Control Your ESP8266 Projects from Anywhere Using aREST

by Marco Schwartz of aREST.io
https://arest.io/

I love to build Internet of Things projects, especially with the amazing ESP8266 WiFi chip. However, it can be a bit tricky to get started, as you might feel lost in front of all the options that are available to you when choosing a platform/framework to control your projects from the cloud.

This is why I created the aREST platform: to give everyone a free, easy to use, no registration required platform that can be used for all your Internet of Things projects. And in this guide, I will show you exactly how to use the aREST platform to control your ESP8266 projects from anywhere in the world.

Hardware & Software Requirements

You will need an ESP8266 board, like the Adafruit Feather ESP8266 board that was used for this guide. You will also need one LED, one 330 Ohm resistor, a breadboard, and some jumper wires.

This is a list of all the components that will be used in this project:

- Adafruit Feather ESP8266 WiFi board
- LED
- 330 Ohm resistor
- Breadboard
- Jumper wires

On the software side, you will need the latest version of the Arduino IDE that you can get from:

http://www.arduino.cc/en/Main/Software

Then, follow this procedure to add the ESP8266 board to the Arduino IDE:

- Start the Arduino IDE and open the Preferences window.
- Enter the following URL into the **Additional Board Manager URLs** field: http://arduino.esp8266.com/package_esp8266com_index.json

- Open Boards Manager from Tools > Board menu and install the esp8266 platform.

You will also need the PubSub library that you can get from the Arduino library manager.

Hardware Configuration

Simply start by placing the ESP8266 board on your breadboard.

For the LED, simply connect it in series with the resistor, with the longest pin of the LED connected to the resistor. Then, connect the remaining pin of the resistor to pin 5 of the ESP8266 board, and the remaining pin of the LED to the GND pin.

Finally, connect the ESP8266 board to your computer via USB.

Controlling Your ESP8266 Remotely

We are now going to connect your ESP8266 board to the aREST cloud server. This is the complete code for this part:

```
// Import required libraries
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <aREST.h>

// Clients
WiFiClient espClient;
PubSubClient client(espClient);

// Create aREST instance
aREST rest = aREST(client);

// Unique ID to identify the device for cloud.arest.io
char* device_id = "9u2co4";

// WiFi parameters
const char* ssid = "wifi-name";
const char* password = "wifi-password";

// Functions
void callback(char* topic, byte* payload, unsigned int length);

void setup(void)
{
  // Start Serial
  Serial.begin(115200);
  // Set callback
  client.setCallback(callback);

  // Give name and ID to device
  rest.set_id(device_id);
  rest.set_name("esp8266");

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

}

void loop() {

  // Connect to the cloud
  rest.loop(client);

}

// Handles message arrived on subscribed topic(s)
void callback(char* topic, byte* payload, unsigned int length) {

  rest.handle_callback(client, topic, payload, length);

}
```

This is a basic aREST sketch for the ESP8266, but you need to modify some parameters. What you need to change first is the device ID, which identifies the device on the network:

```
char* device_id = "9u2co4";
```

Then, modify the WiFi network name and password in the code:

```
const char* ssid = "wifi-name";
const char* password = "wifi-password";
```

You can now open the Arduino IDE, and grab the code, and save the sketch somewhere. Make sure to modify the WiFi name & password inside the sketch. Also give an unique ID to your board. Then, upload the sketch to the ESP8266 board. Go to your favourite web browser, and type:

```
cloud.arest.io/9u2co4/id
```

You should immediately get the answer in JSON format:

```
{
  "id": "9u2co4",
  "name": "esp8266",
  "connected": true
}
```

You can now actually completely control your board from the cloud. First, type the following command to set pin number 5 as an output:

```
cloud.arest.io/9u2co4/mode/5/o
```

After that, type the following command to put the LED on:

```
cloud.arest.io/9u2co4/digital/5/1
```

Then, you will get the answer in JSON format:

```
{
  "return_value": 1,
  "id": "9u2co4",
  "name": "esp8266",
  "connected": true
}
```

You can now control your ESP8266 boards from anywhere! As they only cost around $5, it makes it the perfect solution for your Internet of Things projects.

This is already the end of this guide about how to use the ESP8266 WiFi chip with aREST! I hope this simple project gave you an idea of what you can do with the ESP8266 along with the aREST cloud platform, in order to build exciting IoT projects.

If that's not done yet, you can of course follow my website on Facebook & on Twitter.

Thanks again, and all the best for your IoT projects using aREST!

Marco Schwartz
contact@arest.io