Information Technology for Logistics

Part 2: N8N

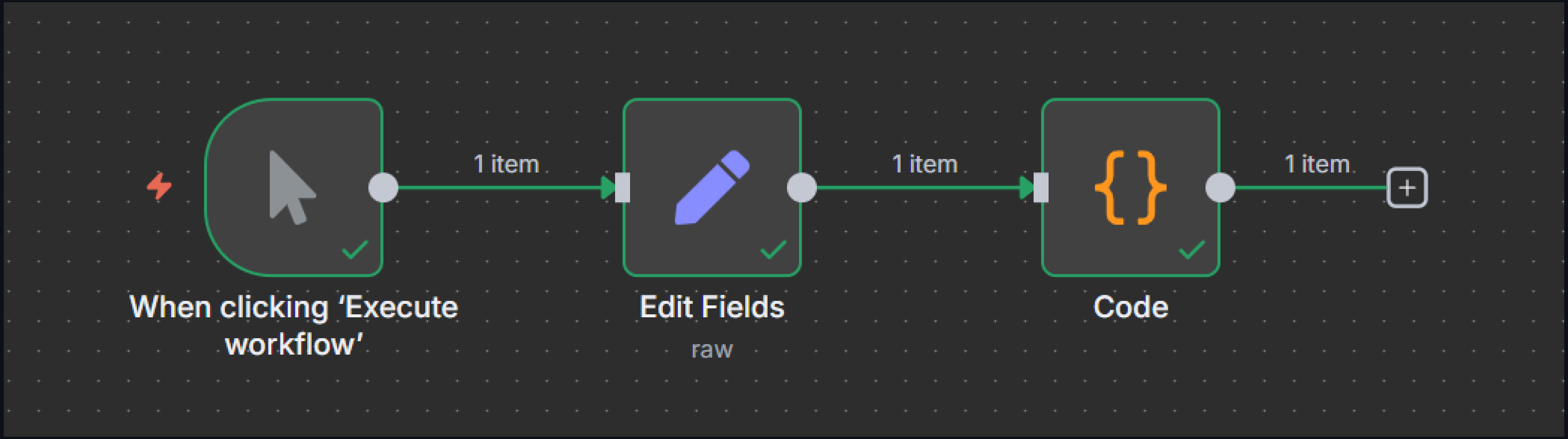
What is it?

> `n8n` is an open-source workflow automation tool that allows users to automate tasks and connect different apps, services, and APIs using a simple, visual interface.

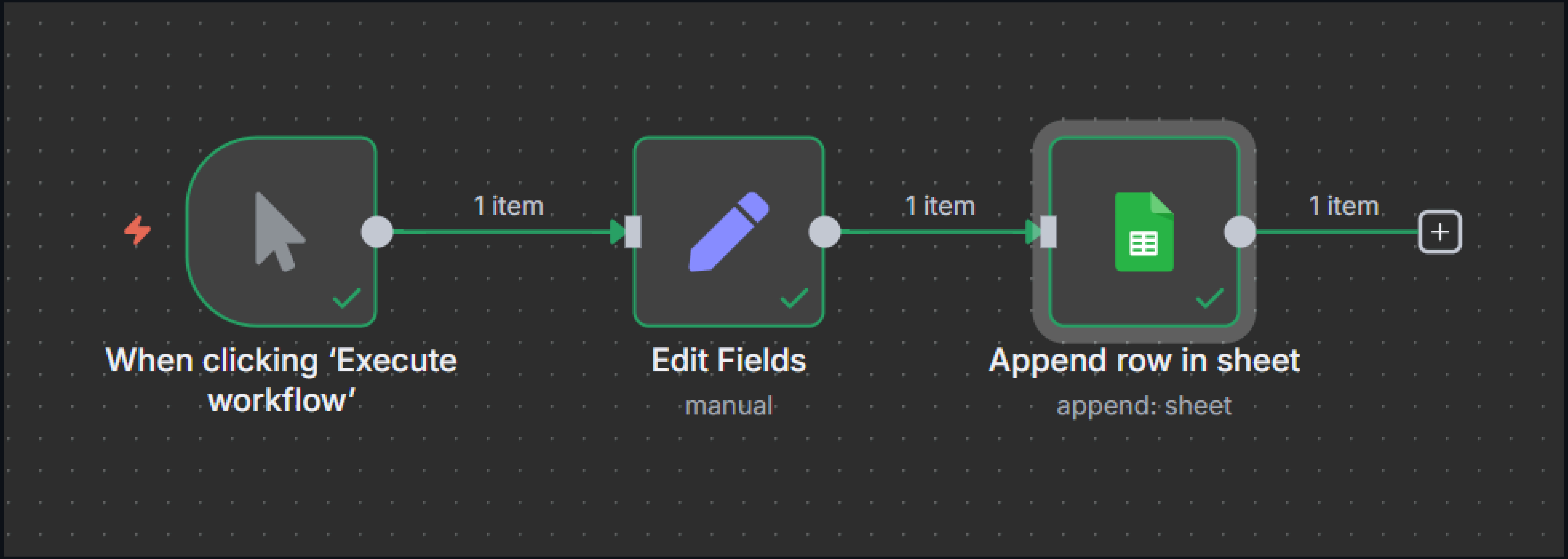
Installation

Guide

Basic Flow 1


1 item
1 item
1 item
When clicking 'Execute workflow'
Edit Fields
raw
Code

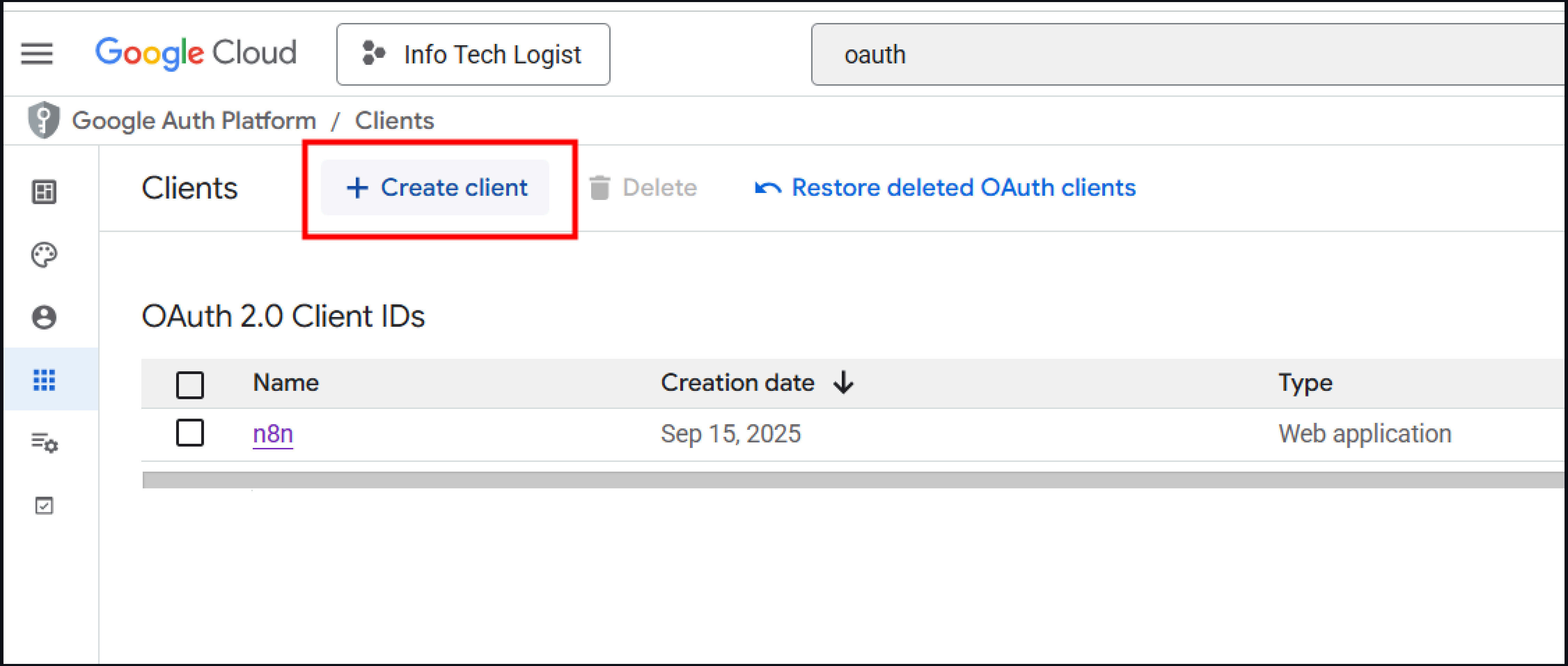
Google Sheet



Google Sheet

- Create a new `project` in Google Cloud Platform
- Enable `Google Sheet API`
- Create `App` from the `Consent Screen`
- Create a `client` from the `App`
- Add a test `user`
- Enable `Google Drive API`

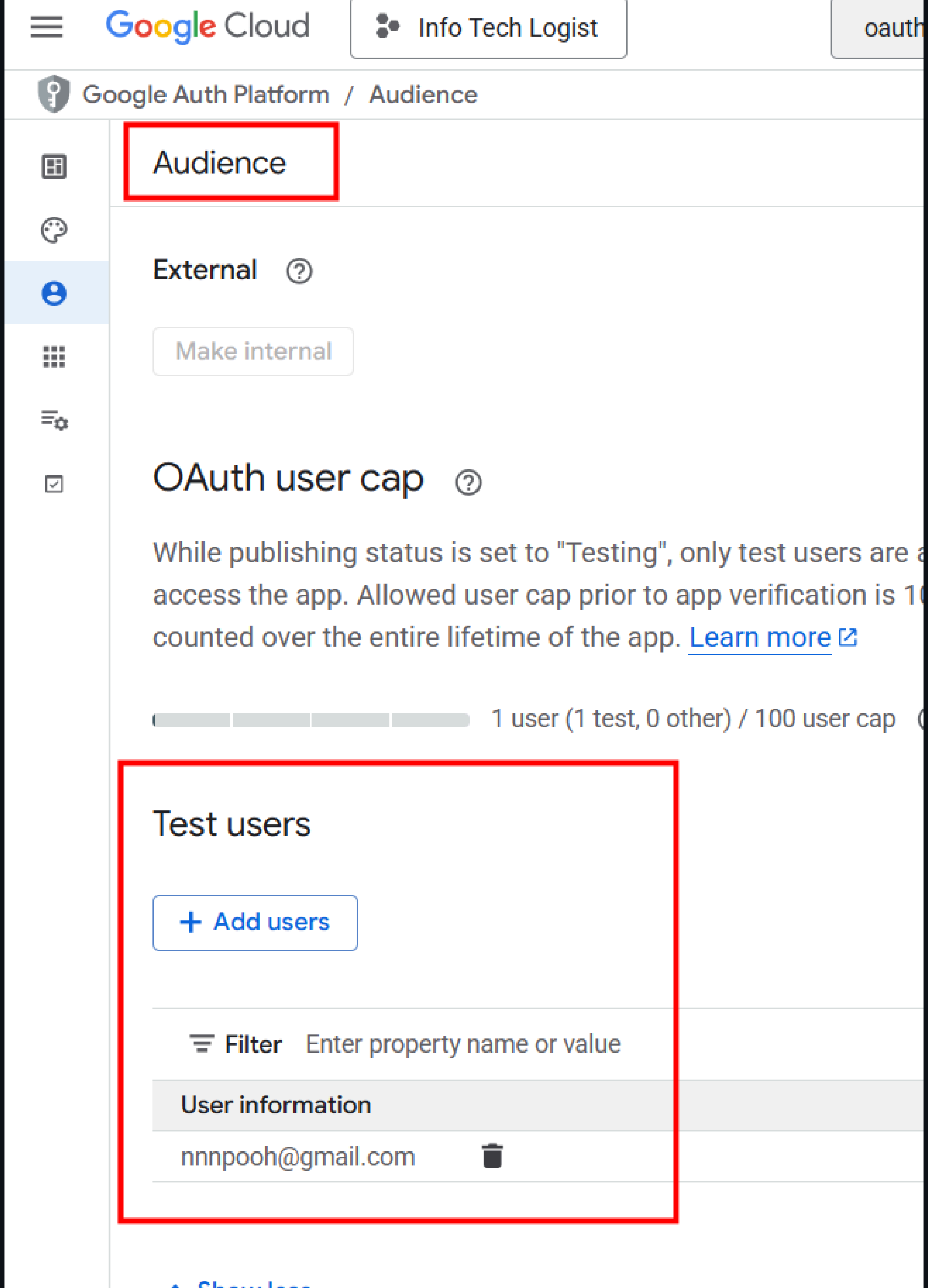
Create Client



OAuth 2.0 Client IDs

Name	Creation date ↓	Type
n8n	Sep 15, 2025	Web application

Add Test User



Audience

External

Make internal

OAuth user cap

While publishing status is set to "Testing", only test users are a
access the app. Allowed user cap prior to app verification is 1
counted over the entire lifetime of the app. Learn more

1 user (1 test, 0 other) / 100 user cap

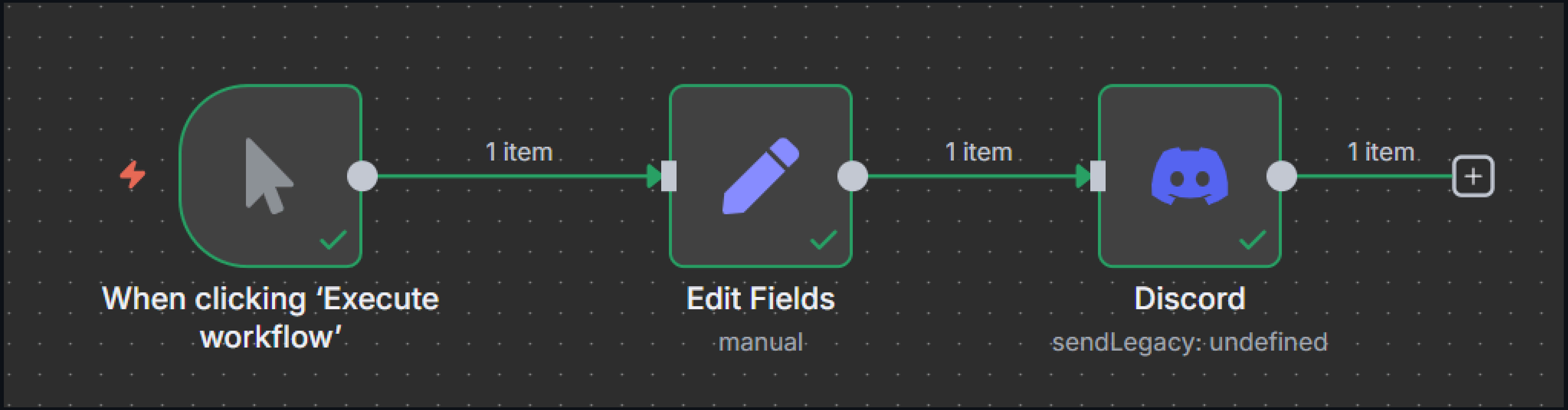
Test users

+ Add users

Filter Enter property name or value

User information
nnnpooh@gmail.com

Notification

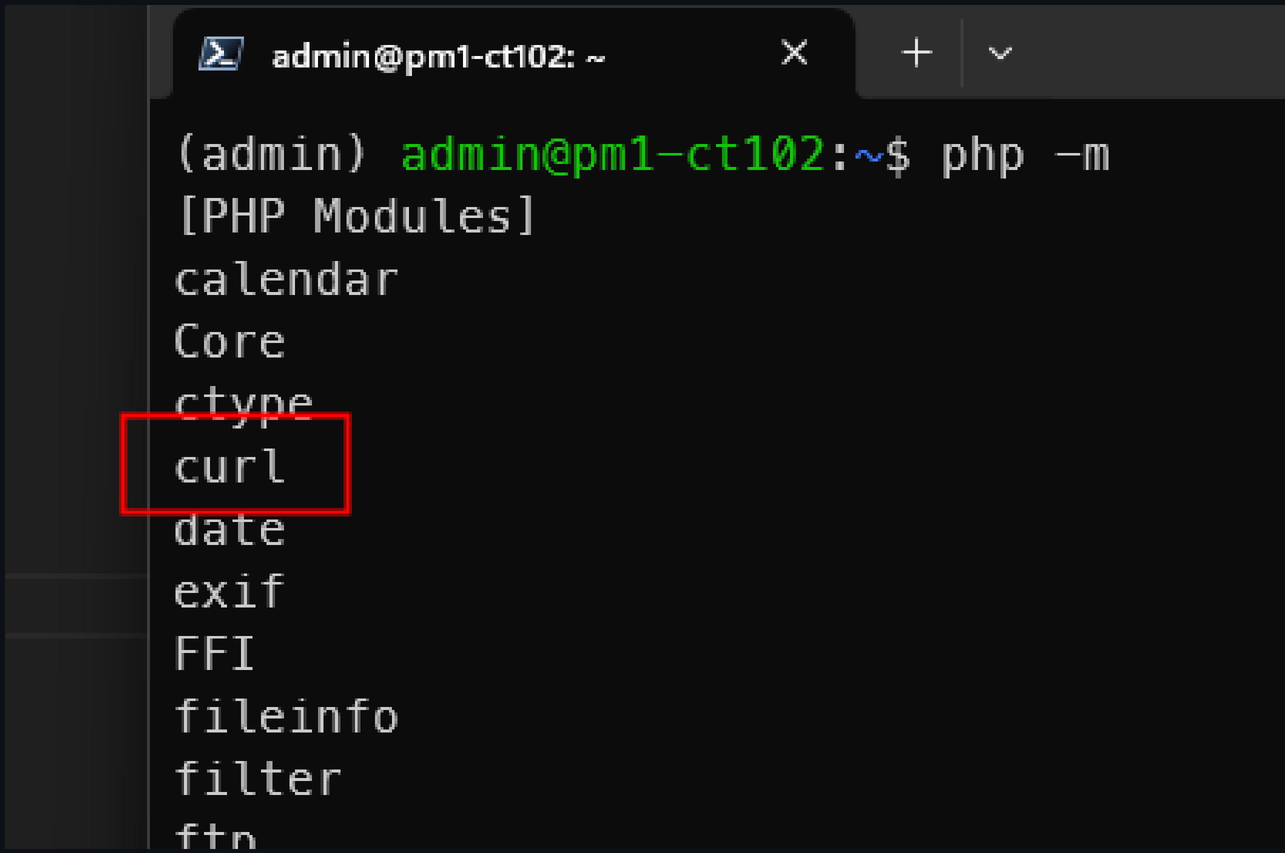


Drink Ordering System

Connecting Web Application to Other Services

Install `curl` in `php`

- `sudo apt-get install php-curl`
- `sudo systemctl restart nginx`
- `php -m`



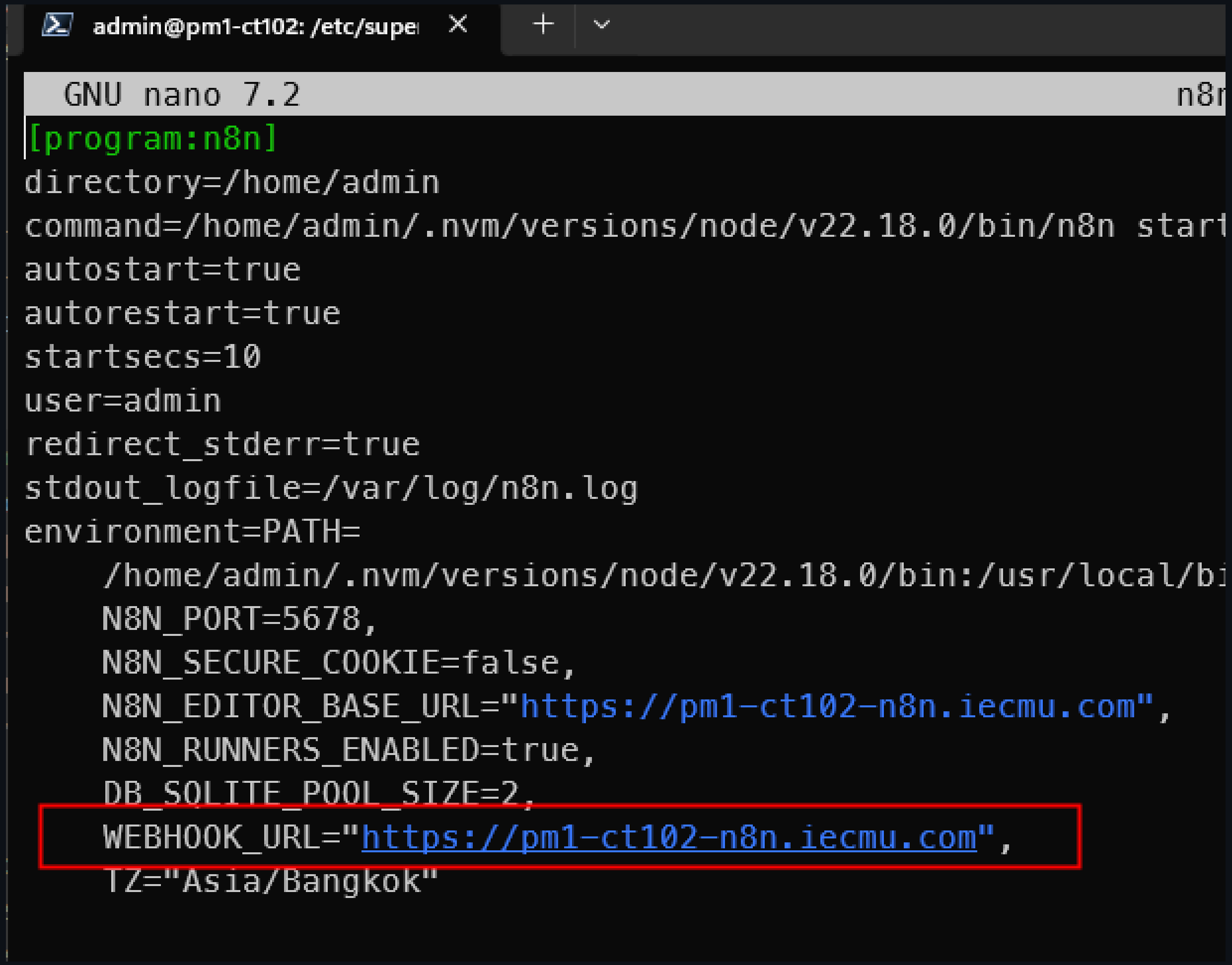
```
(admin) admin@pm1-ct102:~$ php -m
[PHP Modules]
calendar
Core
ctype
curl
date
exif
FFI
fileinfo
filter
ftp
```

Modify `n8n` config

- `sudo nano /etc/supervisor/conf.d/n8n.conf`
- Add this line *(see next page)*

```
WEBHOOK_URL="https://pmX-ctXXX-n8n.iecmu.com",
```

- `sudo systemctl restart supervisor`



```
  GNU nano 7.2                                                 n8
[program:n8n]
directory=/home/admin
command=/home/admin/.nvm/versions/node/v22.18.0/bin/n8n star
autostart=true
autorestart=true
startsecs=10
user=admin
redirect_stderr=true
stdout_logfile=/var/log/n8n.log
environment=PATH=
    /home/admin/.nvm/versions/node/v22.18.0/bin:/usr/local/bi
    N8N_PORT=5678,
    N8N_SECURE_COOKIE=false,
    N8N_EDITOR_BASE_URL="https://pm1-ct102-n8n.iecmu.com",
    N8N_RUNNERS_ENABLED=true,
    DB_SQLITE_POOL_SIZE=2,
    WEBHOOK_URL="https://pm1-ct102-n8n.iecmu.com",
    TZ="Asia/Bangkok"
```

Database

- Create a new database

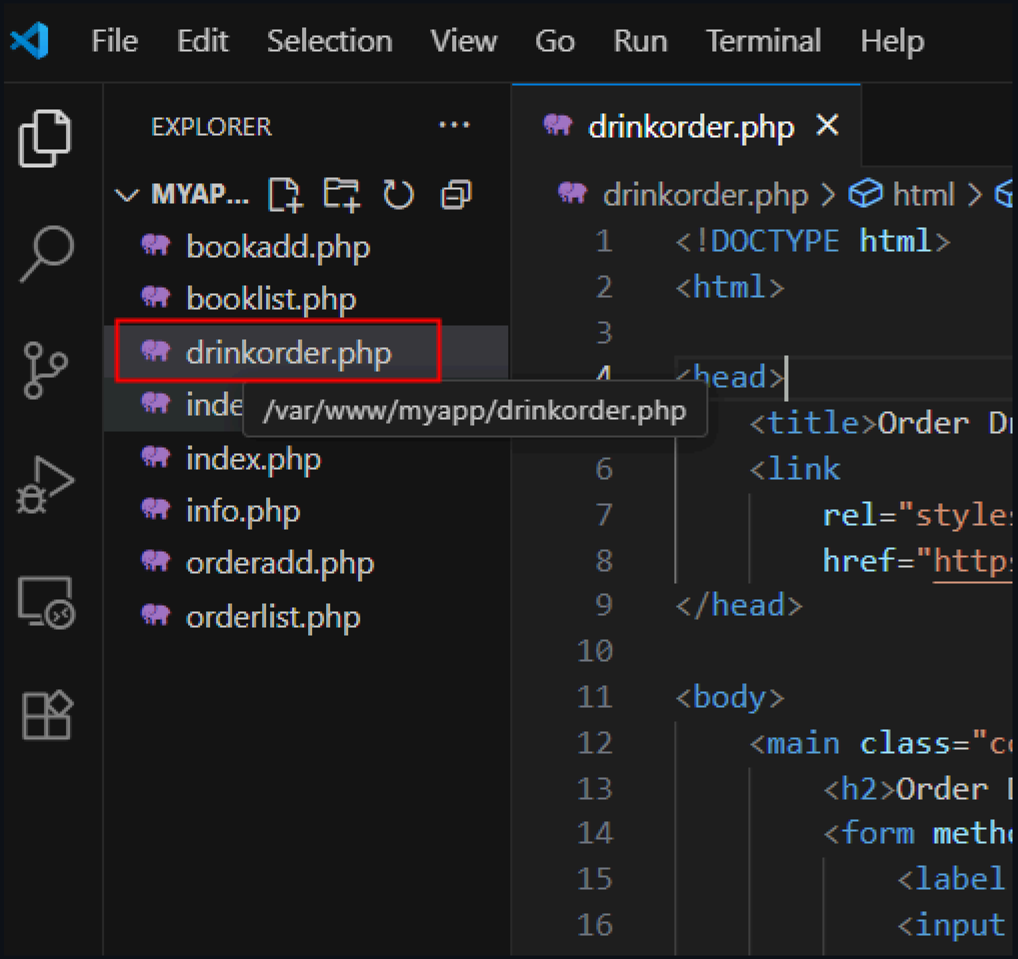
```sql
CREATE DATABASE IF NOT EXISTS iedrink;
```

- Create a new table

```sql
CREATE TABLE IF NOT EXISTS iedrink.orders (
  id INT NOT NULL AUTO_INCREMENT,
  drink_name VARCHAR(100),
  customer_name VARCHAR(100),
  quantity INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id)
);
```

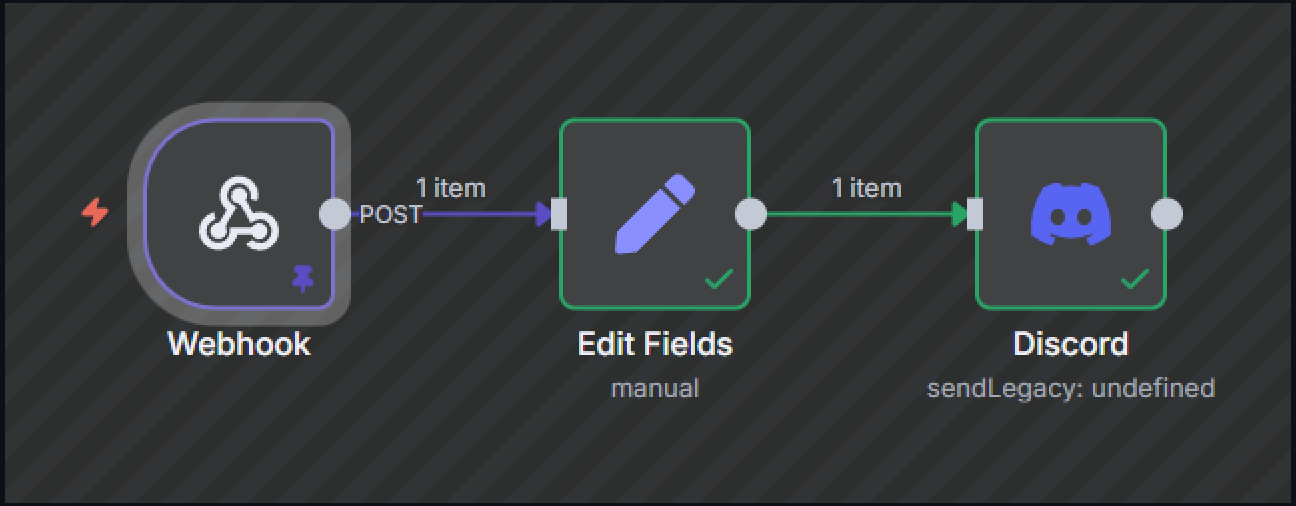
Web Application

- Create `drinkorder.php` (Link)
  - Modify your database credential.
- Visit your page at https://pmX-ctXXX-web.iecmu.com/drinkorder.php

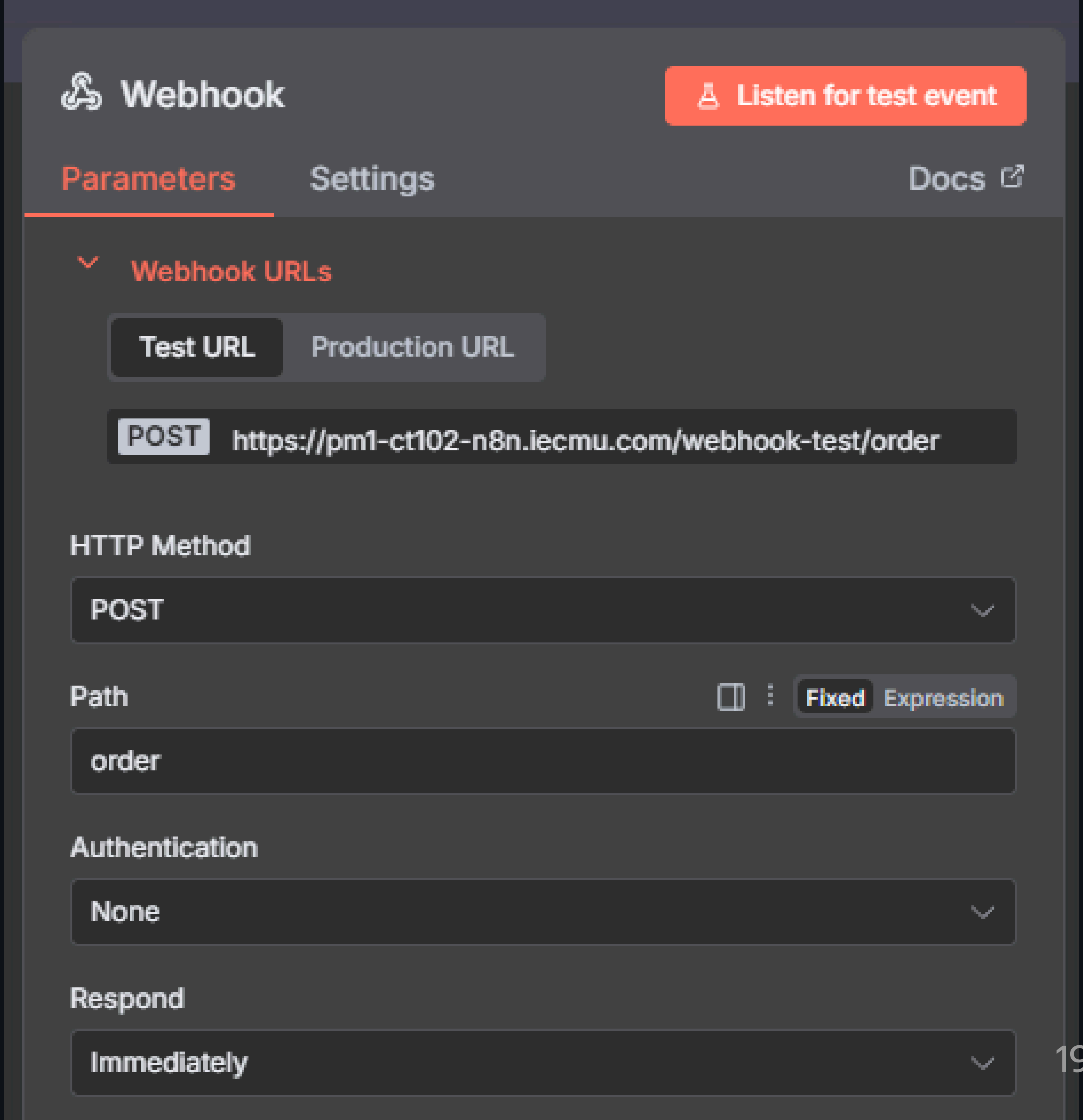


```
<!DOCTYPE html>
<html>

<head>
    <title>Order Dr
    <link
        rel="style
        href="http
</head>

<body>
    <main class="c
        <h2>Order 
        <form meth
            <label
            <input
```

Webhook Flow



Webhook



POST https://pm1-ct102-n8n.iecmu.com/webhook-test/order

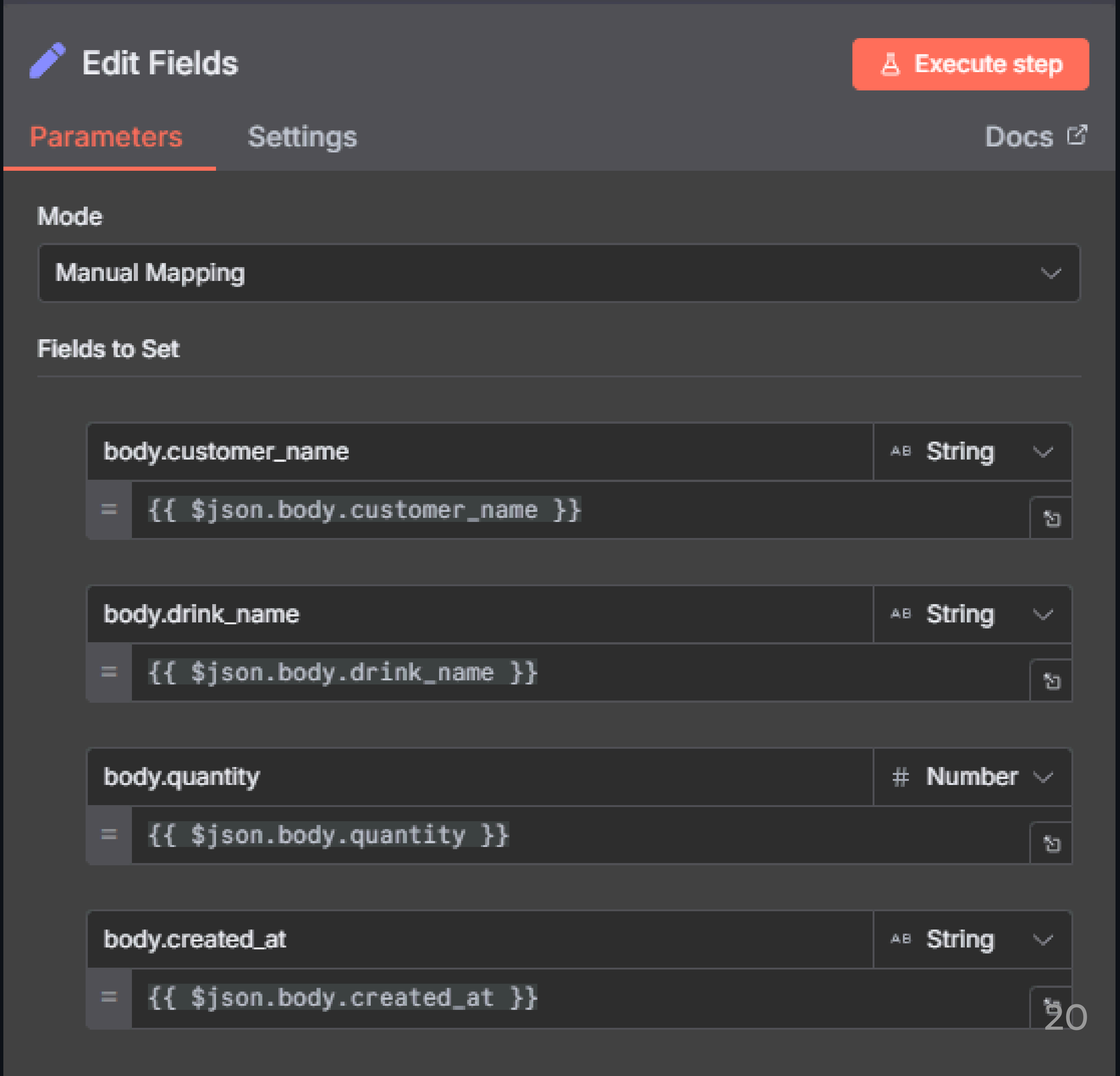
HTTP Method: POST

Path (Fixed / Expression): order

Authentication: None

Respond: Immediately

Edit Fields



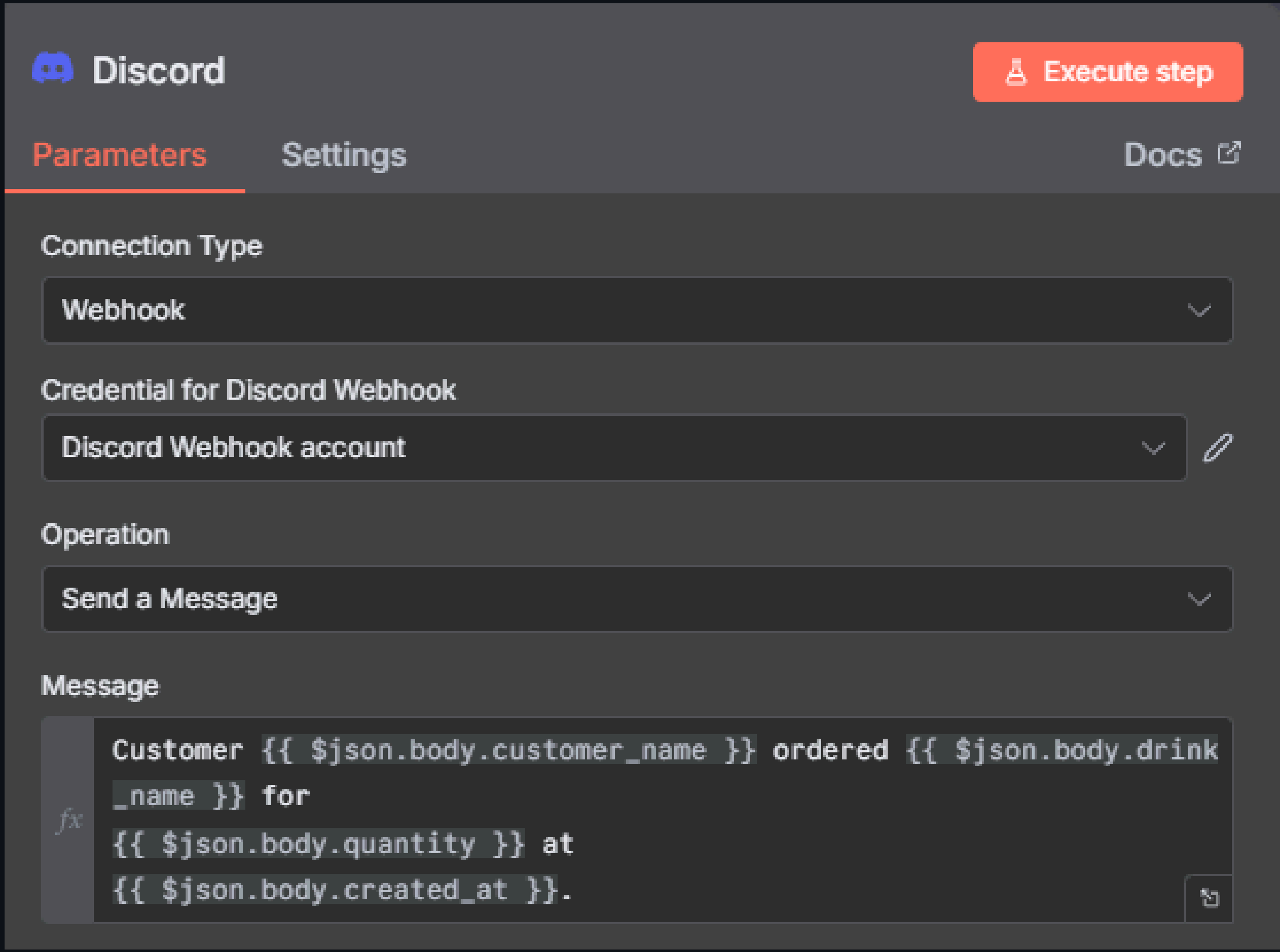
Mode

Manual Mapping

Fields to Set

body.customer_name	String
`{{ $json.body.customer_name }}`	
body.drink_name	String
`{{ $json.body.drink_name }}`	
body.quantity	Number
`{{ $json.body.quantity }}`	
body.created_at	String
`{{ $json.body.created_at }}`	

Discord



Connection Type
Webhook

Credential for Discord Webhook
Discord Webhook account

Operation
Send a Message

Message
```
Customer {{ $json.body.customer_name }} ordered {{ $json.body.drink_name }} for
{{ $json.body.quantity }} at
{{ $json.body.created_at }}.
```

Discord

```
Customer {{ $json.body.customer_name }} ordered {{ $json.body.drink_name }} for
{{ $json.body.quantity }} at
{{ $json.body.created_at }}.
```

Extra

> Can you write the data into Google Sheet too?

MQTT

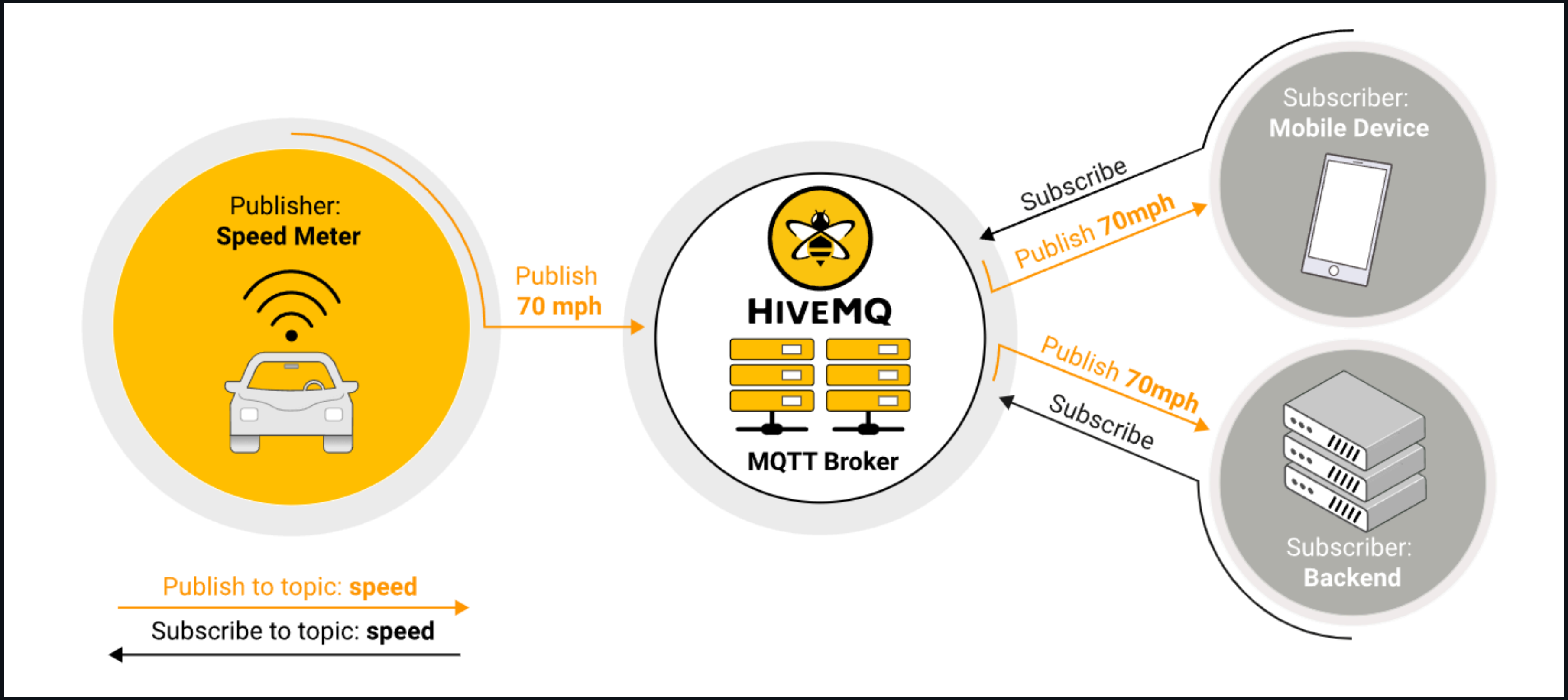
- MQTT is a Client Server publish/subscribe messaging transport protocol.
- It is light weight, open, simple, and designed so as to be easy to implement.
- Ideal for use in many situations
  - Machine to Machine (M2M)
  - **Internet of Things (IoT)**

Publish/subscribe pattern

- The publish/subscribe pattern (also known as `pub/sub`) provides an **alternative** to traditional client-server architecture.
    - In client-server architecture, a client communicates directly with an endpoint.

Broker

- The connection between `publishers` and `subscibers` is handled by a third component (the `broker`).
- The job of the broker is to filter all incoming messages and distribute them correctly to subscribers.


Publisher:
Speed Meter
Publish
70 mph
HIVEMQ
MQTT Broker
Subscribe
Publish 70mph
Subscriber:
Mobile Device
Publish 70mph
Subscribe
Subscriber:
Backend
Publish to topic: speed
Subscribe to topic: speed

Sensor

Storing Sensor Data

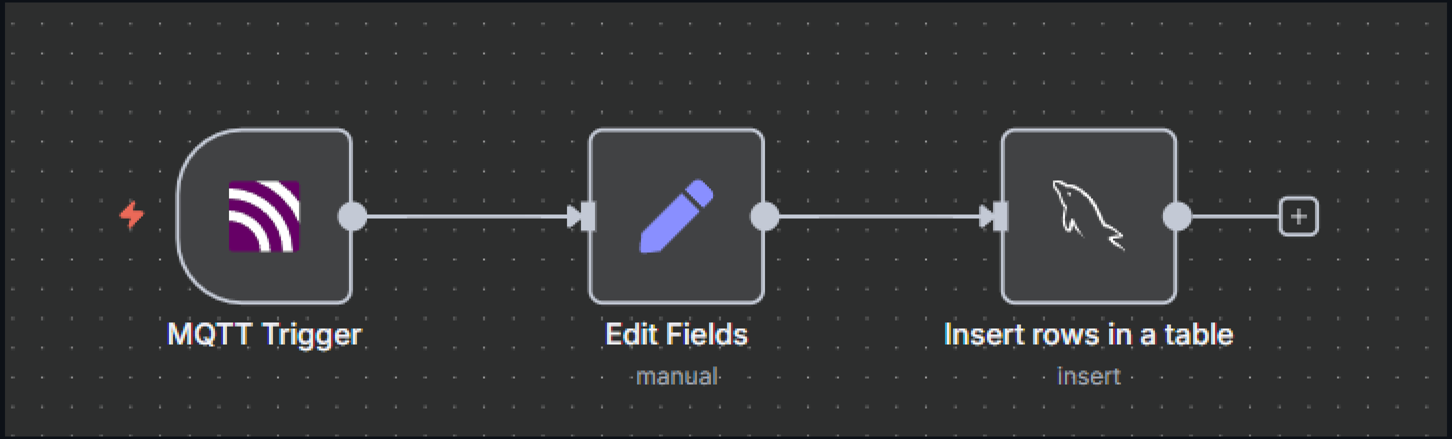
Database

- Create a new database

```
CREATE DATABASE IF NOT EXISTS sensor;
```

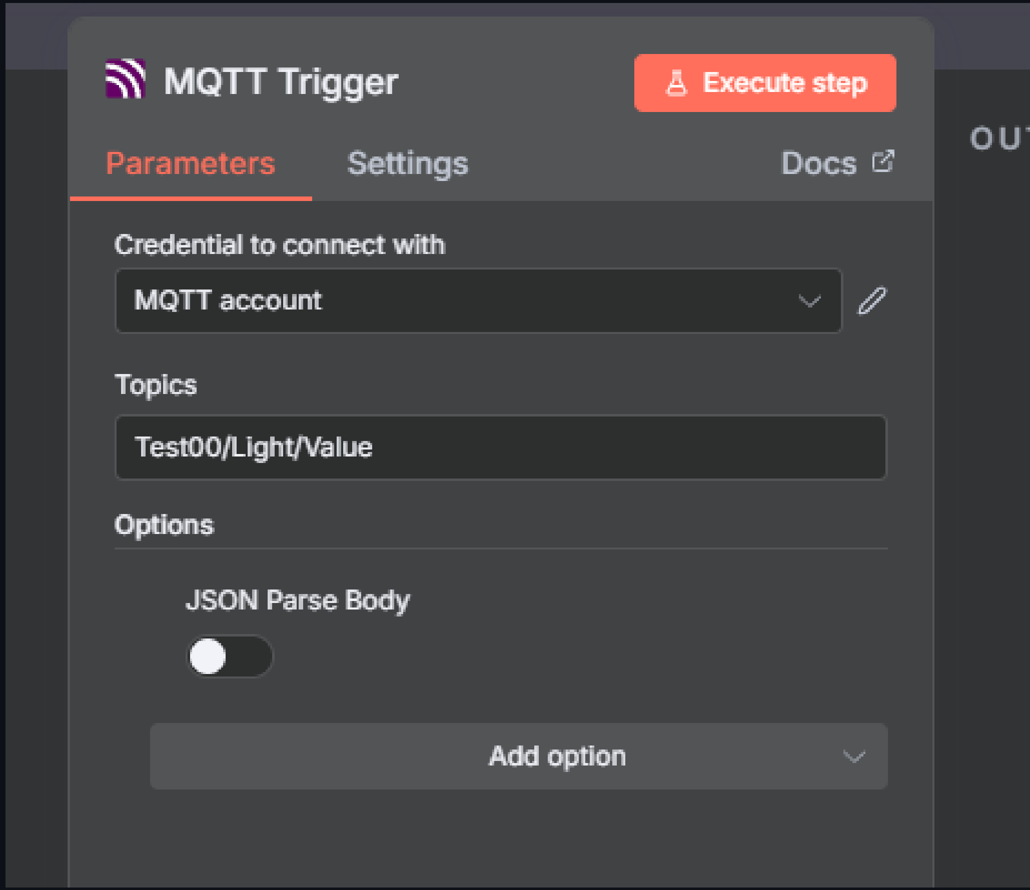
```
CREATE TABLE IF NOT EXISTS sensor.light (
  id INT AUTO_INCREMENT PRIMARY KEY,
  sensor_name VARCHAR(50),
  sensor_value FLOAT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Database Record Flow



MQTT

- Topic: `Test00/Light/Value`



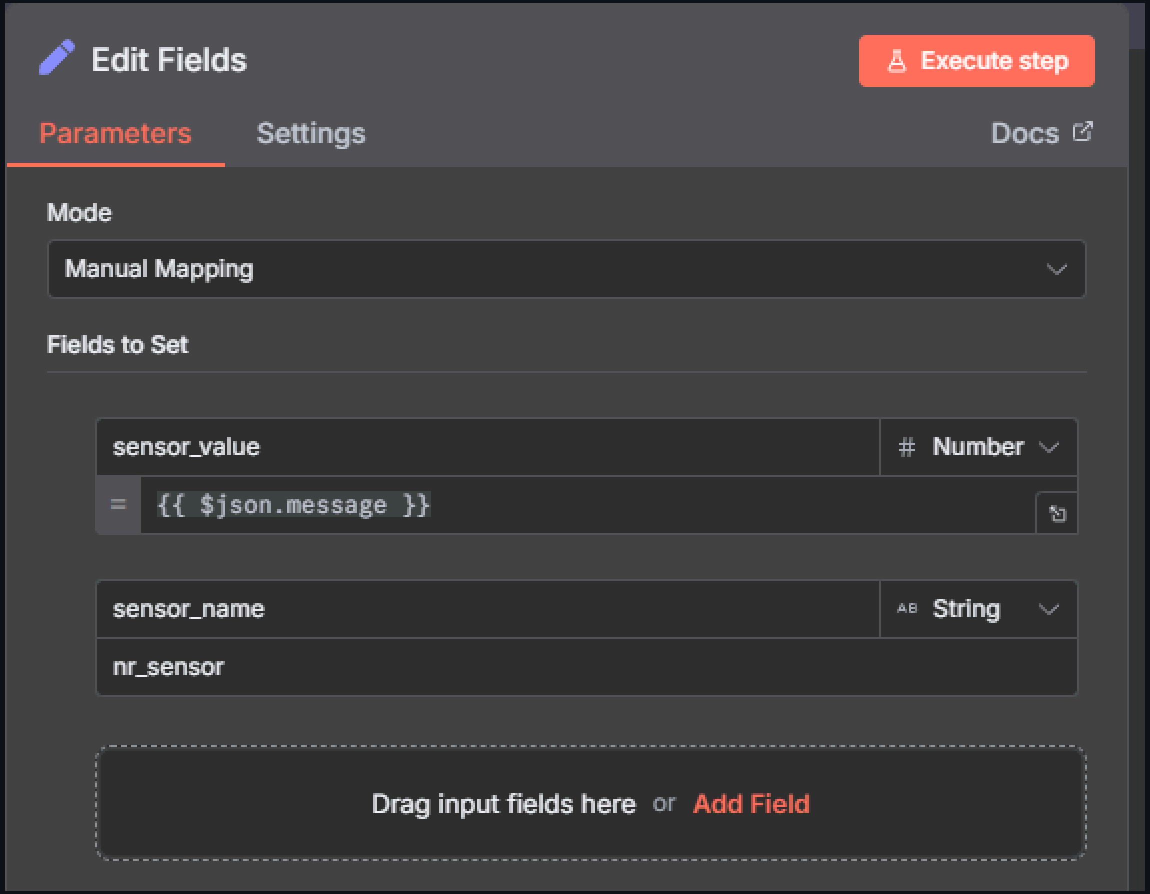
Credential to connect with: MQTT account

Topics: Test00/Light/Value

Options

JSON Parse Body

Edit Fields



Mode

Manual Mapping

Fields to Set

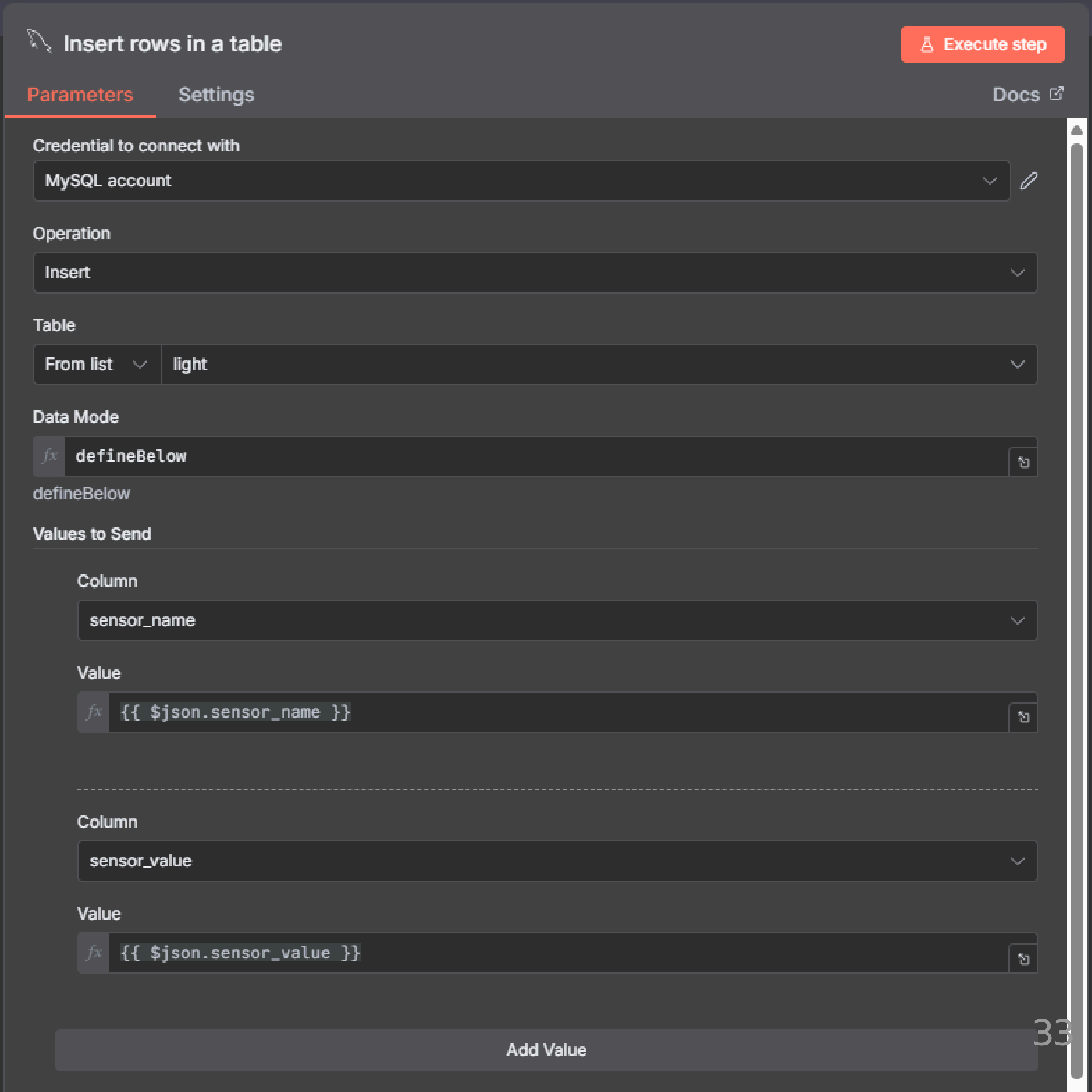
sensor_value — Number

= `{{ $json.message }}`

sensor_name — String

nr_sensor

MySQL Node



Credential to connect with: MySQL account

Operation: Insert

Table: From list | light

Data Mode: defineBelow

defineBelow

Values to Send

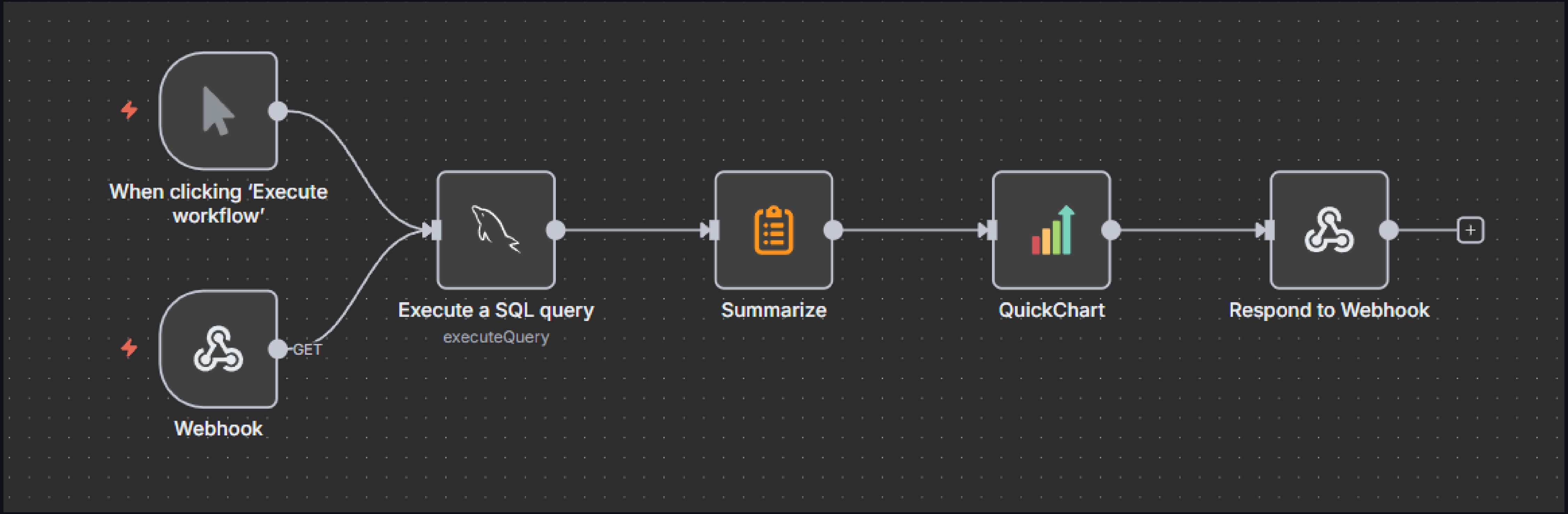
Column: sensor_name

Value:
```
{{ $json.sensor_name }}
```

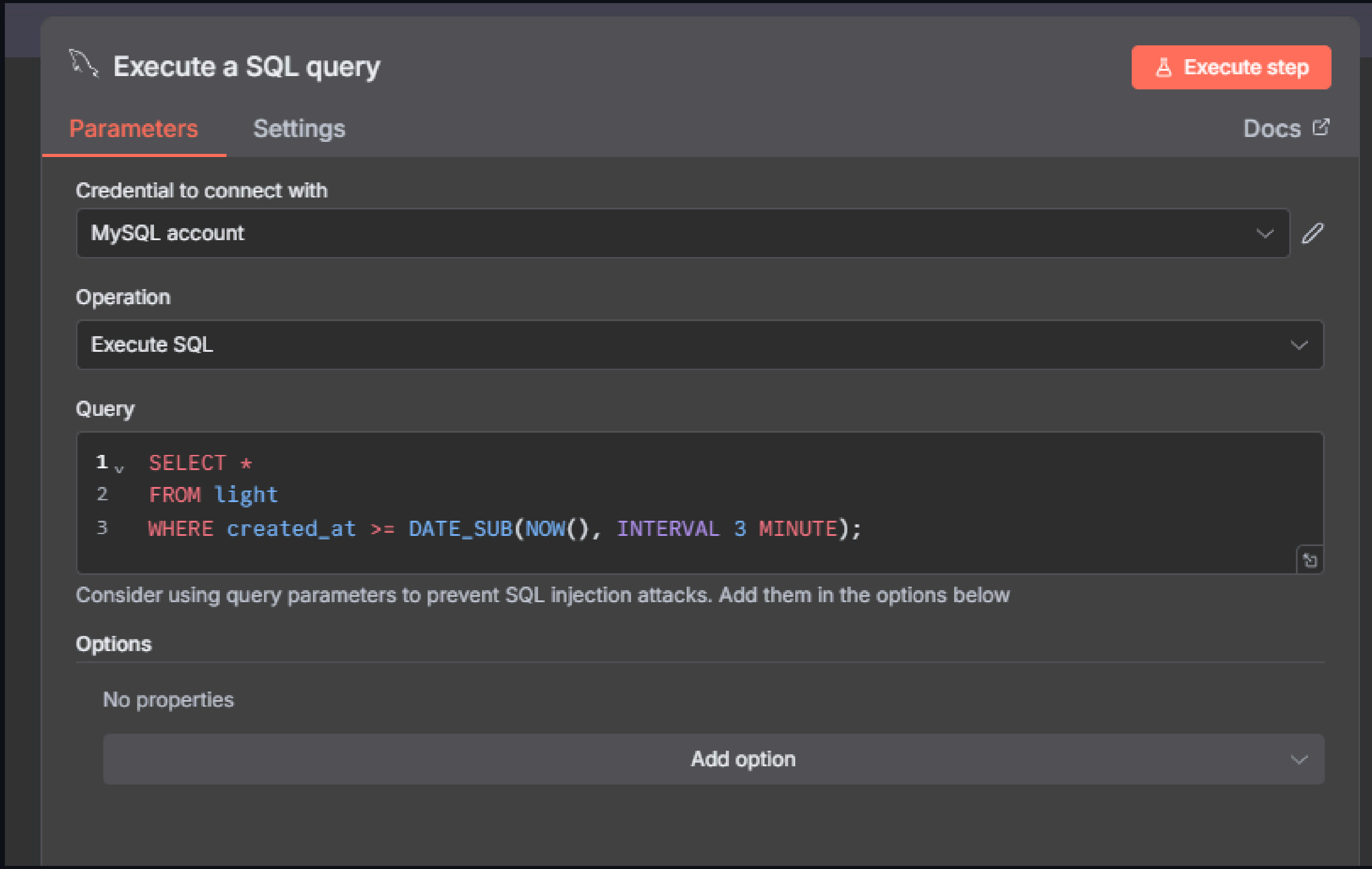
Column: sensor_value

Value:
```
{{ $json.sensor_value }}
```

Viewing Data



MySQL



Credential to connect with

MySQL account

Operation

Execute SQL

Query

```
SELECT *
FROM light
WHERE created_at >= DATE_SUB(NOW(), INTERVAL 3 MINUTE);
```

Consider using query parameters to prevent SQL injection attacks. Add them in the options below

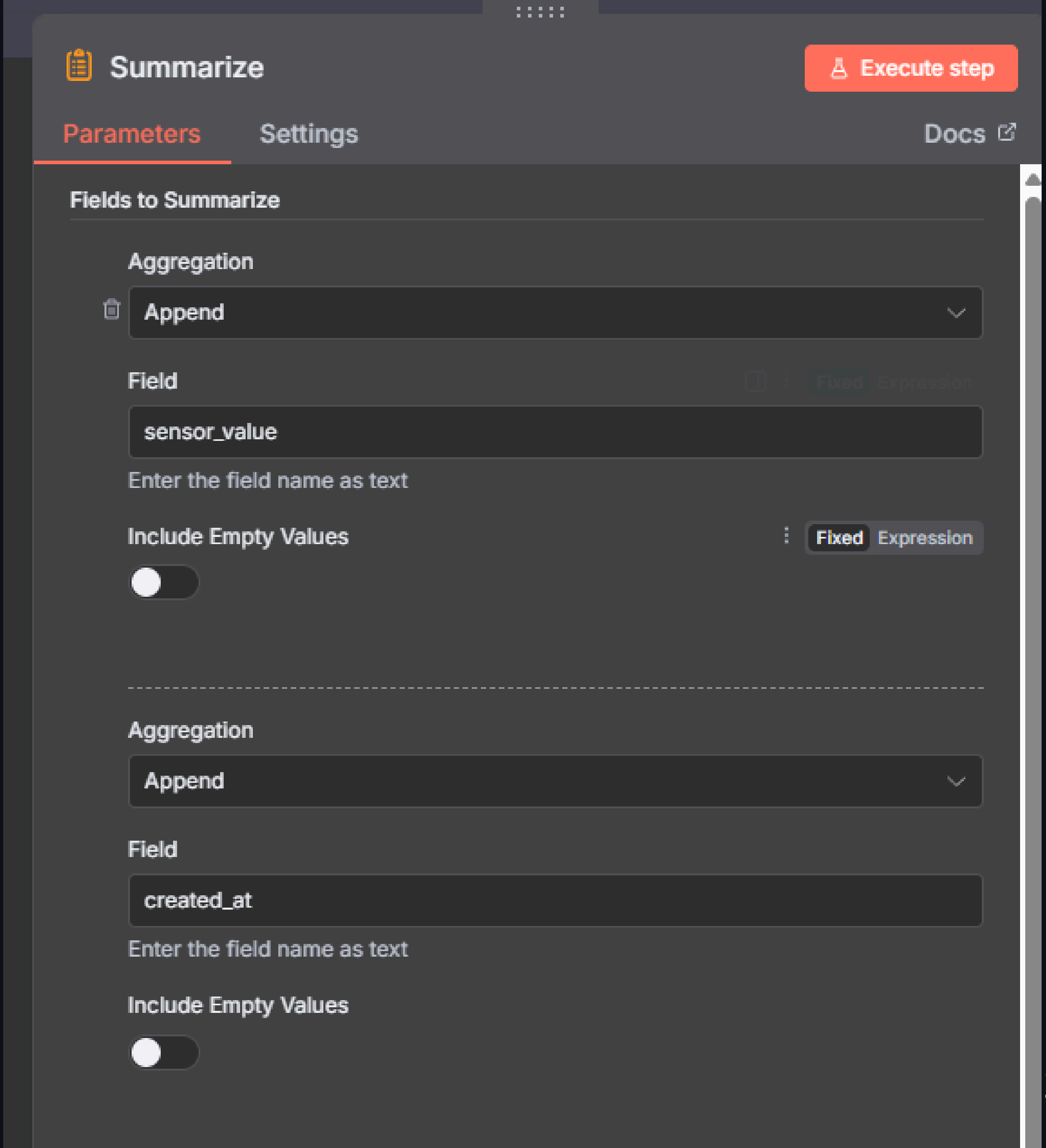
Options

No properties

MySQL

```sql
SELECT *
FROM light
WHERE created_at >= DATE_SUB(NOW(), INTERVAL 3 MINUTE);
```

Summarize



Aggregation: Append

Field: sensor_value

Enter the field name as text

Include Empty Values: off

Aggregation: Append

Field: created_at

Enter the field name as text

Include Empty Values: off

Quick Chart

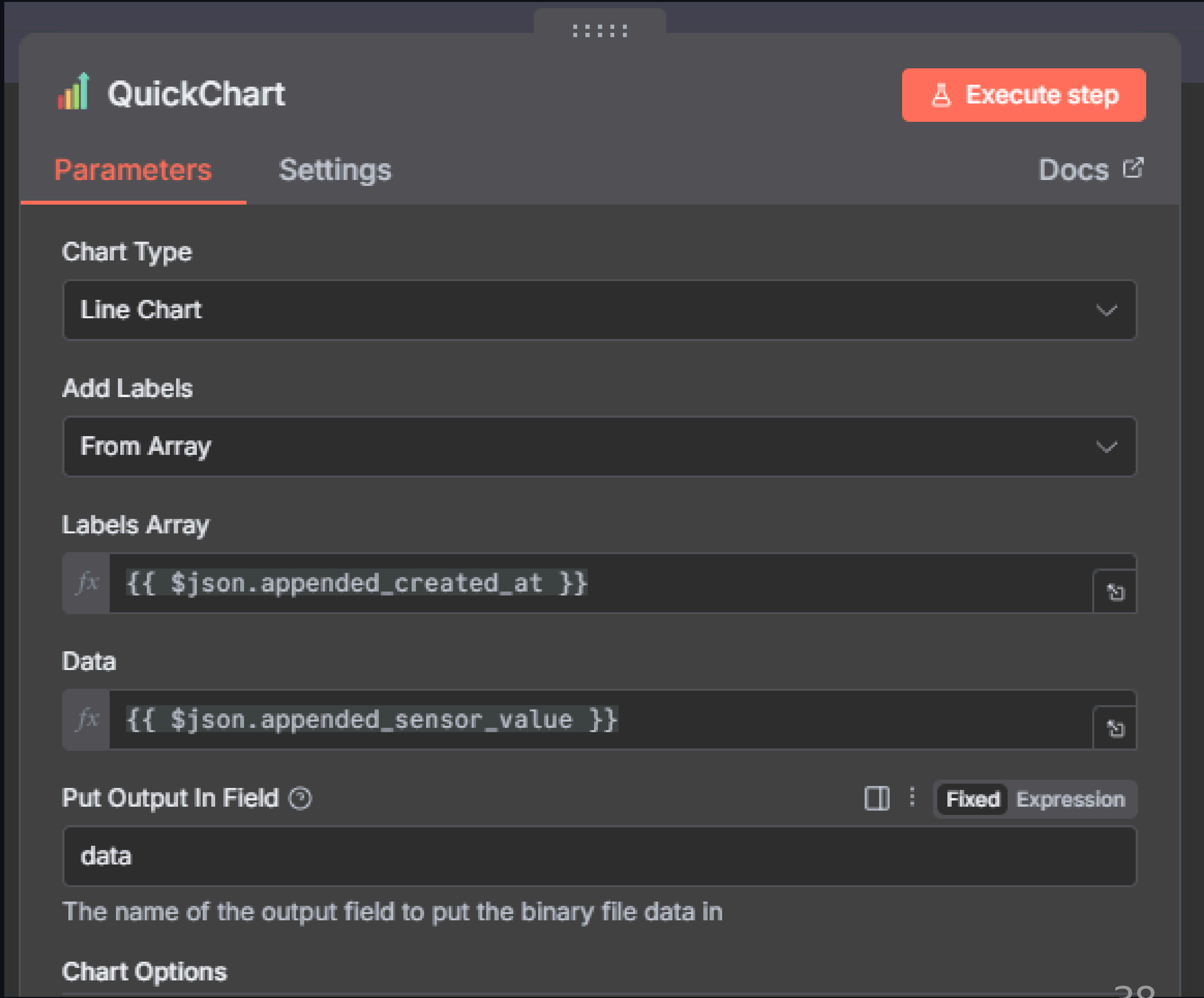


Chart Type

Line Chart

Add Labels

From Array

Labels Array

```
{{ $json.appended_created_at }}
```

Data

```
{{ $json.appended_sensor_value }}
```

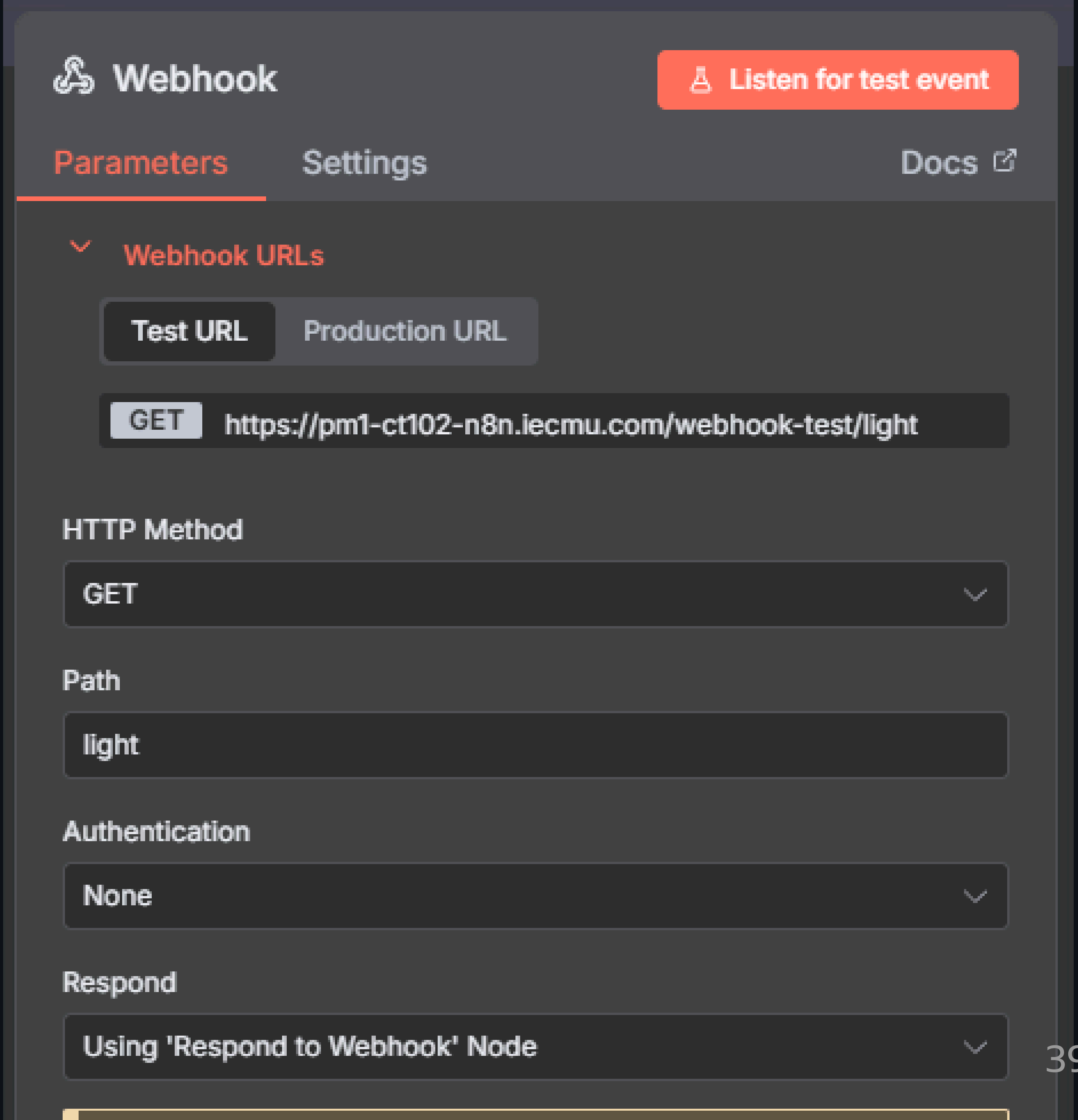
Put Output In Field

data

The name of the output field to put the binary file data in

Chart Options

Webhook


Webhook

Listen for test event

Parameters Settings Docs

Webhook URLs

Test URL Production URL

GET https://pm1-ct102-n8n.iecmu.com/webhook-test/light

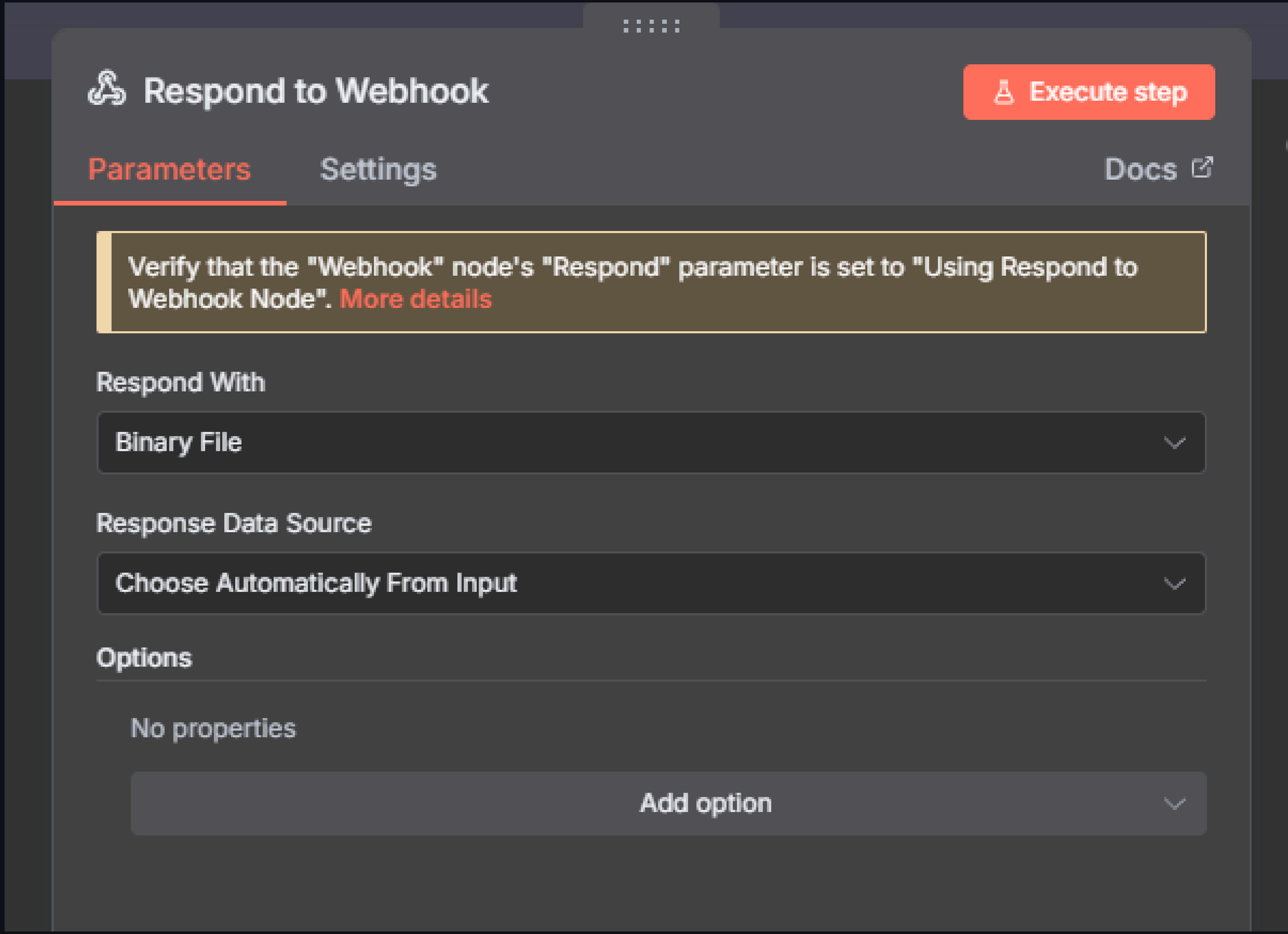
HTTP Method: GET

Path: light

Authentication: None

Respond: Using 'Respond to Webhook' Node

Respond to Webhook


Respond to Webhook

Verify that the "Webhook" node's "Respond" parameter is set to "Using Respond to Webhook Node". More details

Respond With: Binary File

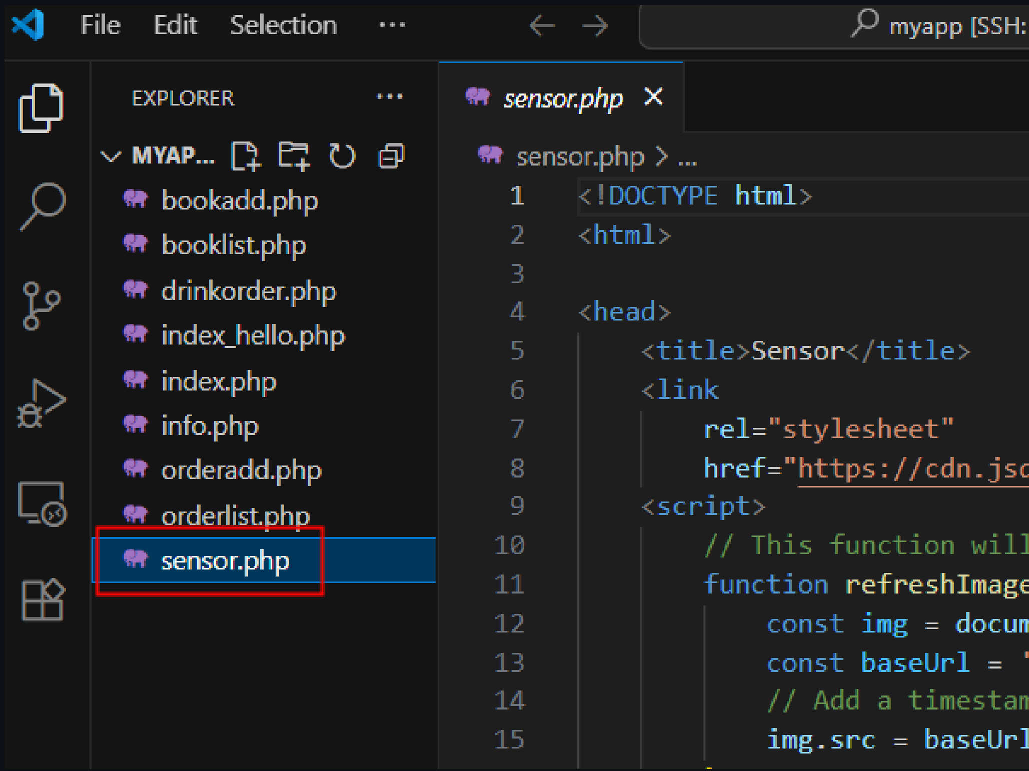
Response Data Source: Choose Automatically From Input

Options

No properties

Web Application

- Create `sensor.php` on the webserver. (Link)
  - Fix the webhook link accordingly.



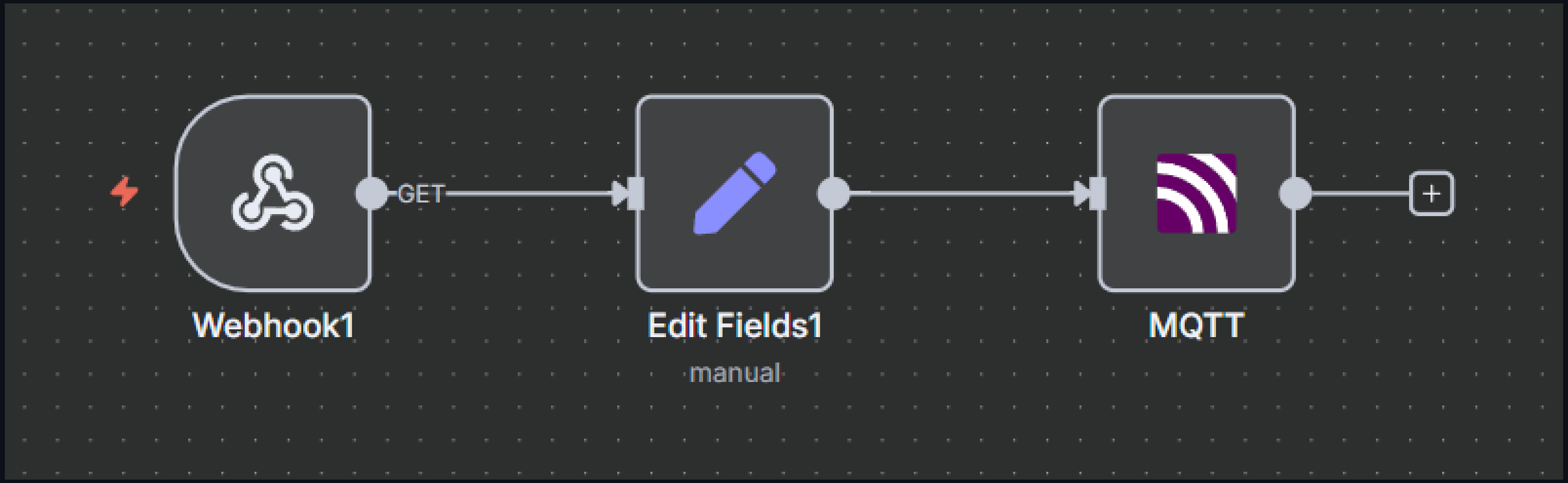
bookadd.php
booklist.php
drinkorder.php
index_hello.php
index.php
info.php
orderadd.php
orderlist.php
sensor.php

```
<!DOCTYPE html>
<html>

<head>
    <title>Sensor</title>
    <link
        rel="stylesheet"
        href="https://cdn.jsc
    <script>
        // This function will
        function refreshImage
            const img = docum
            const baseUrl = '
            // Add a timestam
            img.src = baseUrl
```

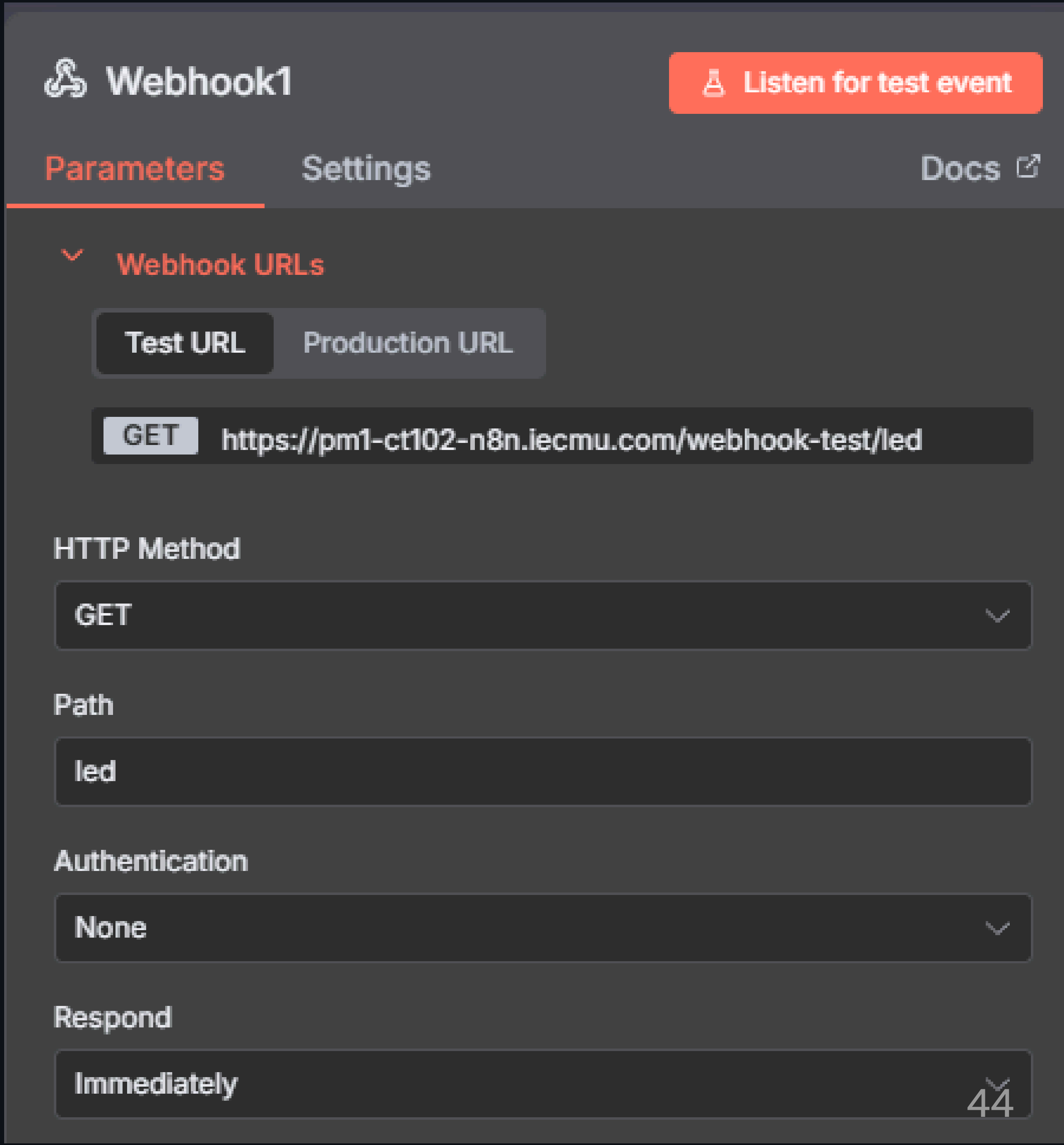
Actuator

> Controlling devices

Control Flow



Webhook



GET https://pm1-ct102-n8n.iecmu.com/webhook-test/led

HTTP Method: GET

Path: led

Authentication: None

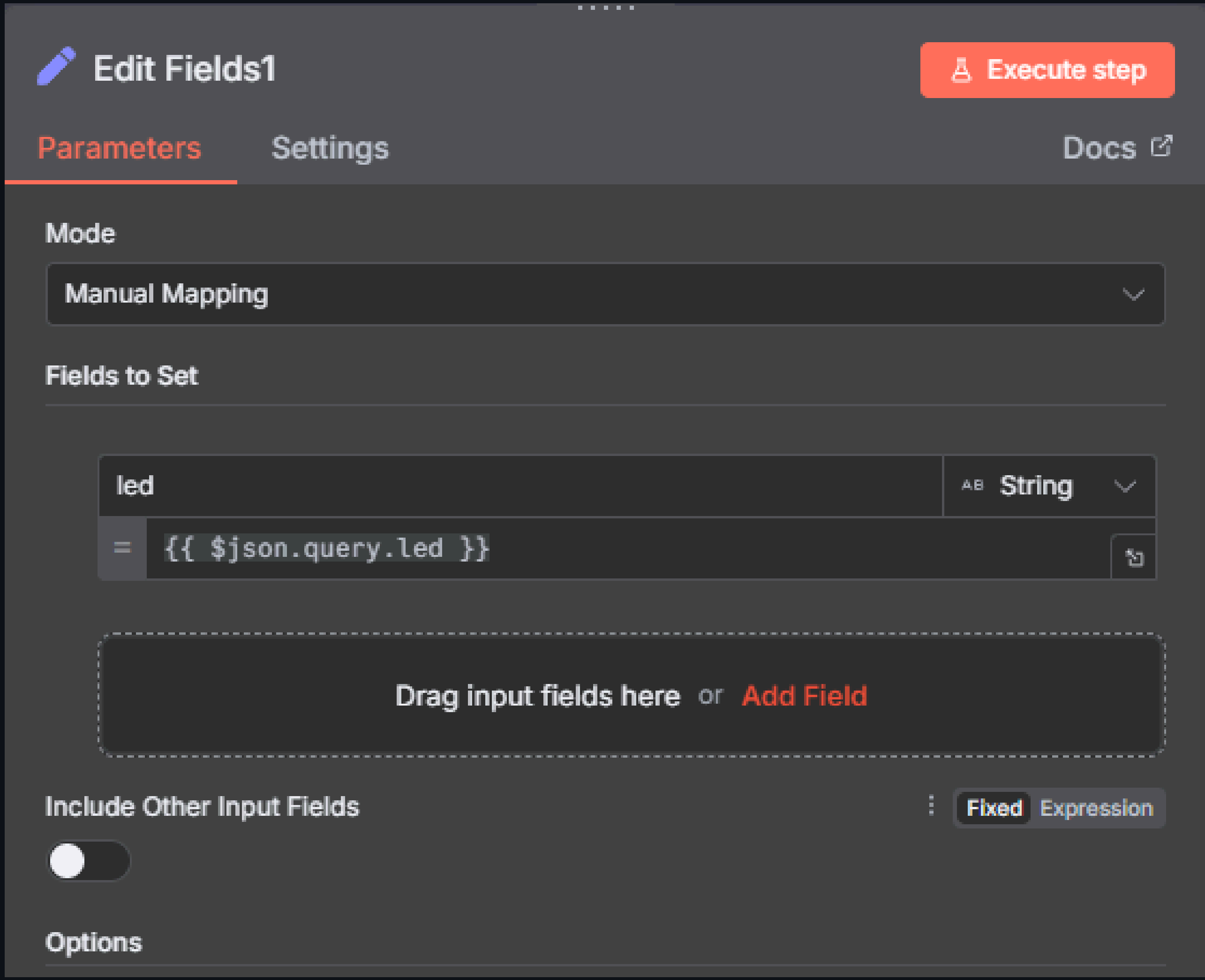
Respond: Immediately

Webhook

We want to send the following (using query parameter)

- ON: https://pmX-ctXXX-n8n.iecmu.com/webhook/led?led=on
- OFF: https://pmX-ctXXX-n8n.iecmu.com/webhook/led?led=off

Edit Fields



Mode

Manual Mapping

Fields to Set

led — String

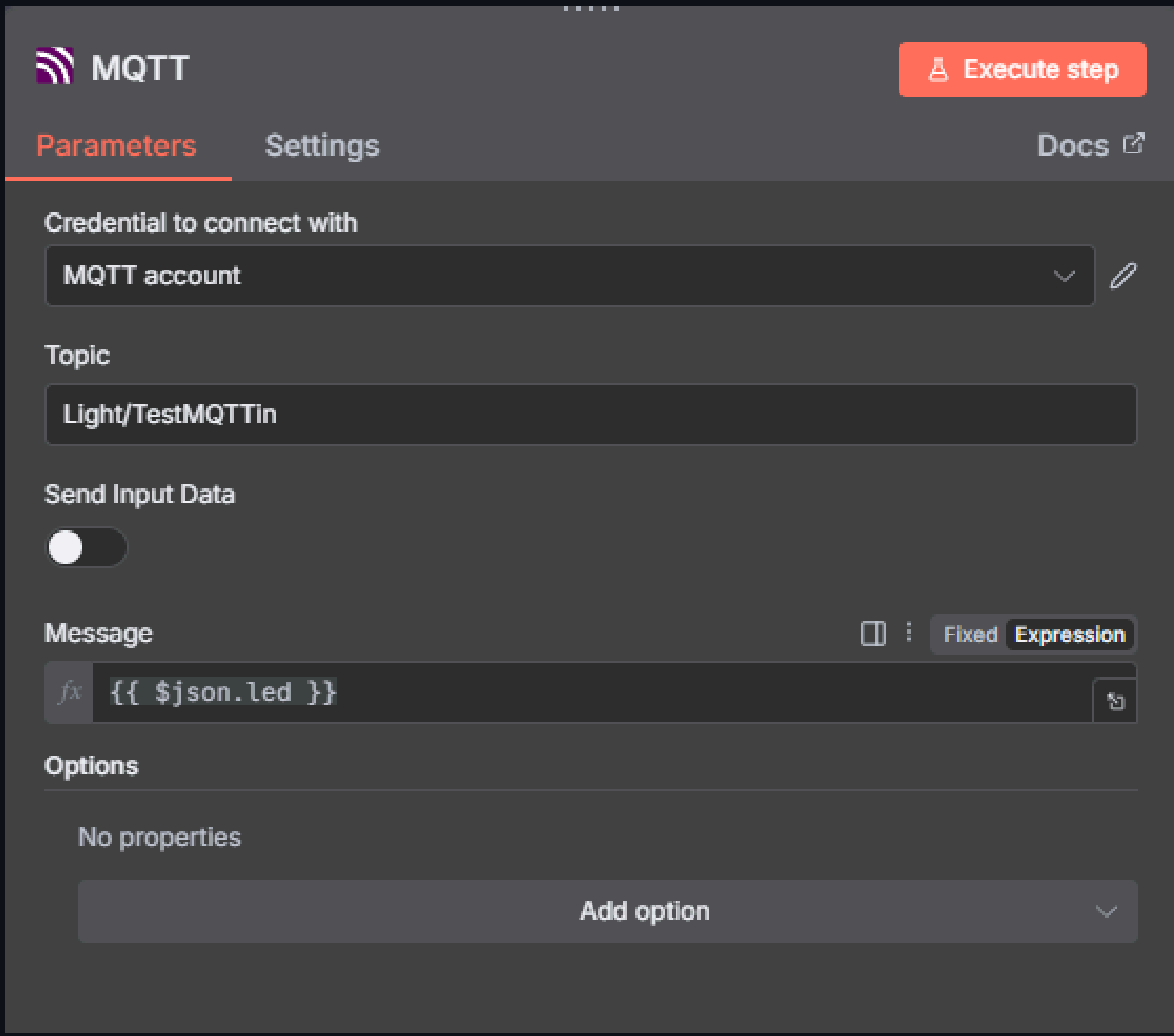
= `{{ $json.query.led }}`

Include Other Input Fields

Options

MQTT

- Topic: `Light/TestMQTTin`



Credential to connect with: MQTT account

Topic: Light/TestMQTTin

Send Input Data: off

Message (Fixed / Expression):

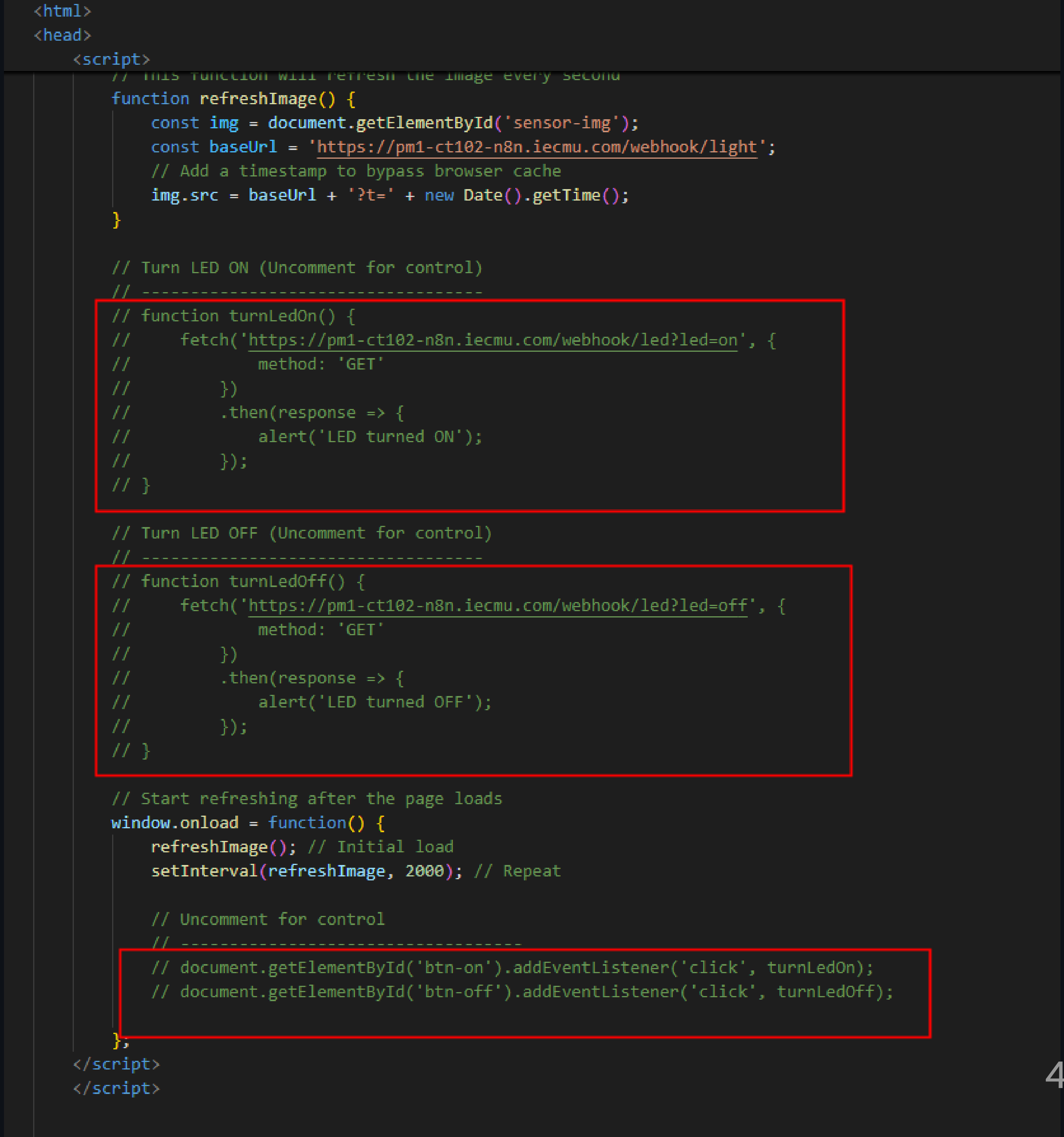
```
{{ $json.led }}
```

Options

No properties

Web Application

- Uncomment



```
<html>
<head>
    <script>
        // This function will refresh the image every second
        function refreshImage() {
            const img = document.getElementById('sensor-img');
            const baseUrl = 'https://pm1-ct102-n8n.iecmu.com/webhook/light';
            // Add a timestamp to bypass browser cache
            img.src = baseUrl + '?t=' + new Date().getTime();
        }

        // Turn LED ON (Uncomment for control)
        // -----------------------------------
        // function turnLedOn() {
        //     fetch('https://pm1-ct102-n8n.iecmu.com/webhook/led?led=on', {
        //             method: 'GET'
        //         })
        //         .then(response => {
        //             alert('LED turned ON');
        //         });
        // }

        // Turn LED OFF (Uncomment for control)
        // -----------------------------------
        // function turnLedOff() {
        //     fetch('https://pm1-ct102-n8n.iecmu.com/webhook/led?led=off', {
        //             method: 'GET'
        //         })
        //         .then(response => {
        //             alert('LED turned OFF');
        //         });
        // }

        // Start refreshing after the page loads
        window.onload = function() {
            refreshImage(); // Initial load
            setInterval(refreshImage, 2000); // Repeat

            // Uncomment for control
            // -----------------------------------
            // document.getElementById('btn-on').addEventListener('click', turnLedOn);
            // document.getElementById('btn-off').addEventListener('click', turnLedOff);

        };
    </script>
    </script>
```