#include <WiFi.h>

#include <HTTPClient.h>

#include <ArduinoJson.h>

#if CONFIG\_FREERTOS\_UNICORE

#define ARDUINO\_RUNNING\_CORE 0

#else

#define ARDUINO\_RUNNING\_CORE 1

#endif

const char\* ssid = "C1406airtel";

const char\* password = "indo1234";

// define task

void TaskAPI( void \*pvParameters );

void setup() {

Serial.begin(115200);

//////////

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");

WiFi.mode(WIFI\_STA);

while( WiFi.status()!= WL\_CONNECTED){

Serial.print(".");

delay(500);

}

Serial.println("\nConnected to WiFi Network");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

//////////

// Now set up task.

xTaskCreatePinnedToCore(

TaskAPI

, "TaskAPI" // A name just for humans

, 8000 // This stack size can be checked & adjusted by reading the Stack Highwater

, NULL

, 1 // Priority, with 3 (configMAX\_PRIORITIES - 1) being the highest, and 0 being the lowest.

, NULL

, ARDUINO\_RUNNING\_CORE);

// Now the task scheduler, which takes over control of scheduling individual tasks, is automatically started.

}

void loop()

{

// Empty. Things are done in Tasks.

}

/\*---------------------- Tasks ---------------------\*/

void TaskAPI(void \*pvParameters) // This is a task.

{

(void) pvParameters;

for (;;) // A Task shall never return or exit.

{

if((WiFi.status() == WL\_CONNECTED))

{

long rnd = random(1, 10);

HTTPClient client;

client.begin("https://fastag-internal.parkzap.com/account/mockable\_test/" + String(rnd));

int httpCode = client.GET();

String payload;

if(httpCode > 0)

{

payload = client.getString();

Serial.println(payload);

client.end();

}

else

{

Serial.println("Error in API request");

}

char json[1000];

payload.toCharArray(json, 1000);

// Serial.println(json); // data in payload is converted to char array

DynamicJsonDocument doc(8192);

serializeJson(doc, json);

JsonObject obj = doc.to<JsonObject>();

Serial.println(obj);

Serial.println(obj.size());

String key;

Serial.println("Enter the key you want to search: ");

while (Serial.available()==0)

{

while (Serial.available()==0){}

key = Serial.readString();

long startTime = millis(); //Start time

Serial.println(key);

Serial.print(" ");

Serial.print(obj.containsKey(key));

Serial.print(" - ");

const char\* value = obj[key];

Serial.print(value);

long elapsedTime = millis() - startTime; //Stop time

Serial.println(" Time for query: "); //print time

Serial.print( (long )(elapsedTime)); //print time

Serial.println(" msec "); //print time

}

WiFi.disconnect();

}

}

}