Documentation for ESP8266 WiFi Configuration and the Cloud Integration

This document provides a detailed explanation of the code written for configuring WiFi credentials on an ESP8266 module and integrating it with ThingSpeak for data exchange. The code is written in Arduino using the ESP8266WiFi, ESP8266HTTPClient, ESP8266WebServer, EEPROM, and ThingSpeak libraries.

Overview
The purpose of this code is to allow dynamic configuration of WiFi credentials on an ESP8266 module through a web interface. Additionally, it integrates with ThingSpeak to read and control fields on a ThingSpeak channel.

Libraries Used
- `ESP8266WiFi`: Provides functions for connecting the ESP8266 to a WiFi network.
- `ESP8266HTTPClient`: Enables HTTP client functionality for making requests to ThingSpeak.
- `ESP8266WebServer`: Implements a simple web server to configure WiFi credentials.
- `EEPROM`: Allows reading and writing to the EEPROM memory of the ESP8266.
- `ThingSpeak`: A library for interacting with the ThingSpeak IoT platform.

Variables
- `ssid`: Default WiFi SSID.
- `passphrase`: Default WiFi password.
- `myChannelNumber`: ThingSpeak channel number.
- `myWriteAPIKey`: Write API key for ThingSpeak.
- `myCounterReadAPIKey`: Read API key for ThingSpeak.
- `ledPin1`, `ledPin2`, `ledPin3`, `ledPin4`: Pins to control LEDs.
- `LED`, `LEDValue`: Flags to control LED states.

Functions

`setup()`
- Initializes Serial communication.
- Disconnects from the current WiFi network.
- Initializes EEPROM.
- Reads saved SSID and password from EEPROM.
- Connects to WiFi or sets up a hotspot if connection fails.
- Initializes ThingSpeak connection.
- Sets up pin modes for LEDs.

`loop()`
- Checks if WiFi is connected.
- Reads values from ThingSpeak fields.
- Controls LEDs based on ThingSpeak data.

`testWifi()`
- Waits for WiFi connection and returns true if connected.
- Opens an Access Point (AP) if connection times out.

`launchWeb()`
- Prints local and SoftAP IP addresses.
- Calls `createWebServer()` to set up the web server.
- Starts the server.

`setupAP()`
- Scans available WiFi networks.
- Creates an HTML list of networks and sets up SoftAP.
- Calls `launchWeb()`.

`createWebServer()`
- Defines web server routes for updating WiFi credentials.
- Handles root route for displaying network information.
- Handles `/scan` route to go back.
- Handles `/setting` route to update WiFi credentials.

Web Interface
- Accessing the root route (`/`) displays network information, a scan button, and a form to update WiFi credentials.
- The scan route (`/scan`) displays a message to go back.
- The setting route (`/setting`) updates WiFi credentials and triggers an ESP reset.

Conclusion
This code provides a flexible solution for updating WiFi credentials on an ESP8266 module using a web interface. Additionally, it integrates with ThingSpeak to read and control fields on a ThingSpeak channel. Users can easily modify and adapt this code for their specific IoT projects.