

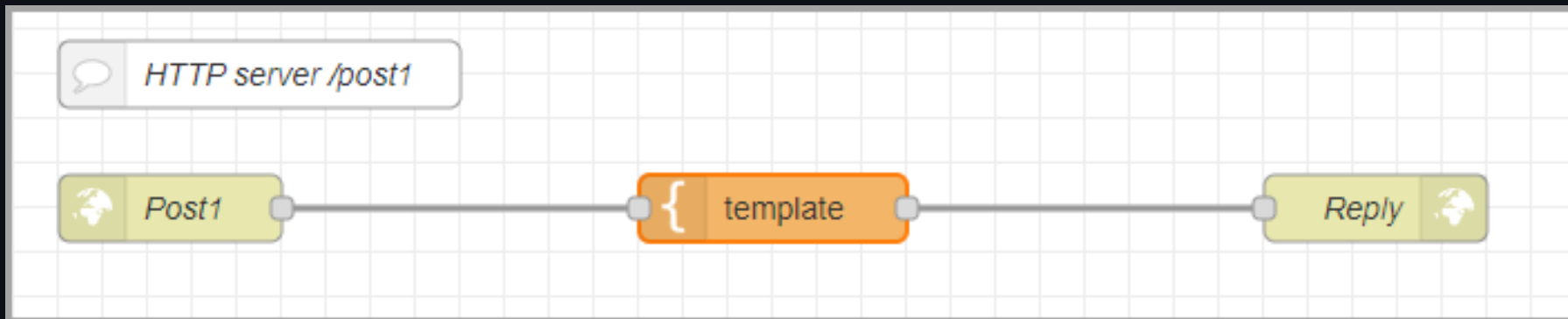
# **Production Supporting Systems in Factories**

**ระบบสนับสนุนการผลิตในโรงงานอุตสาหกรรม**

# 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:  ↗️

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

- http response node

**Edit http response node**

Delete Cancel Done

⚙️ **Properties** ⚙️ 📄 🔄

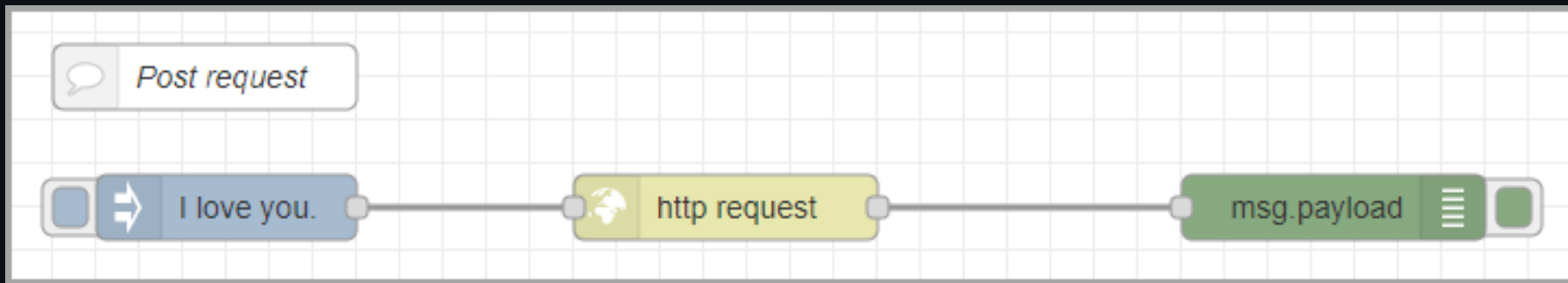
🏷️ Name Reply

← 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 these is a section labeled "Properties" with a gear icon and three sub-icons (a gear, a document, and a screen). Under "Properties", there is a "Name" field with a lock icon and the text "Name". Below this is a list of three properties, each with a menu icon on the left, a label, an equals sign, a value field, and a close button on the right. The first property is "msg. payload" with the value "I love you.". The second property is "msg. topic" with the value "a\_z". The third property is "msg. url" with the value "http://localhost:1880/post1".

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

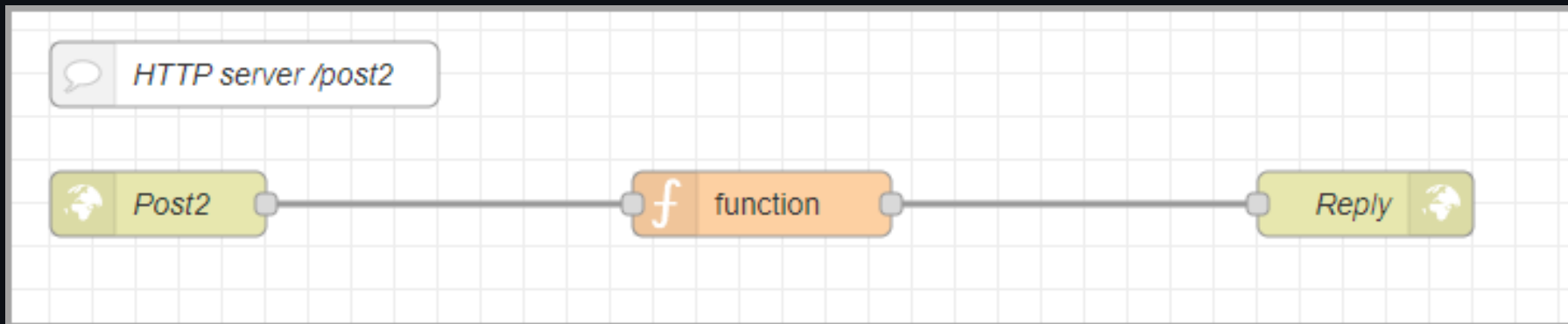
**Send POST request from another program**

# Software

- Postman Desktop
- Chrome
  - Talent API Tester
- 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 you can use `ngrok` to create a public url and send POST request across the internet.