp= 32,num=12 ,mac= E8:DB:84:95:DF:49
temp= 32,num=13 ,mac= E8:DB:84:95:DF:49
temp= 32,num=14 ,mac= E8:DB:84:95:DF:49
temp= 32,num=15 ,mac= E8:DB:84:95:DF:49
temp= 32,num=16 ,mac= E8:DB:84:95:DF:49
temp= 32,num=17 ,mac= E8:DB:84:95:DF:49
temp= 32,num=18 ,mac= E8:DB:84:95:DF:49
temp= 32,num=19 ,mac= E8:DB:84:95:DF:49
```

- ⇒ On peut voir le contenu du fichier exemple2.txt, qui contient les valeurs de température « temp= », l'adresse mac « mac= » et du numéro de requête « num ». La valeur de « num » est sous forme de chaîne de caractère, et s'incrémente à chaque nouvelle requête grâce à notre code Arduino.
-

- Copie du programme Arduino, permettant d'envoyer l'URL à plusieurs champs

```
/**
  BasicHTTPClient.ino

  Created on: 24.05.2015

 */

#include <Arduino.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

#include <WiFiClient.h>

ESP8266WiFiMulti WiFiMulti;
String MAC_Address="";

void setup() {

  Serial.begin(115200);
  // Serial.setDebugOutput(true);

  Serial.println();
  Serial.println();
  Serial.println();

  for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
  }

  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP("binome_4", "tpRT9025");

}

void loop() {
  // wait for WiFi connection
  MAC_Address=WiFi.macAddress();
  Serial.print("ESP Board MAC Address :"+MAC_Address);
  if ((WiFiMulti.run() == WL_CONNECTED)) {

    WiFiClient client;

    HTTPClient http;

    Serial.print("[HTTP] begin...\n");
    String
site="https://192.168.1.122:80/echo_parametres_mac.php?temp=1&num=1&mac="+MAC_Address
;
  }
```

```
Serial.println(site);
if (http.begin(client,site)) { // HTTP

    Serial.print("[HTTP] GET...\n");
    // start connection and send HTTP header
    int httpCode = http.GET();

    // httpCode will be negative on error
    if (httpCode > 0) {
        // HTTP header has been send and Server response header has been handled
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);

        // file found at server
        if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
            String payload = http.getString();
            Serial.println(payload);
        }
    } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }

    http.end();
} else {
    Serial.printf("[HTTP] Unable to connect\n");
}
}

delay(10000);
}
```