Project name:

ESP8226 REST demo

Goal:

Demo project for the REST library.

Setup:

This project demonstrates how to use the REST library and what is needed to use it. The context we used was to operate a relay to open a door.

Physical Setup:

* ESP8266

External libraries:

* Rapi (Rest API)
* ArduinoJson [GitHub](https://github.com/bblanchon/ArduinoJson)

Arduino / esp libraries:

* ESP8266WiFi [GitHub](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi)
* WiFiClient [Arduino](https://www.arduino.cc/en/Reference/WiFiClient)
* ESP8266WebServer [GitHub](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WebServer)
* Fs (SPIFFS File System) [GitHub](https://github.com/esp8266/Arduino/blob/master/doc/filesystem.md), [Tool download](https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.2.0/ESP8266FS-0.2.0.zip)
* ESP8266 mDNS [GitHub](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266mDNS)

Extra notes:

Project has been made on PlatformIO but is compatible with Arduino.cc after installing the [ESP8226 environment](https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/installing-the-esp8266-arduino-addon).

In the /tools folder, you will find the tool that is used [to upload files to the file system](https://github.com/esp8266/Arduino/blob/master/doc/filesystem.md#uploading-files-to-file-system) from the data folder

It uses the ActiStaff network.

How to test:

Once the sketch and SPIFFS are uploaded, we wait for the esp to connect to the network.  
Then, on a web browser, go to [www.esp8226server.local/door/1/open](http://www.esp8226server.local/door/1/open) the led should turn green.

If the mDNS responder doesn’t work you can still replace the hostname with the IP address.