Sending Data from ESP8266 to Telegram

Sending data from **ESP8266 to Telegram** involves connecting the ESP8266 to WiFi, reading sensor data, and using the **Telegram Bot API** to send messages via HTTPS requests. The ESP8266 formats the data into a URL and sends it using `WiFiClientSecure` and `HTTPClient`. 🚀

Steps to create an BoT in Telegram:-

- Open **Telegram App**.
- Search for **BotFather**.
- Type **/newbot** and press Enter.
- Enter a **bot name** (e.g., "ESP8266_Bot").
- Choose a **username** (must end with "bot", e.g., "ESP8266SensorBot").
- **Copy the Bot Token** (you'll need it later).

✅ **Example Token:** 123456789:ABCDEF123456789abcdef123456789abc

Steps to get an Chat ID:-

- Enter the following code in the arduino ide to get an chat ID.
- **Upload** this code to ESP8266.
- **Open Serial Monitor** (Baud Rate: 115200).
- **Send any message** to your bot on Telegram.
- **Wait a few seconds** → Your Chat ID will appear in the Serial Monitor.

Code:

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Wi-Fi credentials
const char* ssid = "SSID"; // Your Wi-Fi SSID
const char* password = "PASSWORD";   // Your Wi-Fi Password

// Telegram Bot Token
#define BOT_TOKEN "6786802086:AAG4mJ78loSqkmIDGBB1Y3UVRK1vmDpaXIE"
```

```
WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);

void setup() {
Serial.begin(115200);
WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi...");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("\nConnected to WiFi!");

client.setInsecure();  // Allow SSL connection

Serial.println("Waiting for a message from Telegram...");

// Wait for incoming messages
while (true) {
int messageCount = bot.getUpdates(bot.last_message_received + 1);
if (messageCount > 0) {
Serial.println("Message received!");

String chat_id = bot.messages[0].chat_id;
Serial.print("Your Chat ID: ");
Serial.println(chat_id);

break;
}
delay(1000);
}
}

void loop() {
}
```

Sample Output:-

```
Connecting to WiFi...

Connected to WiFi

IP Address: 192.168.1.100

Sending message...

Message Sent!

{"ok":true,"result":{"message_id":123,"chat":{"id":1925617393,...
```

Sending an Sample Message to an Telegram:-

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h> // Required for HTTPS
#include <ESP8266HTTPClient.h>

const char* ssid = "SSID";        // Your Wi-Fi SSID
const char* password = "PASSWORD"; // Your Wi-Fi Password

String botToken = "6786802086:AAG4mJ78loSqkmIDGBB1Y3UVRK1vmDpaXIE"; // Telegram Bot Token
String chatId = "1925617393"; // Your Telegram Chat ID
String message = "GOOD MORNING! Your ESP8266 is working.";

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  Serial.println("Connecting to WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("\nConnected to WiFi");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  sendMessage(message);
}

void sendMessage(String message) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClientSecure client;
    client.setInsecure();  // Skip SSL certificate validation (required for ESP8266)

    HTTPClient http;

    String url = "https://api.telegram.org/bot" + botToken + "/sendMessage?chat_id=" + chatId
+ "&text=" + message;

    Serial.println("Sending message...");
    Serial.println(url);  // Debug URL

    http.begin(client, url);  // Use HTTPS with WiFiClientSecure

    int httpCode = http.GET(); // Send request

    if (httpCode > 0) {
      Serial.println("Message Sent!");
      Serial.println(http.getString());  // Print Telegram response
    } else {
      Serial.print("Error in sending message: ");
      Serial.println(httpCode);
    }

    http.end();  // Close connection
  } else {
    Serial.println("WiFi Disconnected!");
  }
}

void loop() {
  // No need to do anything in the loop
}
```