

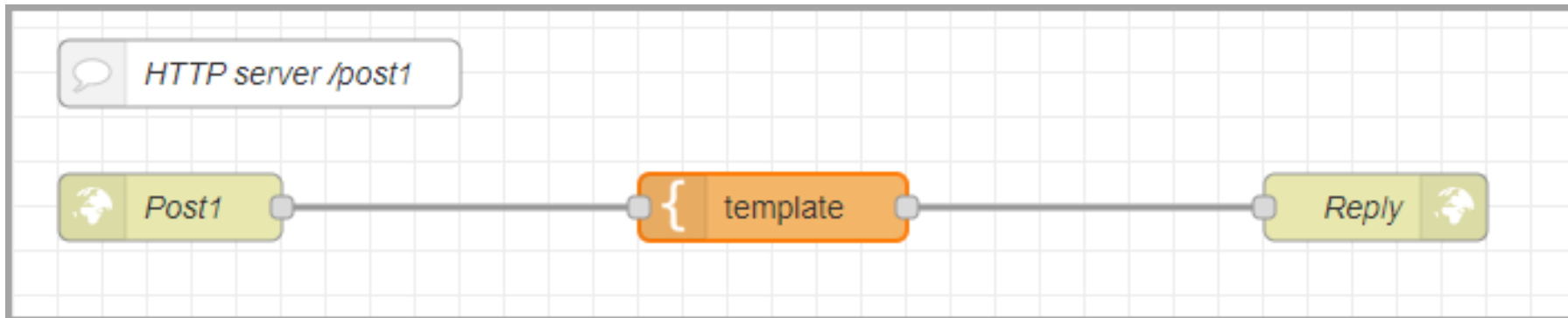
Production Supporting Systems in Factories

ระบบสนับสนุนการผลิตในโรงงานอุตสาหกรรม

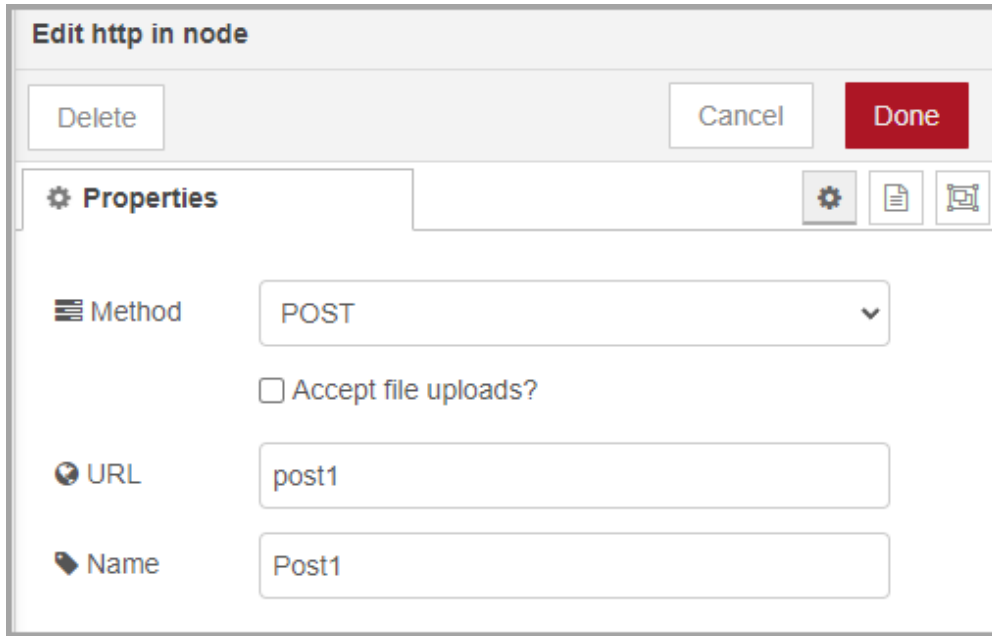
Module 2-1: HTTP POST Server

Server listening to **POST** requests

- Create a server that receives **POST** requests.
 - Flow



- http in node



The screenshot shows a dialog box titled "Edit http in node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a tabbed interface with a "Properties" tab selected. The "Properties" tab contains the following fields:

- Method:** A dropdown menu currently showing "POST".
- Accept file uploads?:** An unchecked checkbox.
- URL:** A text input field containing "post1".
- Name:** A text input field containing "Post1".

- Send the transformed message back.
- Try sending the post requests to your friends public address.

- **template node**

Edit template node

Delete

Cancel

Done

Properties

Name

Property

▼ msg. payload

Template

Syntax Highlight:

mustache

1

I got your message: `{{payload}}` !

- http response node

Edit http response node

Delete Cancel Done

⚙️ **Properties** ⚙️ 📄 🔄

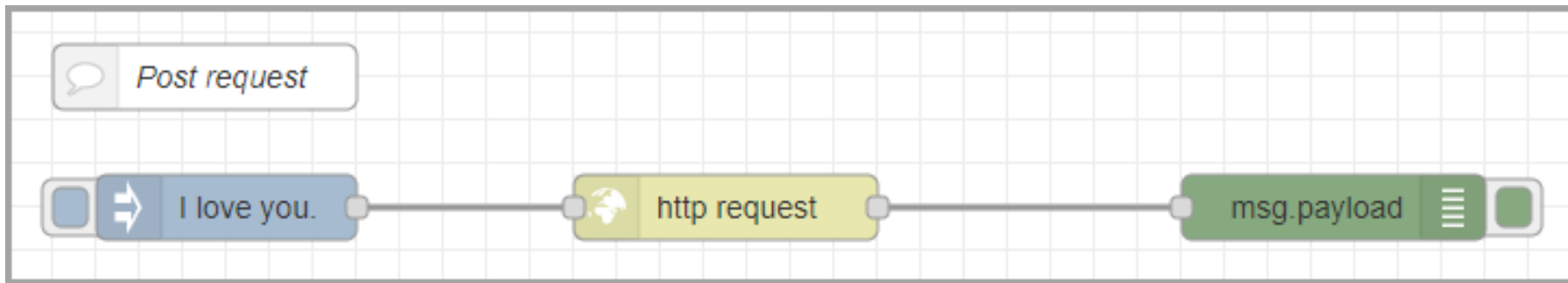
🏷️ Name Reply

← Status code msg.statusCode

☰ Headers

Send POST request from node-red

- Create a **post** request
 - Flow



- inject node

The screenshot shows a dialog box titled "Edit inject node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a section labeled "Properties" with a gear icon and three sub-icons (a gear, a document, and a monitor). Under "Properties", there is a "Name" field with a dropdown arrow and the text "Name". Below this, there are three rows of configuration fields, each with a dropdown arrow and a text input field, and a small "x" button to the right of each row. The first row is labeled "msg. payload" and contains the text "I love you.". The second row is labeled "msg. topic" and contains the text "a_z". The third row is labeled "msg. url" and contains the text "http://localhost:1880/post1".

- Note: If there is a connection error, use `http://127.0.0.1:1880/post1`
 - What?

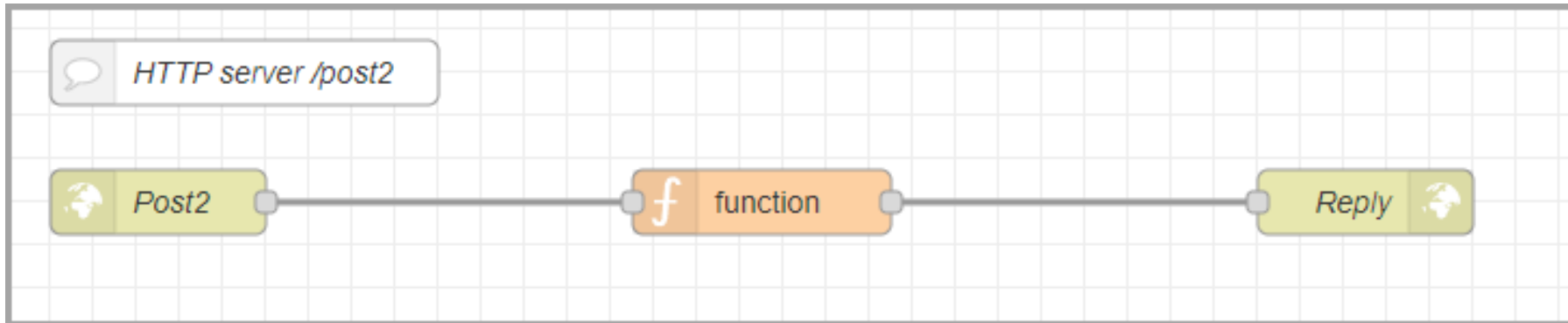
Send POST request from another program

Software

- Postman Desktop
- Chrome
 - Advancd REST Client
- Firefox
 - RESTClient

Serving dynamic content

- Create another url that response with dynamic content.
 - Flow



- `http in node`
 - `url = post2`

- function node



(code on the next page)

```
const message = msg.payload;
const date = new Date();
const time = date.toLocaleTimeString();

msg.payload = `I got your message "${message}" at ${time} !`;
return msg;
```

- Try send in a post message to `.../post2`.
- Also try sending a post message to your friend's public address.

Class Attendance

การทำเรื่องง่ายๆให้มันยาก

วิธีการเช็คชื่อ

- Send `Post` request to

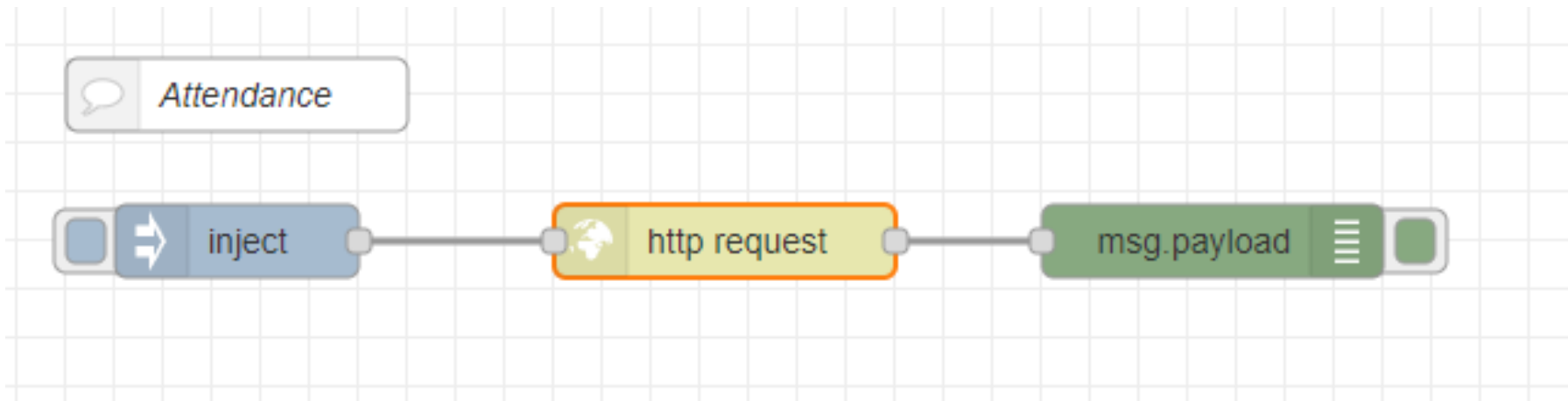
`https://materials-line-oa-production.up.railway.app/prodsup`

- Payload

- `cmu_id` : your 9-digit student id
- `code` : 4-character code from the screen (ไม่เอา `CODE:`)

```
{  
  "cmu_id": "111222333",  
  "code": "xxxx"  
}
```

- Flow



- inject node
 - msg.payload

Edit inject node > **JSON editor**

Cancel Done

Edit JSON Visual editor

format JSON

```
1 {  
2   "cmu_id": "111222333",  
3   "code": "xxxx"  
4 }
```

- http request node

Edit http request node

Delete

Cancel

Done

Properties

Method

POST

▼

URL

https://materials-line.herokuapp.com/prodsup

☐

Enable secure (SSL/TLS) connection

☐

Use authentication

☐

Enable connection keep-alive

☐

Use proxy

☐

Only send non-2xx responses to Catch node

Return

a UTF-8 string

▼

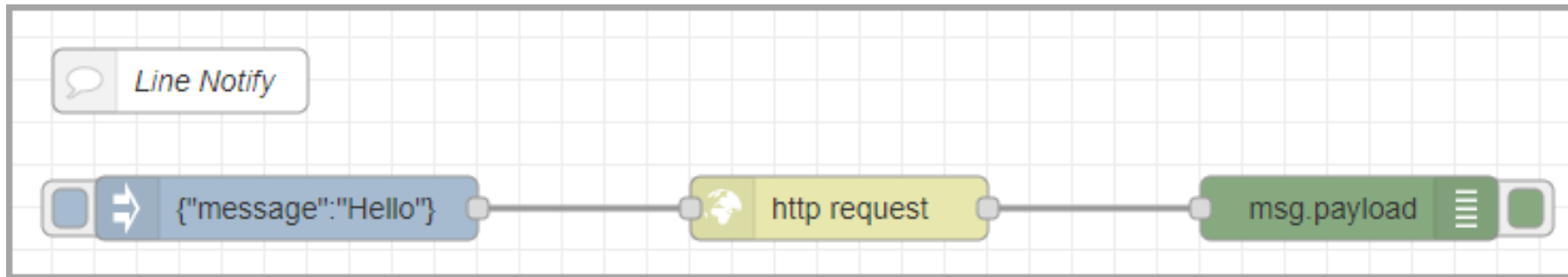
Name

Name

Module 2-2: Line Notify

- ສ້າງກຸ່ມ LINE ກາຍໃນ Group
- ໄປທີ່ <https://notify-bot.line.me/en/>
 - Login (need to verify number on smart phone)
 - Go to My page
 - Click Generate token
 - Enter Token Name and select a chat room.
 - Copy the token and store it somewhere.

- Let send a line notification through Node-Red
 - Flow



- inject node

Edit inject node

Delete Cancel Done

JSON **Edit**

⚙ Properties

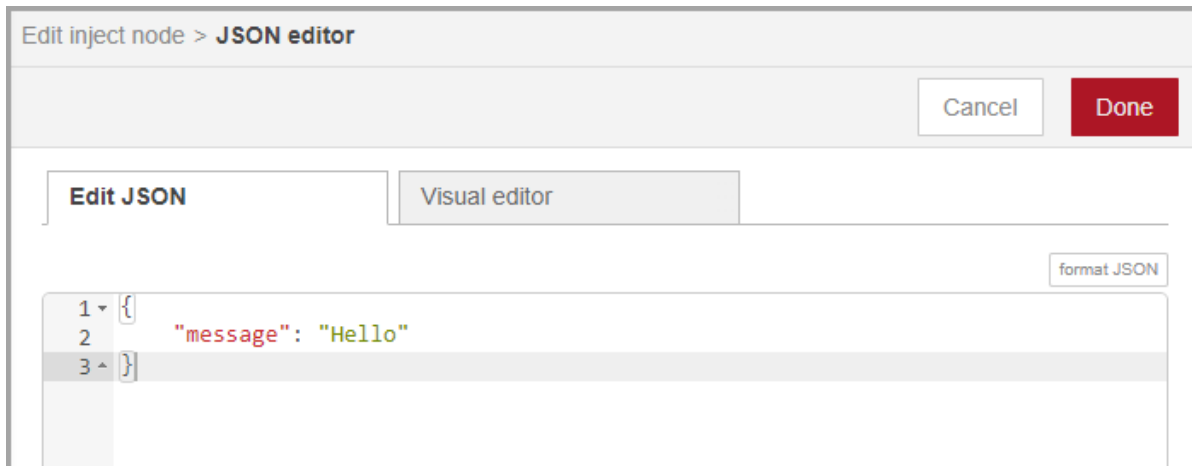
📌 Name Name

msg. payload = {} {"message":"Hello"} ... x

msg. headers = {} {"Content-Type":"application/x-www-..."} ... x

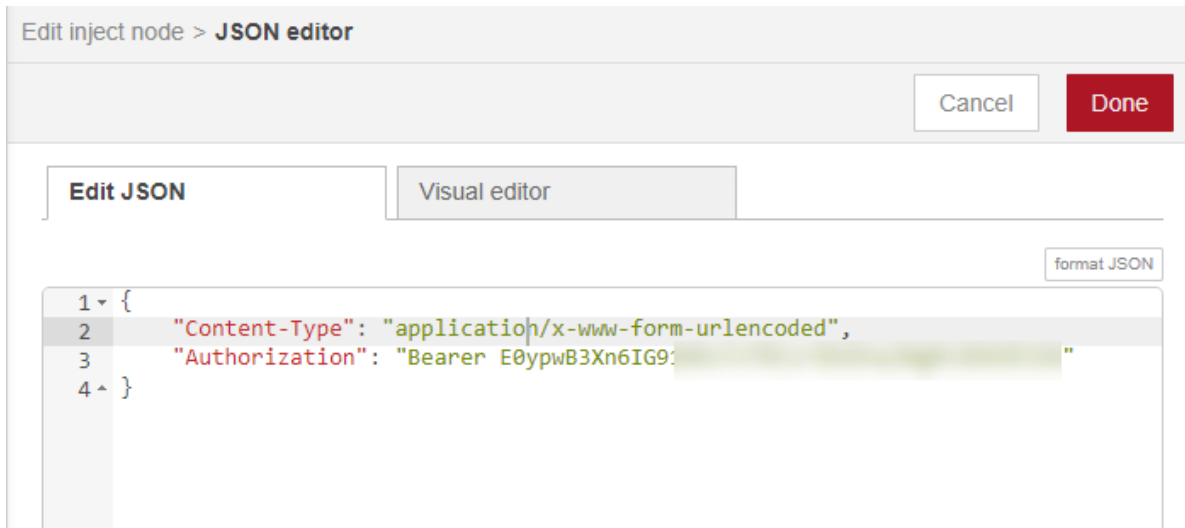
- Set `msg.payload` to (JSON)

```
{  
  "message": "Hello"  
}
```



- Set `msg.headers` to (JSON)

```
{  
  "Content-Type": "application/x-www-form-urlencoded",  
  "Authorization": "Bearer <<YOUR LINE TOKEN>>"  
}
```



- `http request node`
 - **Method:** `POST`
 - **URL:** `https://notify-api.line.me/api/notify`

Edit http request node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

☰ Method

POST

▼

🌐 URL

https://notify-api.line.me/api/notify

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

⬅️ Return

a UTF-8 string

▼

🔖 Name

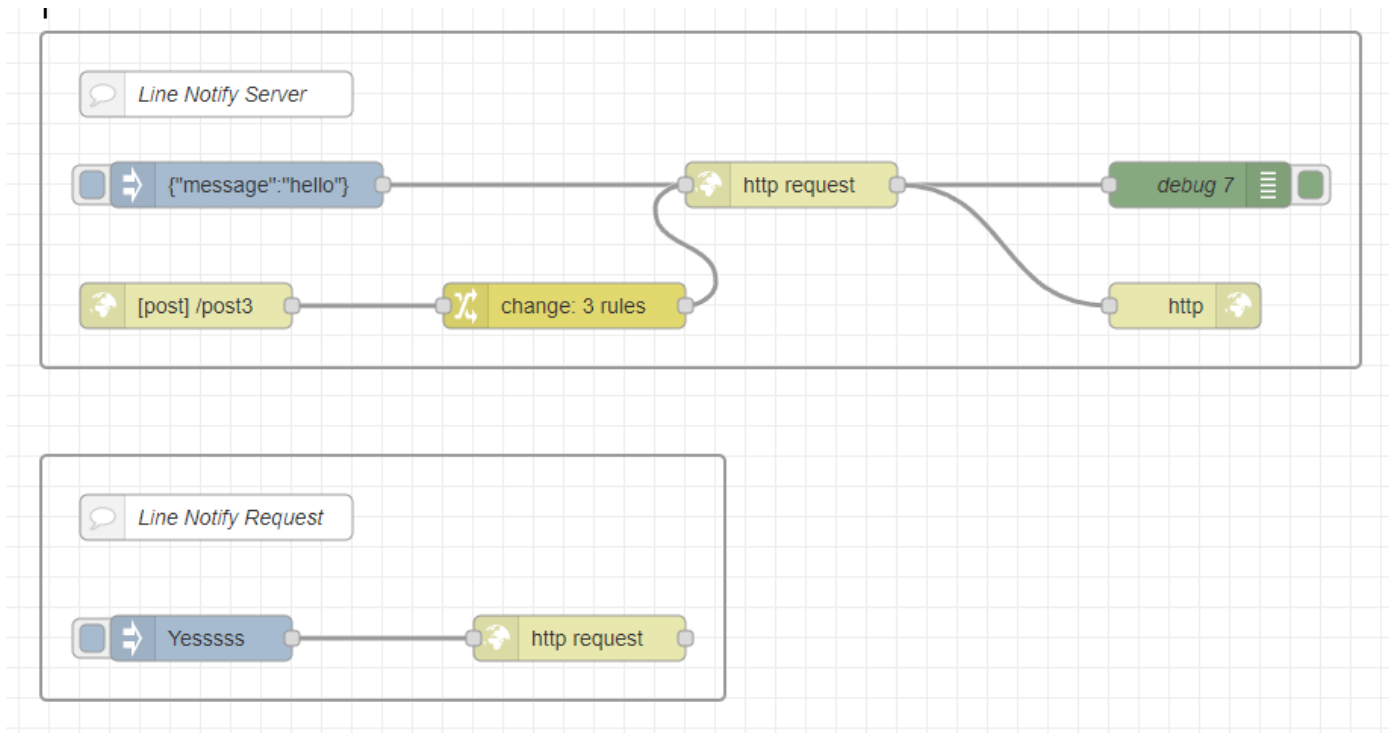
Name

- Add `debug` node to see the output.
 - You can inspect the error.
- Try injecting to see if the notification.

Extra

- Make a server that send the message to `Line`.
 - The message comes from the `POST` request.

Extra



Extra

Edit change node

Delete

Cancel

Done

Properties

Name

Rules

Set

▼

msg. payload

to the value

▼ {} {"message":"Helooooo"}

Set

▼

msg. headers

to the value

▼ {} {"Content-Type":"application/x-www-fi

Set

▼

msg. url

to the value

▼ ^a_z https://notify-api.line.me/api/notify