LinkmeDaddy User Manual

[www.LinkmeDaddy.com alpha version]
Jessie Conn Ralph M. Sam, Developer

[IoT Data Made Easy: A One-Stop Platform for Data Management]

[We provide reliable, scalable API endpoints to seamlessly connect IoT devices and third-party applications to our system for smarter, integrated solutions in data management.]

Revision History

Version	Date	Revision Description
1.0	13/08/2024	Prototype Basic Features and Security.

Table of Contents

Table of Contents

1. Application Overview

Introduction

Purpose

The purpose of this startup is to build an online platform where people and companies can easily manage and use data from their **IoT** (Internet of Things) devices. This platform connects data from smart gadgets like home devices, sensors, and other machines to popular tools like Microsoft Excel, Google Sheets, TensorFlow, MATLAB etc. allowing users to see and understand their data in real-time. Beyond serving as a bridge to these tools, we also provide built-in dashboards and report generation to make data tracking even easier. Our goal is to help everyone, from beginners to large enterprises, track, organize, and make informed decisions with their IoT, web and mobile application data through clear displays.

Audience

Our startup is focused on supporting small enterprises and students working on IoT-related projects, such as thesis and capstone projects involving device data. Our online platform simplifies data management and visualization, connecting IoT data from devices and sensors to popular visualization tools for real-time insights. With built-in dashboards and reporting features, we aim to make it easy for small businesses and students creating IoT devices for their capstone or thesis projects to track, analyze, and make data-driven decisions based on the collected data. As we grow, we plan to expand to support larger enterprises and more complex IoT ecosystems.

Scope

Our platform is designed to help small businesses and students who are working on IoT projects, like thesis or capstone projects. It helps collect data from devices, sensors, and smart gadgets, and puts it all in one place. It also shows the data in simple charts and graphs, so users can quickly see what's happening with their devices. We also have tools and features that create reports, making it easier for students and small businesses to summarize their data. Our platform is easy to use, even for people who are not tech experts. As we grow, we plan to add more features to help larger businesses with even more complex projects.

Project Goals and Objectives

Primary Goals

The objectives of our platform are to make it easy for users to collect and manage data from their IoT devices and sensors. We aim to provide dashboards that allow users to track and monitor their data in real-time, making it simple to see important trends and insights. The platform will integrate smoothly with popular tools like Microsoft Excel, Google Sheets, TensorFlow, and MATLAB, enabling users to easily analyze their data in environments they are familiar with. We also want users can create reports, so users can quickly summaries their data. The platform is designed to be user-friendly, ensuring that even those with little to no technical experience can visualize it easily. As we grow, we plan to continuously enhance the platform's features to support small enterprises and students initially, and later expand to meet the needs of larger businesses with more complex requirements. Additionally, our goal is to offer clear and simple visualizations of IoT data through charts and graphs, helping users make informed decisions quickly and effectively.

Objectives

The following are the objectives of this start up:

1. Allow users to connect and gather data from various IoT devices and sensors.
2. Enable users to set up dashboards that show data for quick insights.
3. Make it easy for users to link their IoT data with familiar tools like Microsoft Excel, Google Sheets, TensorFlow, MATLAB etc. for deeper analysis.
4. Offer features to generate data summaries and reports.
5. Design the platform with an easy-to-use interface so that even beginners or non-technical users can navigate it smoothly.
6. Start by focusing on small businesses and students, then grow to support larger organizations and more complex needs over time.
7. Present IoT data in straightforward charts and graphs to help users quickly understand and act on their insights.

Core Features

Feature Breakdown

The following are the free features of the LinkmeDaddy application:

1. Provide secure API endpoint with fetching and data limits.
2. Easy to set up asynchronous dashboard.
3. Limited report generation.

4. Limited CSV data loads.
5. Limited report publishing.
6. Act as bridge between popular visualization tools (MATLAB, Excel, TensorFlow etc.) and IoT devices.
7. Insights and overview generation.

The following are the premium features of the LinkmeDaddy application:

1. Provide secure API endpoint with higher fetching and data limits.
2. Easy to set up a real time dashboard.
3. Unlimited report generations.
4. Unlimited CSV data loads but not exceeding data limits.
5. Higher limit in report publishing.
6. Act as bridge between popular visualization tools (MATLAB, Excel, TensorFlow etc.) and IoT devices.
8. Can customize Insights and overview generation.

Future Improvements

Planned Features and Enhancements

We aim to add a marketplace where users can buy and sell IoT devices and components. We will integrate machine learning and AI to generate reports and enable data visualization. We are also planning to include a control room feature, allowing users to send requests and commands to their IoT devices, enabling in-system control. Essentially, we are taking a step in the right direction to attract larger enterprises.

Getting Started

Registration

Login

Setting Up Dashboard

API Endpoint Information and Security Settings

Control Room Usage and Settings

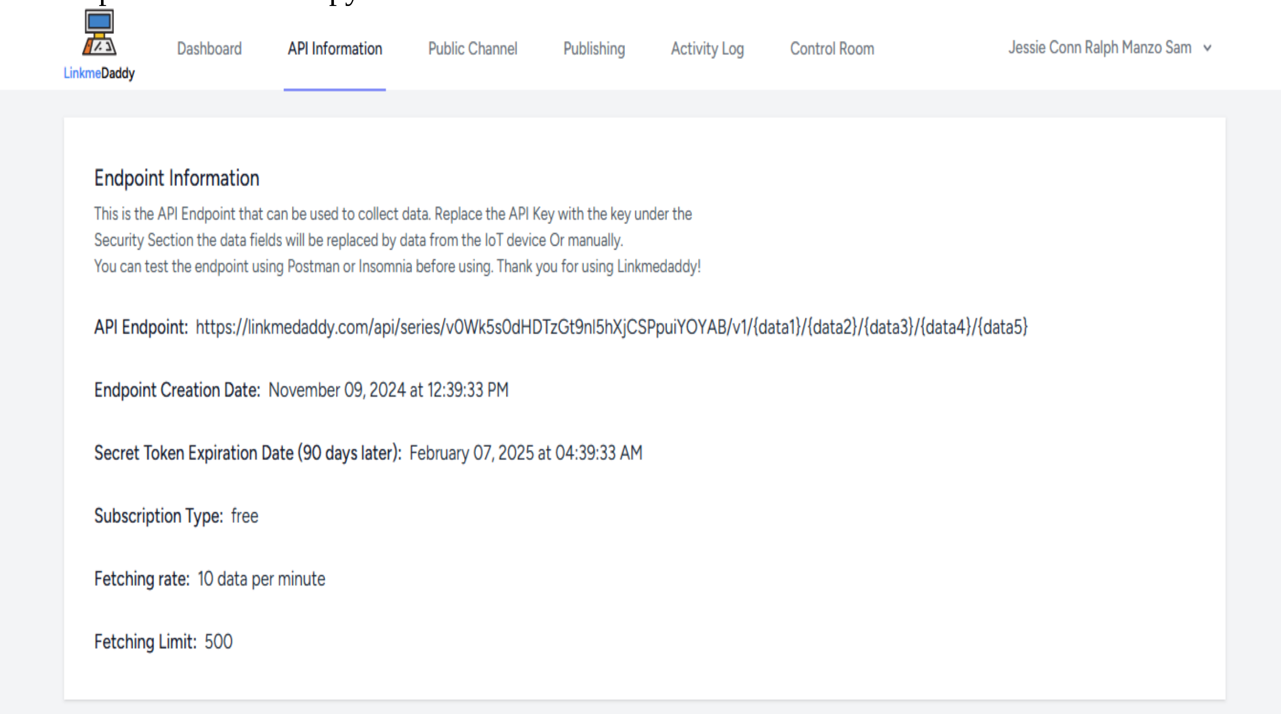
Getting Started with Publishing Report

Setting up public Channels

Making It Real: Practical Applications and Integrations

Testing Endpoint with Insomnia

After registering with LinkmeDaddy, go to "API Information." Click the link in the Endpoint URL and copy it to the URL section in Insomnia.



Endpoint Information

This is the API Endpoint that can be used to collect data. Replace the API Key with the key under the Security Section the data fields will be replaced by data from the IoT device Or manually.
You can test the endpoint using Postman or Insomnia before using. Thank you for using Linkmedaddy!

API Endpoint: https://linkmedaddy.com/api/series/v0Wk5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/{data1}/{data2}/{data3}/{data4}/{data5}

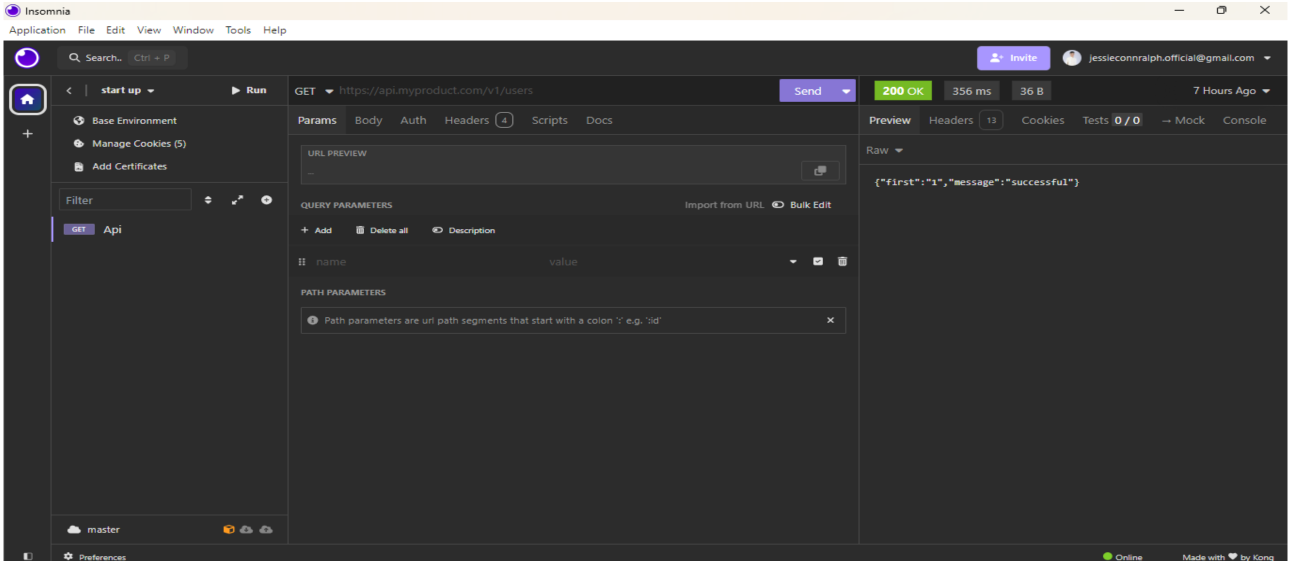
Endpoint Creation Date: November 09, 2024 at 12:39:33 PM

Secret Token Expiration Date (90 days later): February 07, 2025 at 04:39:33 AM

Subscription Type: free

Fetching rate: 10 data per minute

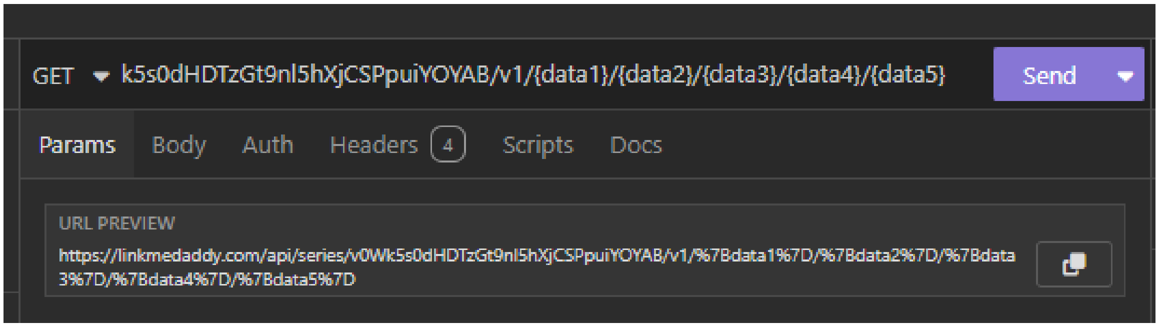
Fetching Limit: 500


Insomnia
Application File Edit View Window Tools Help
Search.. Ctrl + P
Invite
jessieconnralph.official@gmail.com
start up
Run
Base Environment
Manage Cookies (5)
Add Certificates
Filter
GET Api
master
Preferences
Params Body Auth Headers 4 Scripts Docs
URL PREVIEW
QUERY PARAMETERS
Import from URL Bulk Edit
Add Delete all Description
name value
PATH PARAMETERS
Path parameters are url path segments that start with a colon ':' e.g. ':id'
Send
Preview Headers 13 Cookies Tests 0 / 0 → Mock Console
Raw
7 Hours Ago
Online
Made with ♥ by Kong

GET https://api.myproduct.com/v1/users

200 OK 356 ms 36 B

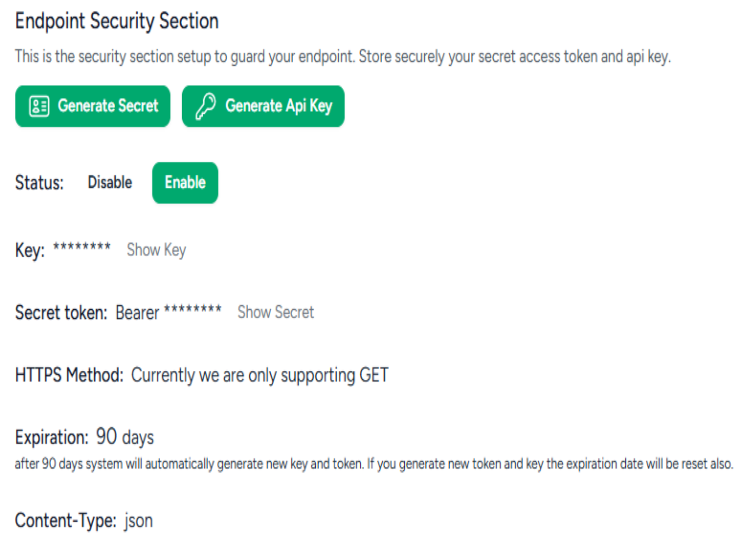
```
{"first":"1","message":"successful"}
```



GET k5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/{data1}/{data2}/{data3}/{data4}/{data5}

URL PREVIEW
https://linkmedaddy.com/api/series/v0Wk5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/%7Bdata1%7D/%7Bdata2%7D/%7Bdata3%7D/%7Bdata4%7D/%7Bdata5%7D

After adding the URL endpoint, go to the "Endpoint Security" section and copy the secret token. Then, add it to the Insomnia header as "Authorization."



Endpoint Security Section

This is the security section setup to guard your endpoint. Store securely your secret access token and api key.

Status: Disable Enable

Key: ******** Show Key

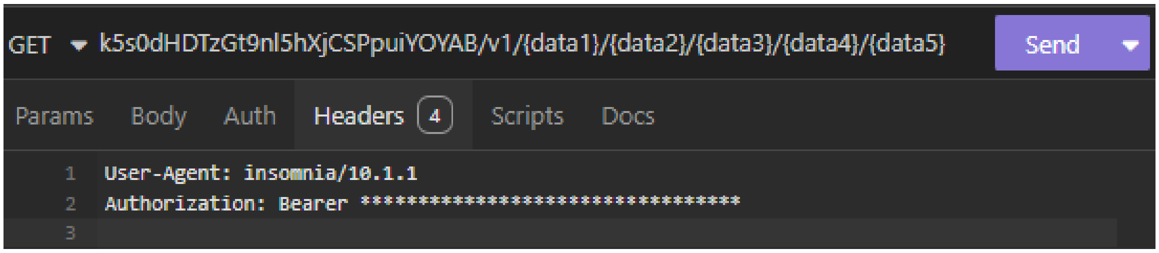
Secret token: Bearer ******** Show Secret

HTTPS Method: Currently we are only supporting GET

Expiration: 90 days

after 90 days system will automatically generate new key and token. If you generate new token and key the expiration date will be reset also.

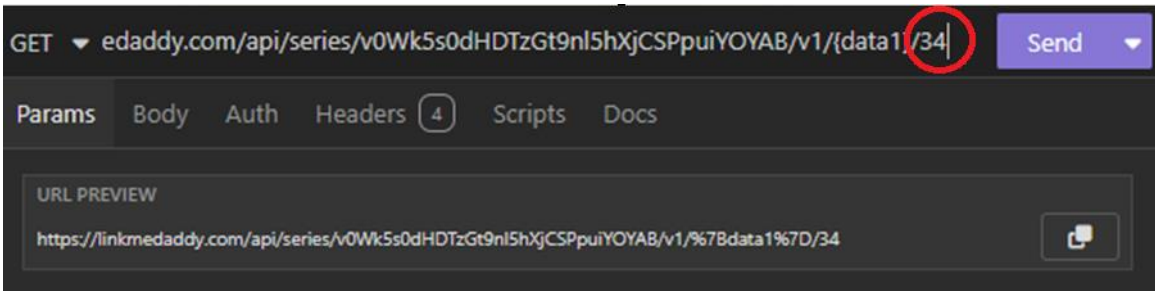
Content-Type: json



```
GET k5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/{data1}/{data2}/{data3}/{data4}/{data5}
```

```
User-Agent: insomnia/10.1.1
Authorization: Bearer **********************************
```

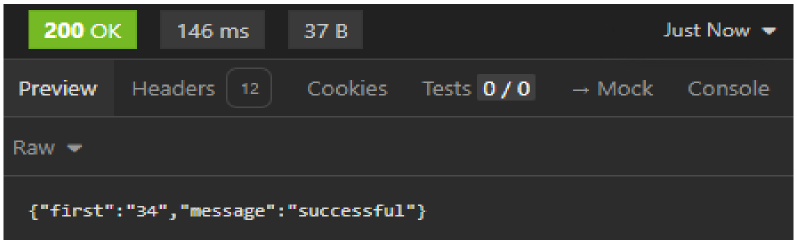
After setting it up, to simulate data fetching, replace the data in the URL with a sample. In a real scenario, this would be populated with actual sensor data.



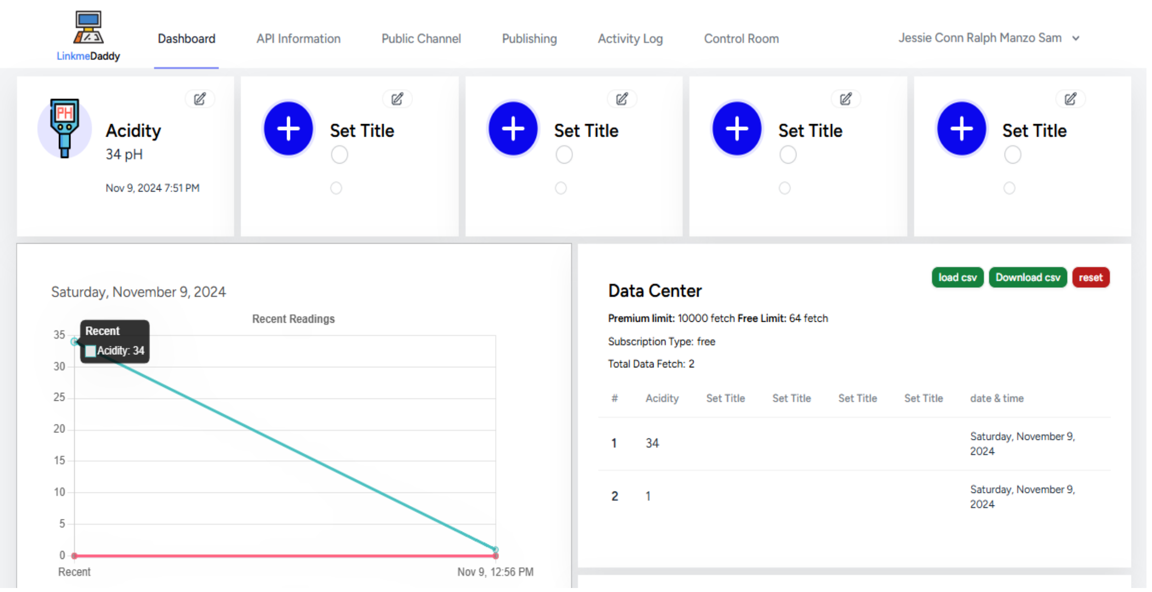
```
GET edaddy.com/api/series/v0Wk5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/{data1}/34
```

```
https://linkmedaddy.com/api/series/v0Wk5s0dHDTzGt9nl5hXjCSPpuiYOYAB/v1/%7Bdata1%7D/34
```

After clicking the "Send" button, the data will be sent to the dashboard in real time.



```
{"first":"34","message":"successful"}
```



Data Center

Premium limit: 10000 fetch **Free Limit:** 64 fetch

Subscription Type: free

Total Data Fetch: 2

#	Acidity	Set Title	Set Title	Set Title	Set Title	date & time
1	34					Saturday, November 9, 2024
2	1					Saturday, November 9, 2024

Send Data using Arduino with GSM Module to LinkmeDaddy

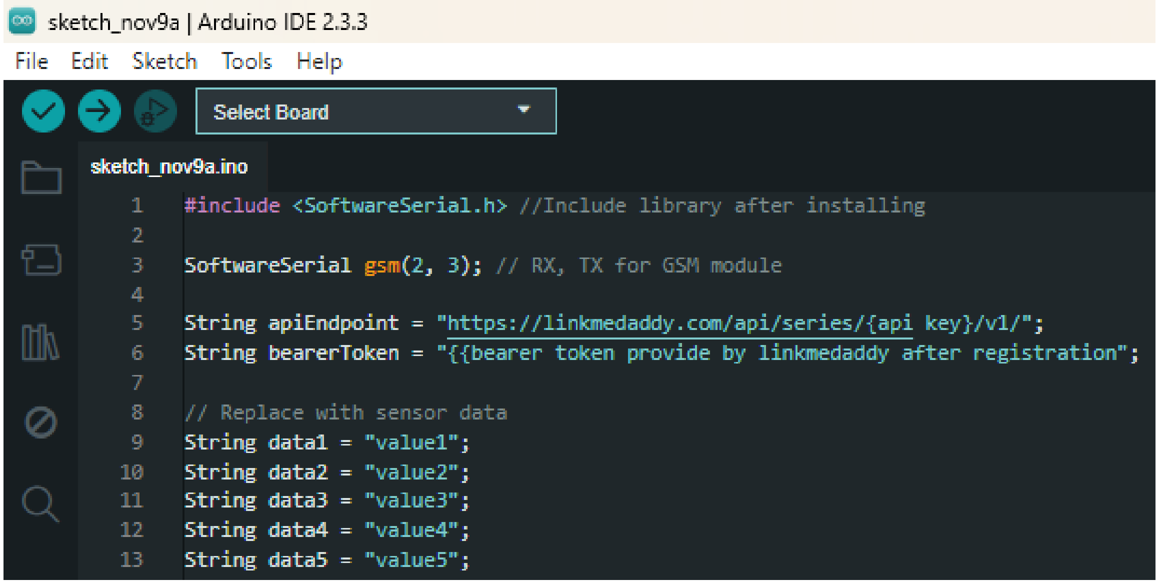
Materials:

1. Arduino Board (e.g., Uno, Mega, etc.)
2. GSM Module (like SIM800L or SIM900)
3. SIM Card with active data plan for the GSM module
4. Arduino IDE installed on your computer

Set Up Your Arduino and GSM Module

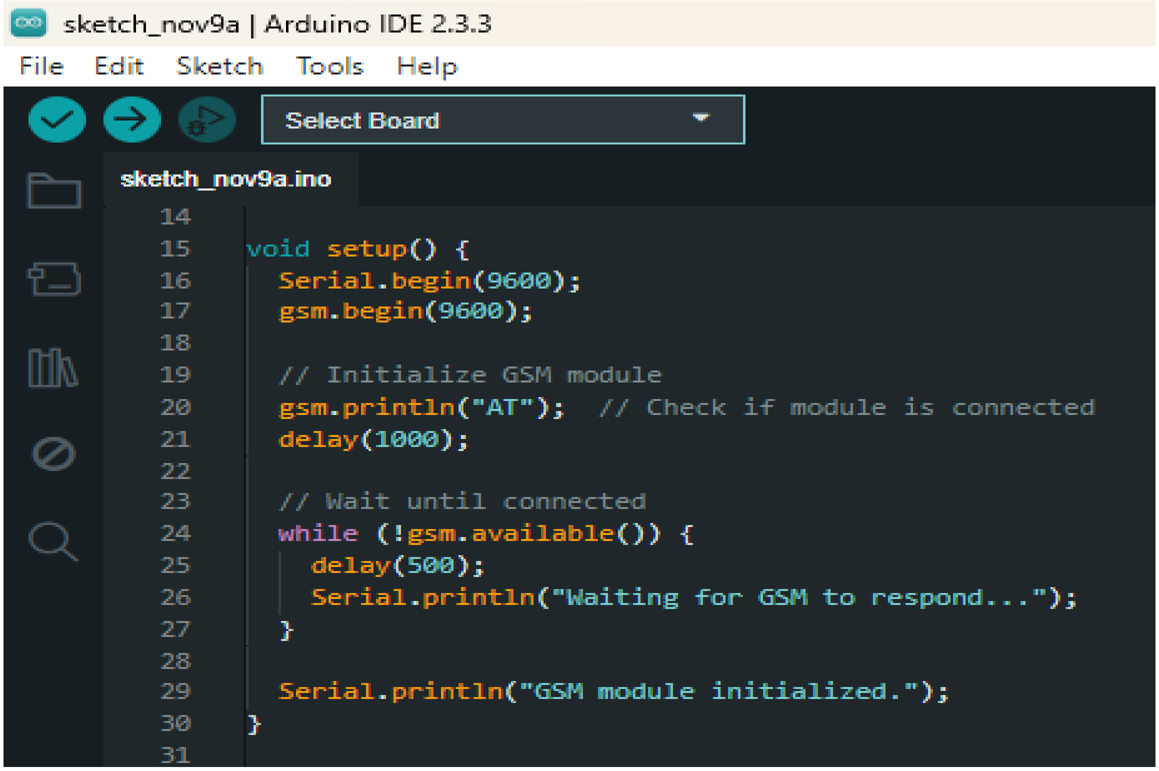
1. Connect your GSM module to the Arduino:
    - SIM800/SIM900 to Arduino Connections:
        - VCC to 5V (or 3.3V depending on your GSM module's voltage requirement).
        - GND to GND.
        - TX of GSM module to RX on Arduino (usually pin 2 or 3 if using SoftwareSerial avoid using pins that Arduino used for serial communication).
        - RX of GSM module to TX on Arduino (usually pin 2 or 3 if using SoftwareSerial).

Arduino Code



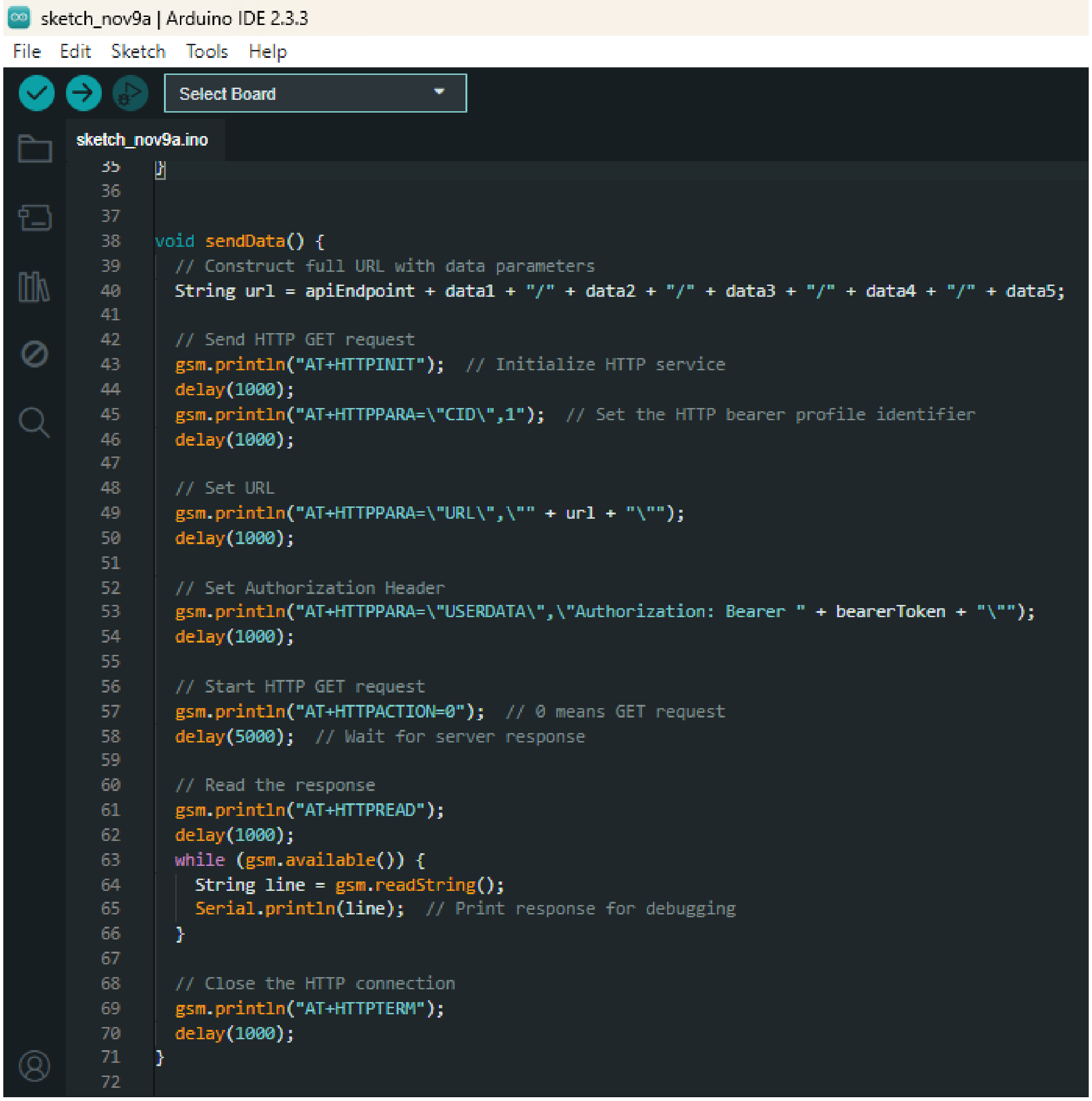
```
#include <SoftwareSerial.h> //Include library after installing

SoftwareSerial gsm(2, 3); // RX, TX for GSM module

String apiEndpoint = "https://linkmedaddy.com/api/series/{api key}/v1/";
String bearerToken = "{{bearer token provide by linkmedaddy after registration";

// Replace with sensor data
String data1 = "value1";
String data2 = "value2";
String data3 = "value3";
String data4 = "value4";
String data5 = "value5";
```

*This code sets up a serial connection with a GSM module on pins 2 and 3 using **SoftwareSerial**. It also defines the API endpoint URL (apiEndpoint) and the authorization token (bearerToken) needed to send data securely to the server.*



```

void setup() {
  Serial.begin(9600);
  gsm.begin(9600);

  // Initialize GSM module
  gsm.println("AT");  // Check if module is connected
  delay(1000);

  // Wait until connected
  while (!gsm.available()) {
    delay(500);
    Serial.println("Waiting for GSM to respond...");
  }

  Serial.println("GSM module initialized.");
}

```

In the setup () function, the code starts serial communication with both the computer and the GSM module at a baud rate of 9600. It sends an "AT" command to the GSM module to check if it's connected, then waits for a response by checking if data is available from the module. If no response is received, it prints "Waiting for GSM to respond..." every half second until the module connects, then confirms with "GSM module initialized."



```
}


void sendData() {
  // Construct full URL with data parameters
  String url = apiEndpoint + data1 + "/" + data2 + "/" + data3 + "/" + data4 + "/" + data5;

  // Send HTTP GET request
  gsm.println("AT+HTTPINIT");  // Initialize HTTP service
  delay(1000);
  gsm.println("AT+HTTPPARA=\"CID\",1");  // Set the HTTP bearer profile identifier
  delay(1000);

  // Set URL
  gsm.println("AT+HTTPPARA=\"URL\",\"" + url + "\"");
  delay(1000);

  // Set Authorization Header
  gsm.println("AT+HTTPPARA=\"USERDATA\",\"Authorization: Bearer " + bearerToken + "\"");
  delay(1000);

  // Start HTTP GET request
  gsm.println("AT+HTTPACTION=0");  // 0 means GET request
  delay(5000);  // Wait for server response

  // Read the response
  gsm.println("AT+HTTPREAD");
  delay(1000);
  while (gsm.available()) {
    String line = gsm.readString();
    Serial.println(line);  // Print response for debugging
  }

  // Close the HTTP connection
  gsm.println("AT+HTTPTERM");
  delay(1000);
}

```

The code is designed to send data from an Arduino with a GSM module to an online API. First, it sets up a connection between the Arduino and GSM module so they can communicate. The code initializes the GSM module and checks to ensure its properly connected. Then, it creates the complete URL by combining a given API endpoint with specific data values that the Arduino will send. Before sending the data, it sets up an authorization header using a token to confirm it has permission to access the server. After setting everything up, the code sends a request to the server (a GET request) to send the data. It waits briefly to get the server's response, then reads and displays that response in the Serial Monitor, which helps in checking if everything worked. Lastly, it closes the HTTP connection to keep everything neat and ready for the next time it sends data. The code repeats this process every five minutes, which you can adjust if needed.

Troubleshooting Tips

To troubleshoot, first make sure the GSM module has an internet connection by checking its network signal. Double-check that the API endpoint URL and token are correct. You can also test if the server is reachable by trying the same URL and authorization header in tools like Postman or Insomnia to see if it works.

Send Data using Arduino with ESP Module to LinkmeDaddy Application

Send Data using Arduino with Ethernet Shield Module to LinkmeDaddy Application

Send Data Using Raspberry pie to LinkmeDaddy

Connect Excel to LinkmeDaddy

Interface LinkmeDaddy with Google Sheets for dynamic data tracking

Connect Arduino with Ultrasonic Sensor to send distance measurements to LinkmeDaddy

Integrate LinkmeDaddy with Node-RED for streamlined IoT data management

Link multiple Raspberry Pi sensors with LinkmeDaddy for networked sensor monitoring

Sync data between LinkmeDaddy and Power BI for enhanced visual data insights

Establish connection between Arduino and LinkmeDaddy via LoRa for long-distance data communication

Connect Firebase to LinkmeDaddy for mobile app data syncing

Integrate LinkmeDaddy with MATLAB for advanced data processing and analysis

Connect LinkmeDaddy with Microsoft Power Automate for automated workflows

Connect LinkmeDaddy to tableau to extend visualization

LinkmeDaddy as Bridge between IoT Devices and TensorFlow to perform machine learning to the Data Collected