

物聯網實務 HW6

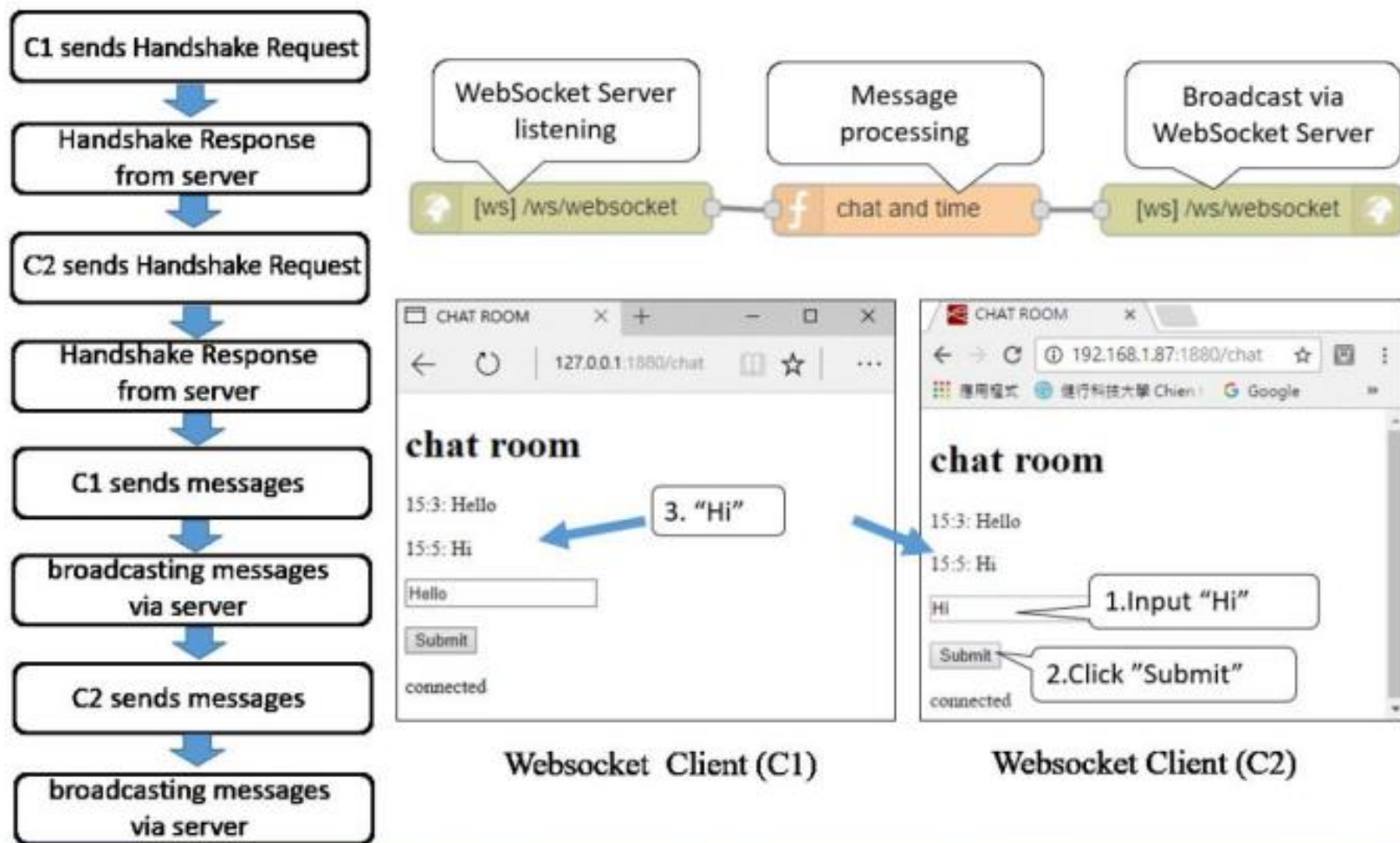
電機碩一 11278008 林佳慧

日期:2023/10/18

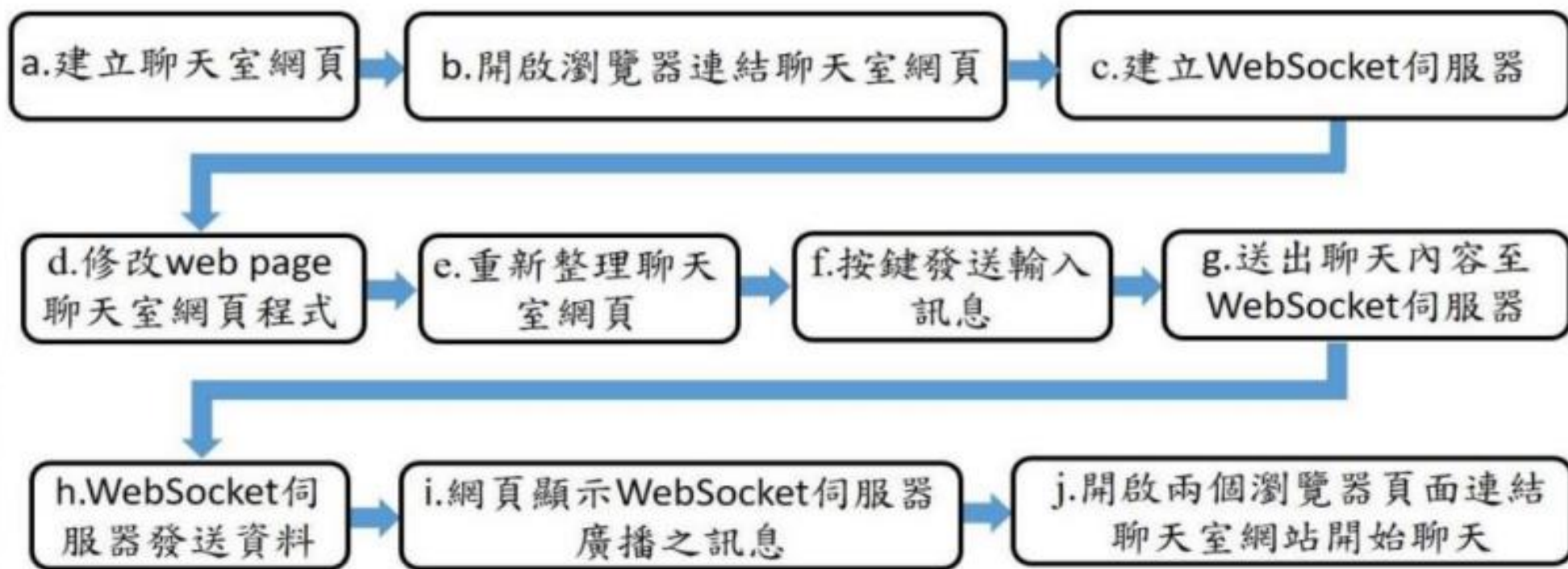
Exercise 6-1

Design a chat room

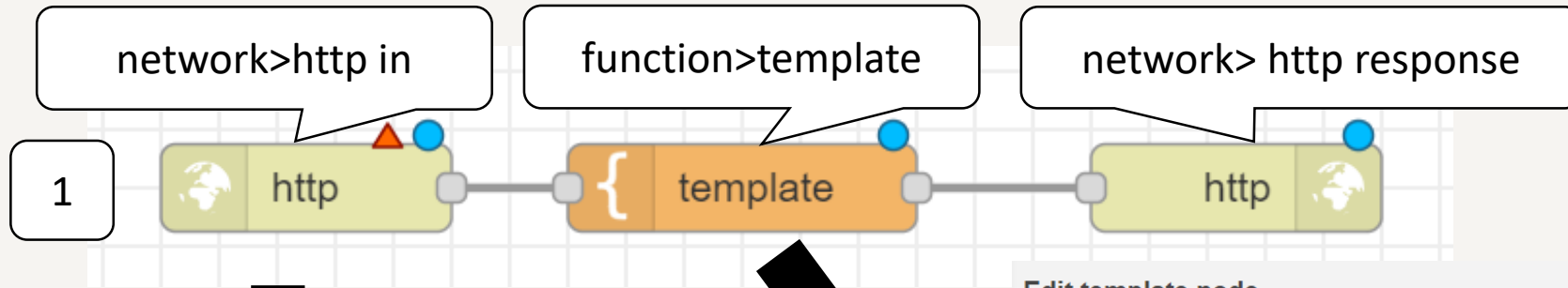
Design a chat room



Processes



Step1: Create chat room HTML



Edit http in node

Delete

Cancel

Done

Properties

Method

GET

URL

/chat

Name

Name

Edit template node

Delete

Cancel

Done

Properties

Name

Name

Property

msg. payload

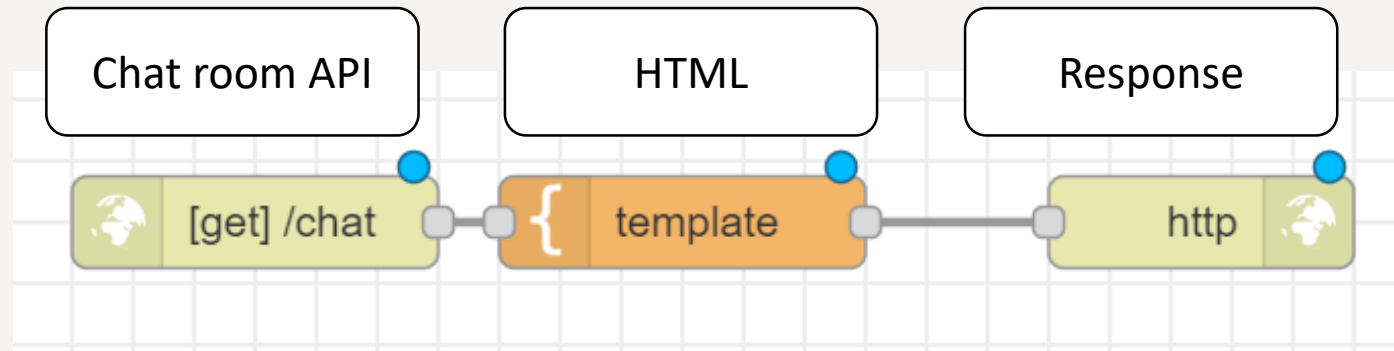
Template

Syntax Highlight: HTML

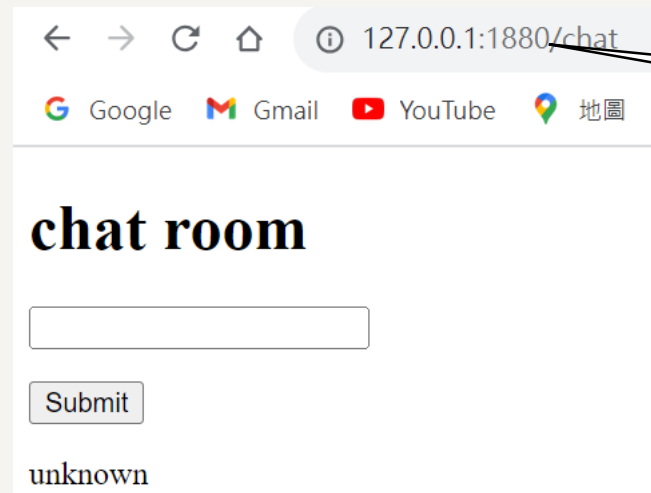
```
15 <p></p>
16 <button>Submit</button>
17 <p></p>
18 <div id="status">unknown</div>
19 </body>
20
21 </html>
```

6-1-1.txt

Chat Room API Flow

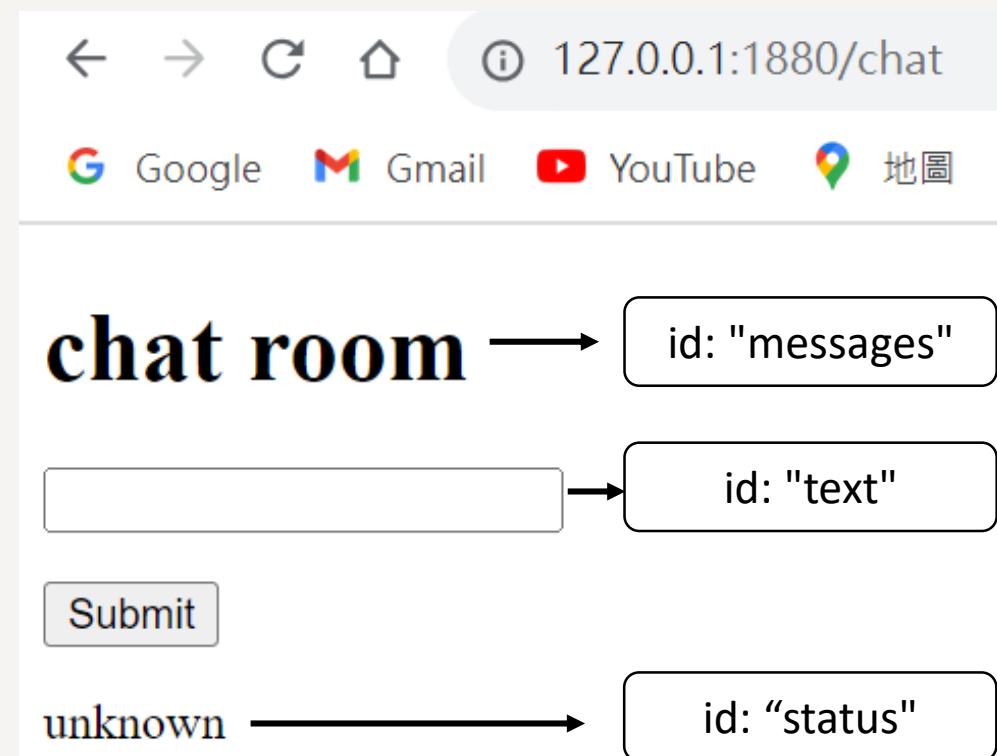


Step2:open the chat room webpage



<http://localhost:1880/chat>

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>CHAT ROOM</title>
  </head>
  <body>
    <div id="messages"> <h1>chat room</h1> </div>
    <form>
      <input type="text" id="text" >
    </form>
    <p></p>
    <button>Submit</button>
    <p></p>
    <div id="status">unknown</div>
  </body>
</html>
```



Step3: Add “websocket in”

The screenshot illustrates the process of adding a "websocket in" node to a Node-RED flow. The interface is divided into several sections:

- Left Panel (Palette):** Contains various input and output nodes. The "websocket in" node is highlighted with a red box.
- Main Canvas:** Shows a flow with nodes: "[get] /chat" → "template" → "http". A new "websocket" node is being added to the flow, indicated by a blue dot and a callout "1. Double click".
- Edit websocket in node dialog:** A modal window for configuring the new node. It includes:
 - Properties:** A dropdown menu for "Type" with "Listen on" selected, highlighted by a red box and callout "2. Listen on".
 - Path:** A text input field containing "/public/test", highlighted by a red box and callout "4. /public/test".
 - Send/Receive:** A dropdown menu set to "payload".
 - Buttons:** "Delete", "Cancel", and "Done".
 - Callout "3. Add new":** Points to a small icon in the "Path" field.
 - Callout "5":** Points to the "Add" button in the top right corner of the dialog.
- Right Panel (Properties):** Shows the configuration for the selected node, including the "Path" field with "/public/test" and a description of the "payload" field.

Edit websocket in node

Delete

Cancel

Done

⚙ Properties



🎯 Type

Listen on



📁 Path

/public/test



🏷 Name

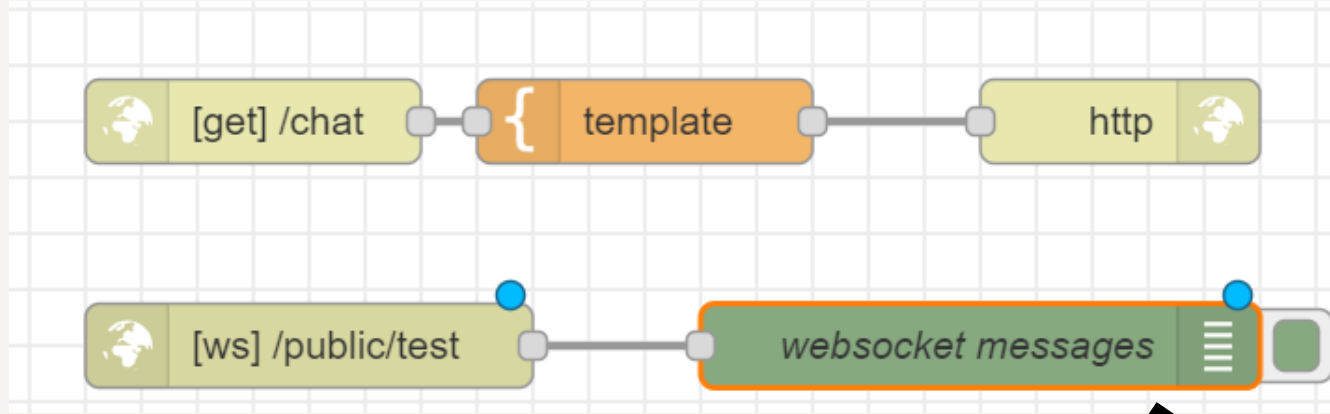
Name

6



[ws] /public/test

Step 4: Add a debug node



Edit debug node

Delete

Cancel

Done

Properties

Output

msg. payload

To

☒ debug window

☐ system console

☐ node status (32 characters)

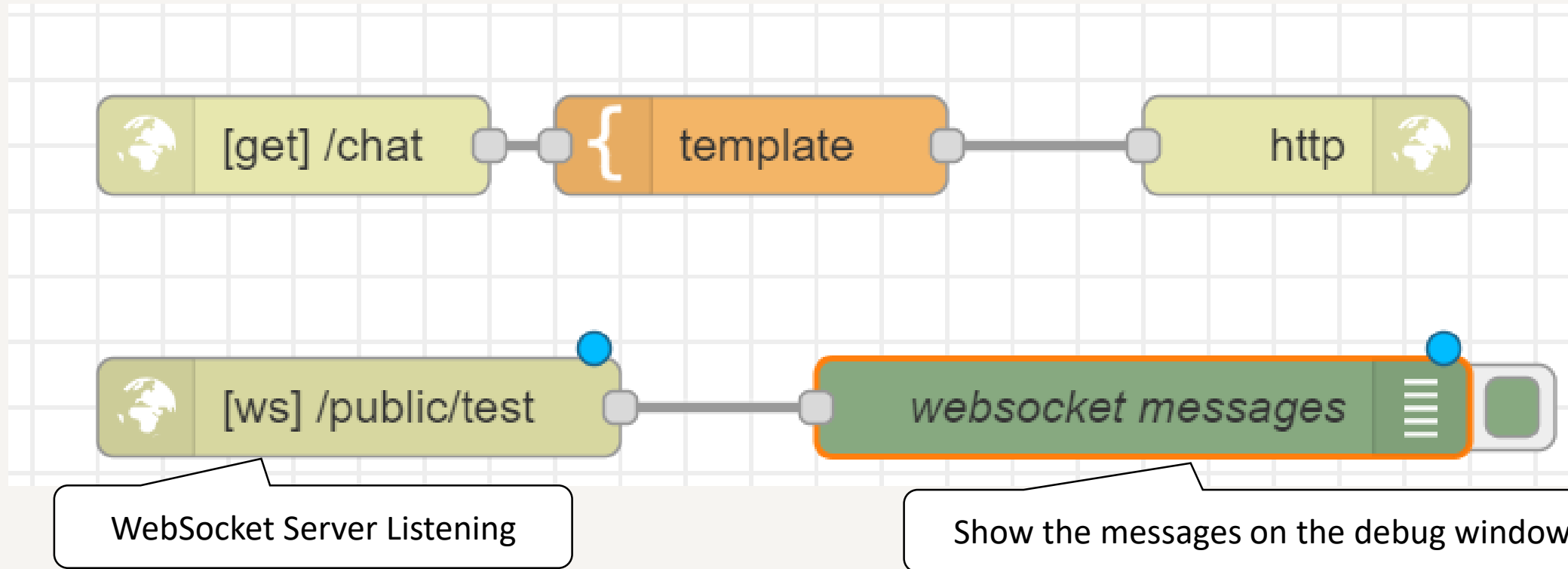
Name

websocket messages



Deploy

WebSocket Flow



Step 5: Modify HTML

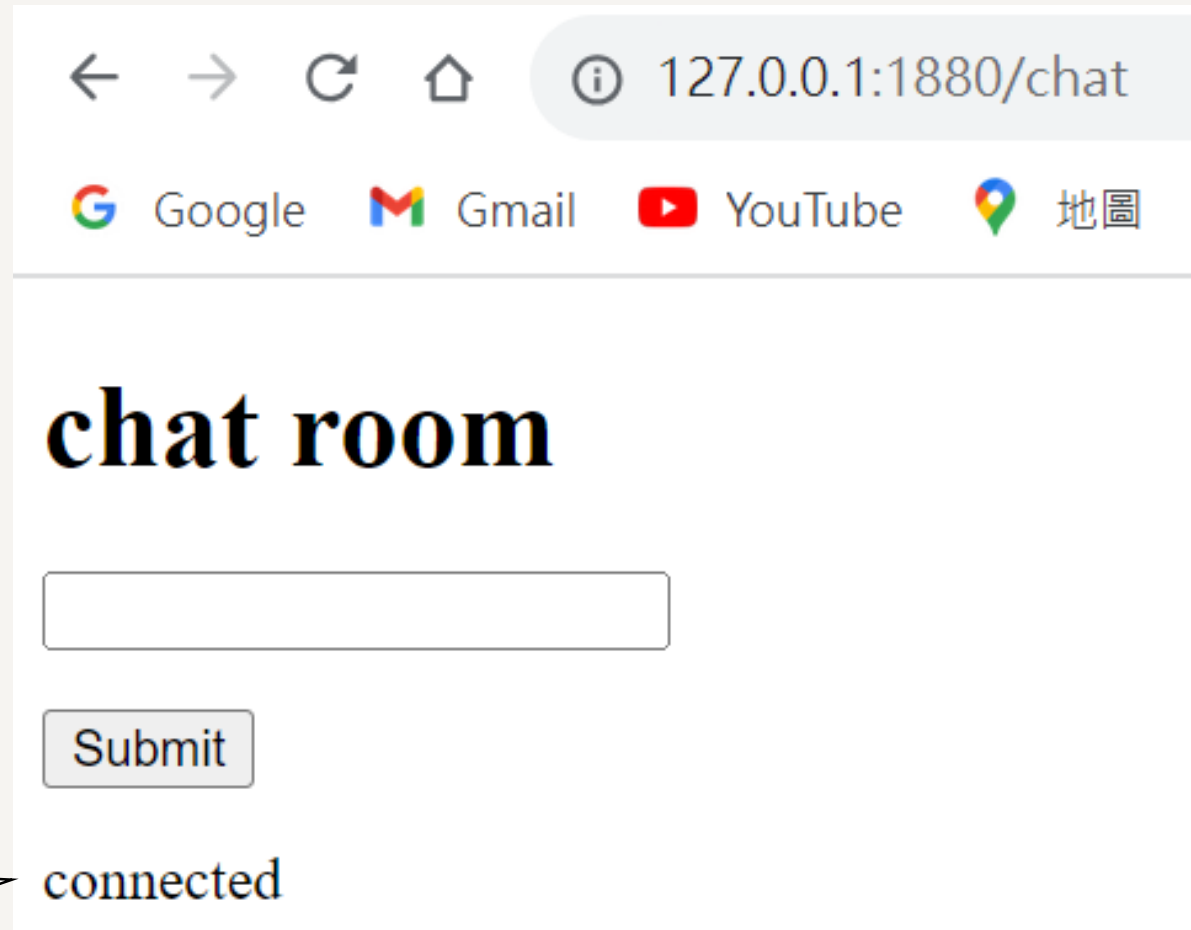
The screenshot shows the Node-RED interface with a flow containing three nodes: a green "[get] /chat" node, an orange "template" node, and a green "http" node. A large black arrow points from the "template" node to the "Edit template node" dialog box.

The "Edit template node" dialog box has a title bar with "Delete", "Cancel", and "Done" buttons. It has three tabs: "Properties", "Template", and "Syntax Highlight". The "Template" tab is active, showing a code editor with the following code:

```
9 var loc = window.location;
10 console.log(loc);
11 if (loc.protocol === "http:") { wsU
12
13
14 wsUri += "//" + loc.host + loc.path
15
16
```

The "Syntax Highlight" dropdown is set to "HTML". A callout box labeled "ex6-1-2.txt" points to the code editor. To the right of the dialog box, a large black arrow points to a red "Deploy" button.

Step 6: Refresh the chat room web page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:1880/chat". Below the address bar are search engines: Google, Gmail, YouTube, and 地圖. The main content area has the heading "chat room" in a large, bold, black serif font. Below the heading is a text input field. Underneath the input field is a button labeled "Submit". At the bottom of the page, the word "connected" is displayed in a blue, monospace-style font.

← → ↻ 🏠 ⓘ 127.0.0.1:1880/chat

🔍 Google 📧 Gmail 📺 YouTube 📍 地圖

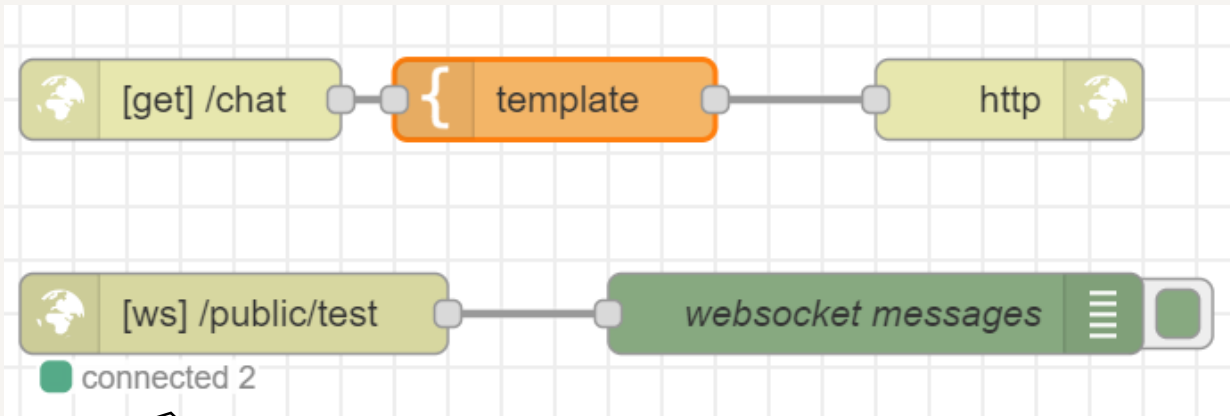
chat room

Submit

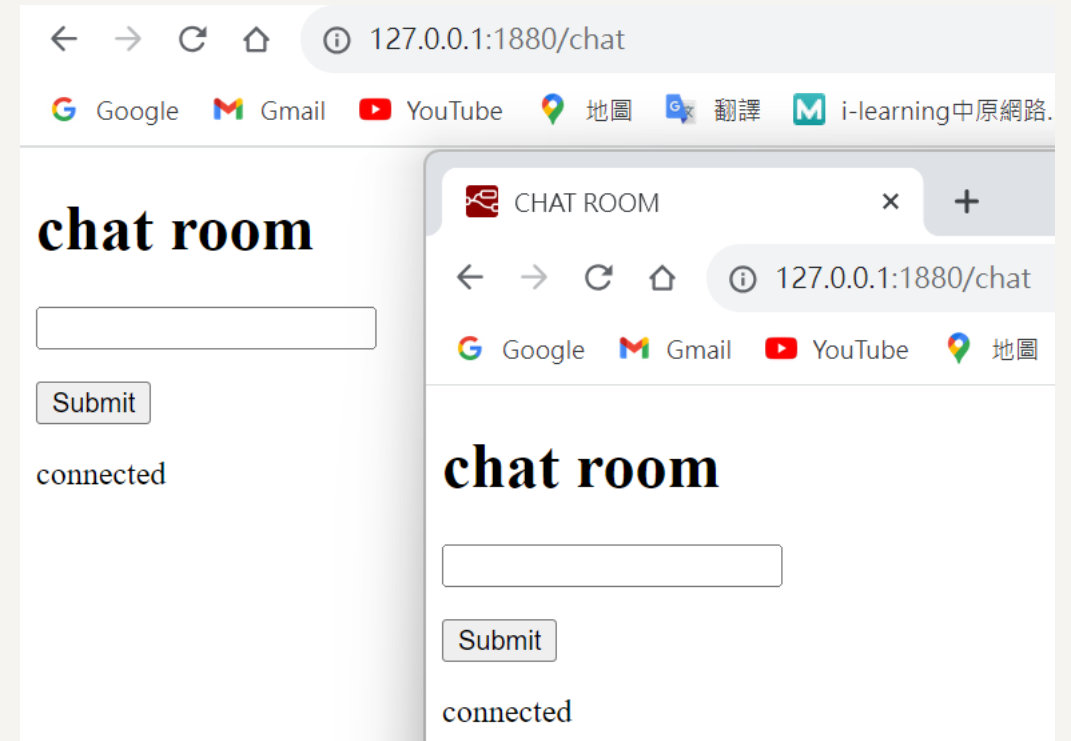
connected

connected

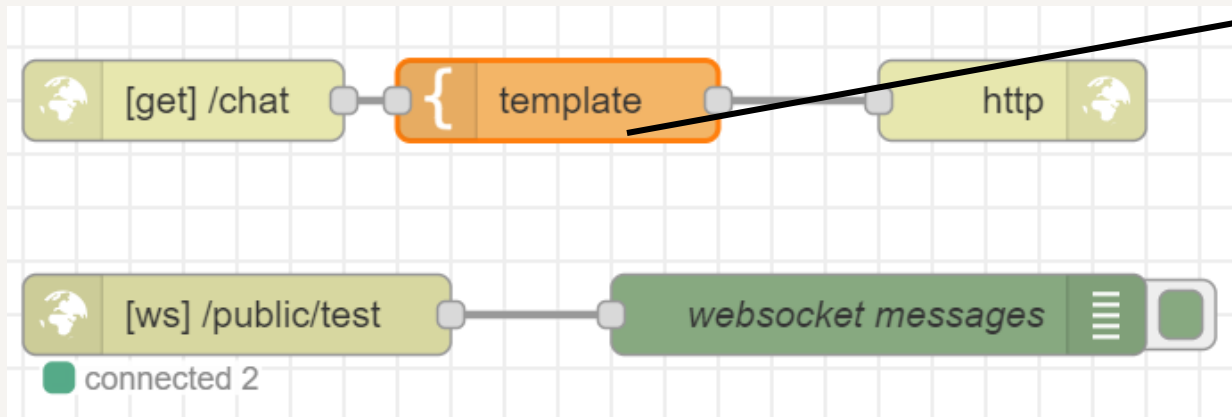
Step 7: Open the chat room web page in new window



connected2



Step 8: Modify HTML



Edit template node

Delete Cancel Done

Properties

Name:

Property: msg. payload

Template Syntax Highlight: HTML

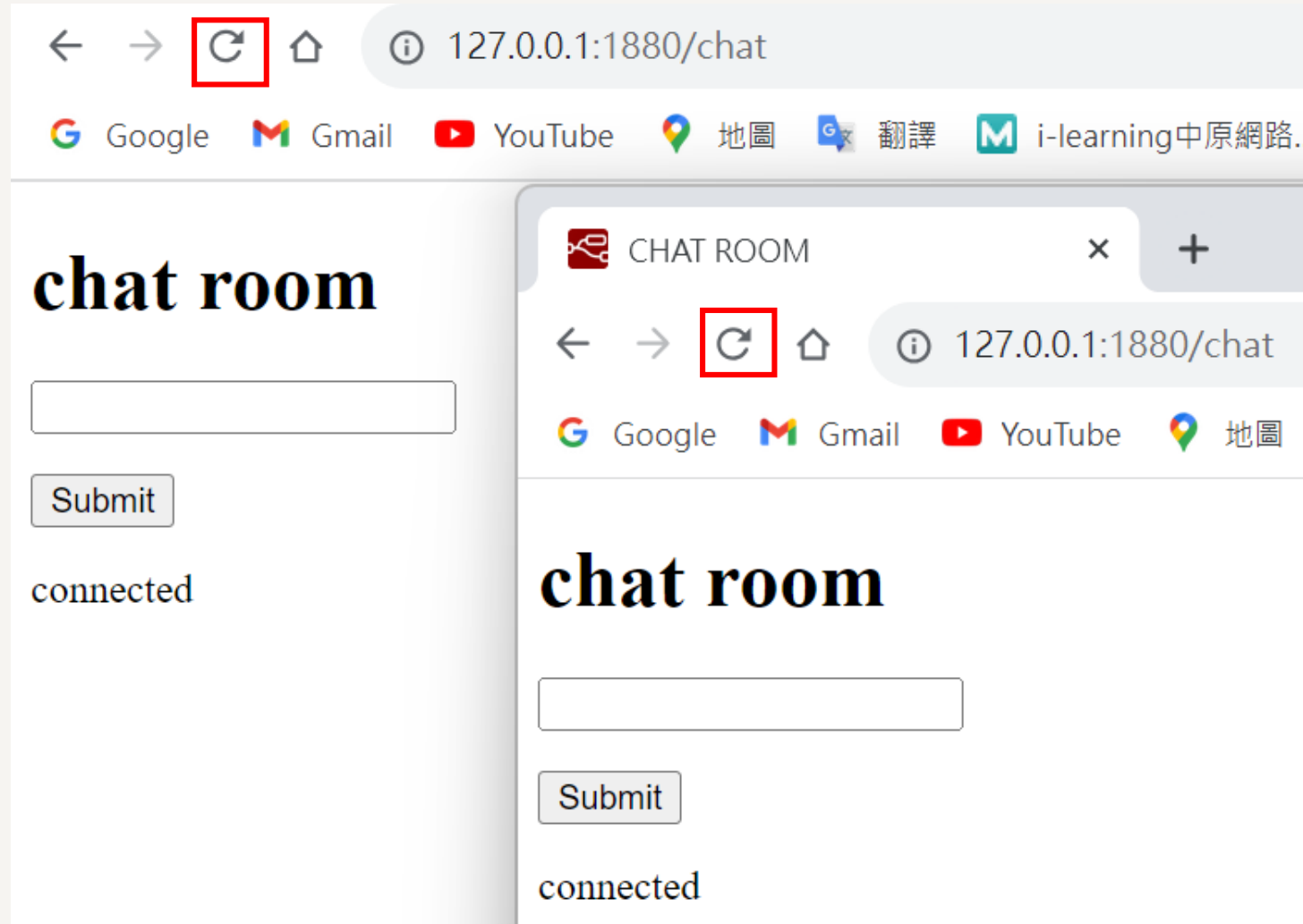
```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>CHAT ROOM</title>
5     <script type="text/javascript">
6       var ws;
7       var wsUri = "wss:";
8       var loc = window.location;
```

Format: Mustache template

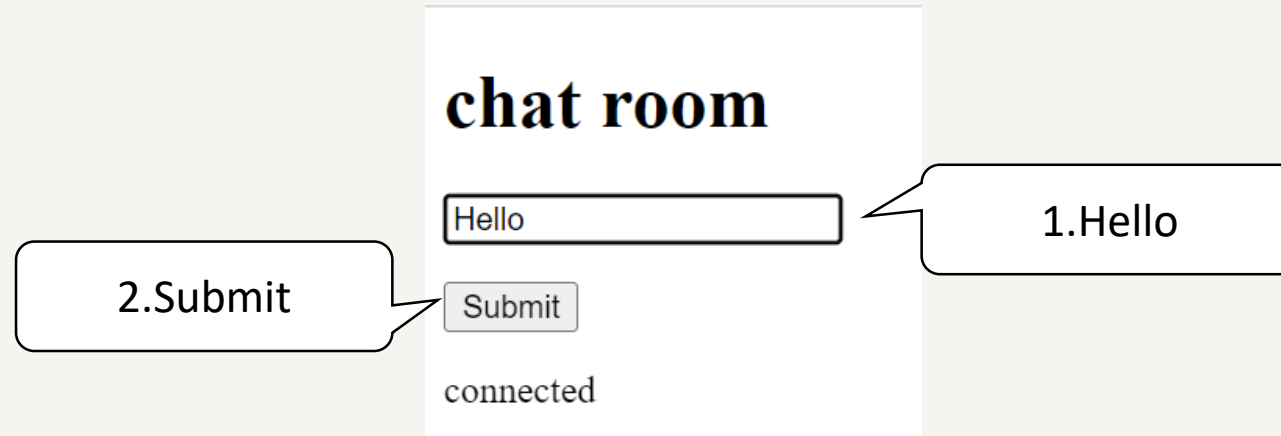
Output as: Plain text

ex6-1-3.txt

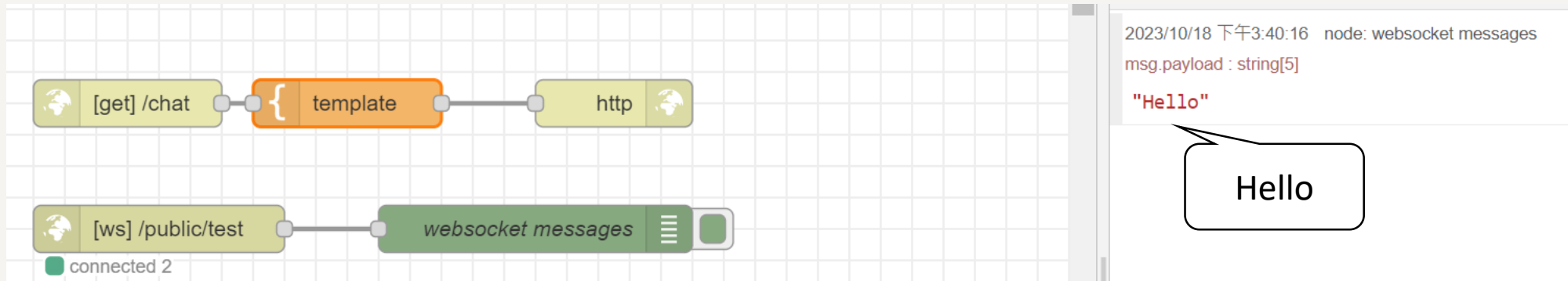
Step 9: Refresh chat room web pages



Step 10: Submit a message



Step 11: Check the debug window

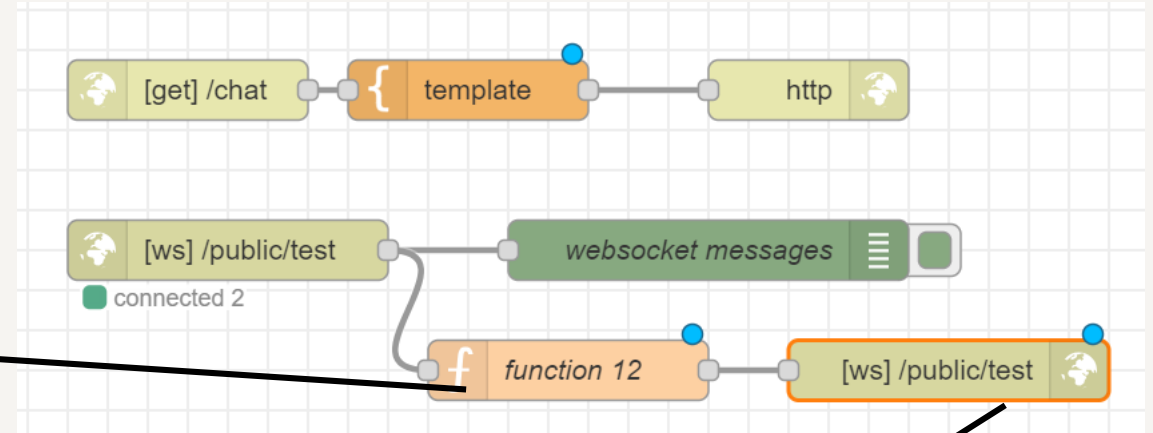


Step 12: Add a “function” node and a “websocket out” node

Name: function 12

Setup On Start On Message On Stop

```
1 var d = new Date();
2 var time= d.getHours()+":"+d.getMinutes()+":";
3 var chat=msg.payload;
4 msg={};
5 msg.payload=time+" "+chat;
6 return msg;
```



Edit websocket out node

Delete Cancel Done

Properties

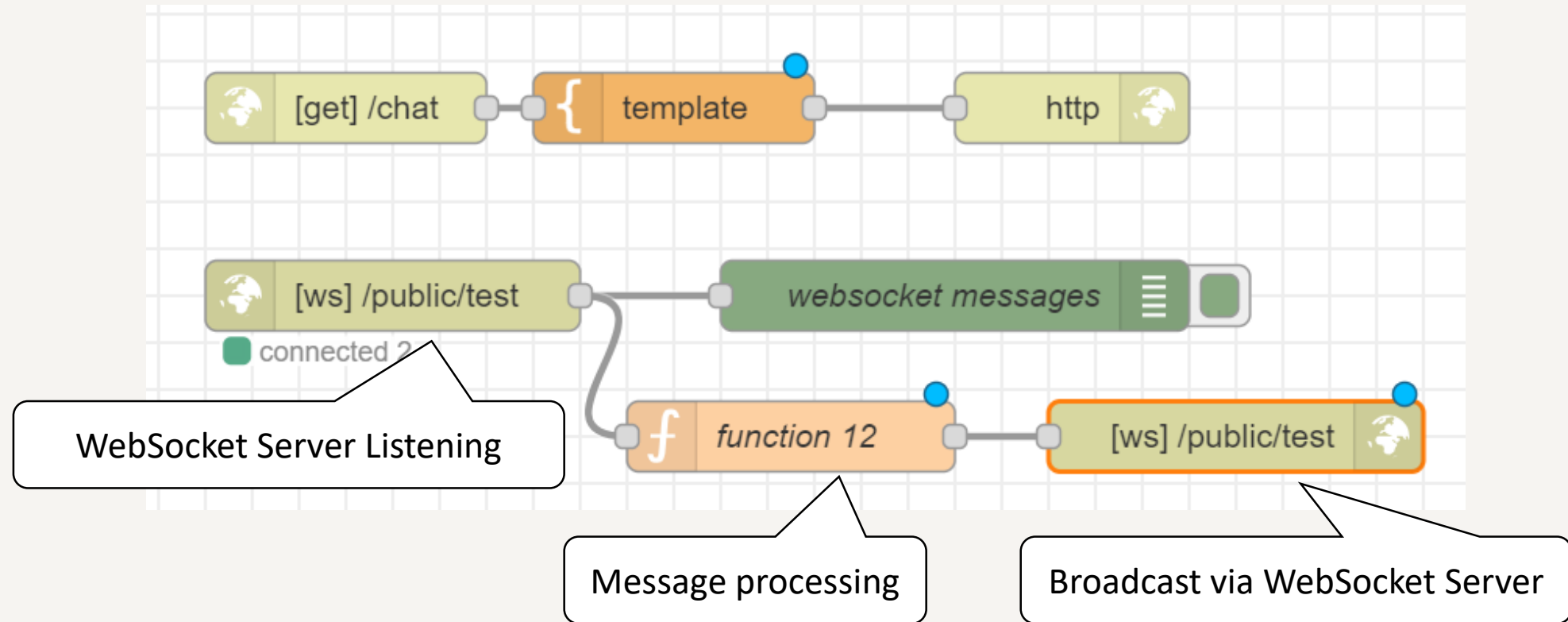
Type Listen on

Path /public/test

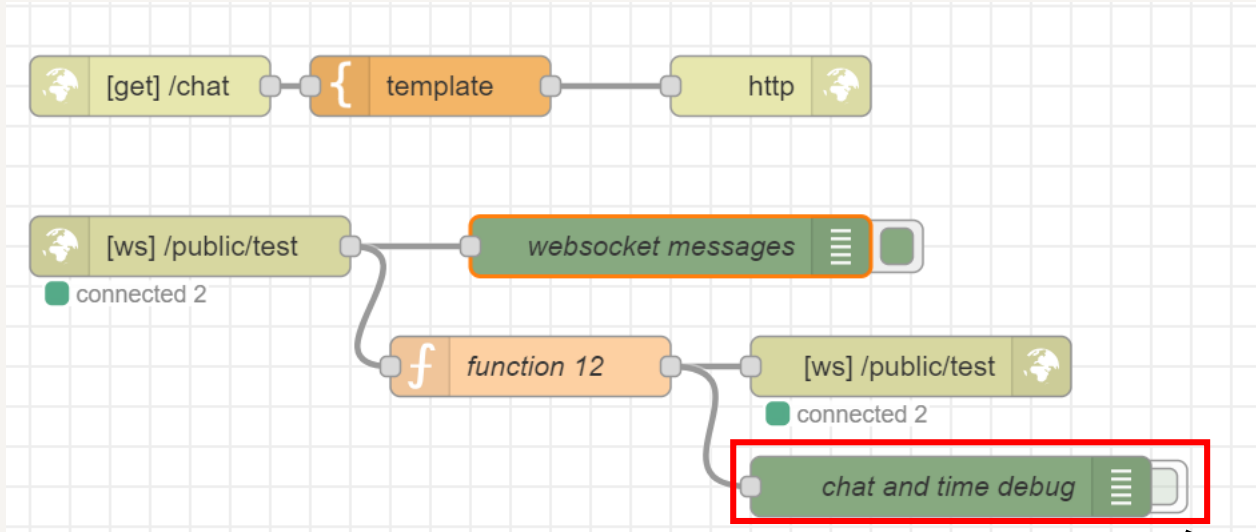
Name Name



Flow



Step 13: Add a debug node



chat room

connected

Edit debug node

Properties

Output

msg. payload

To

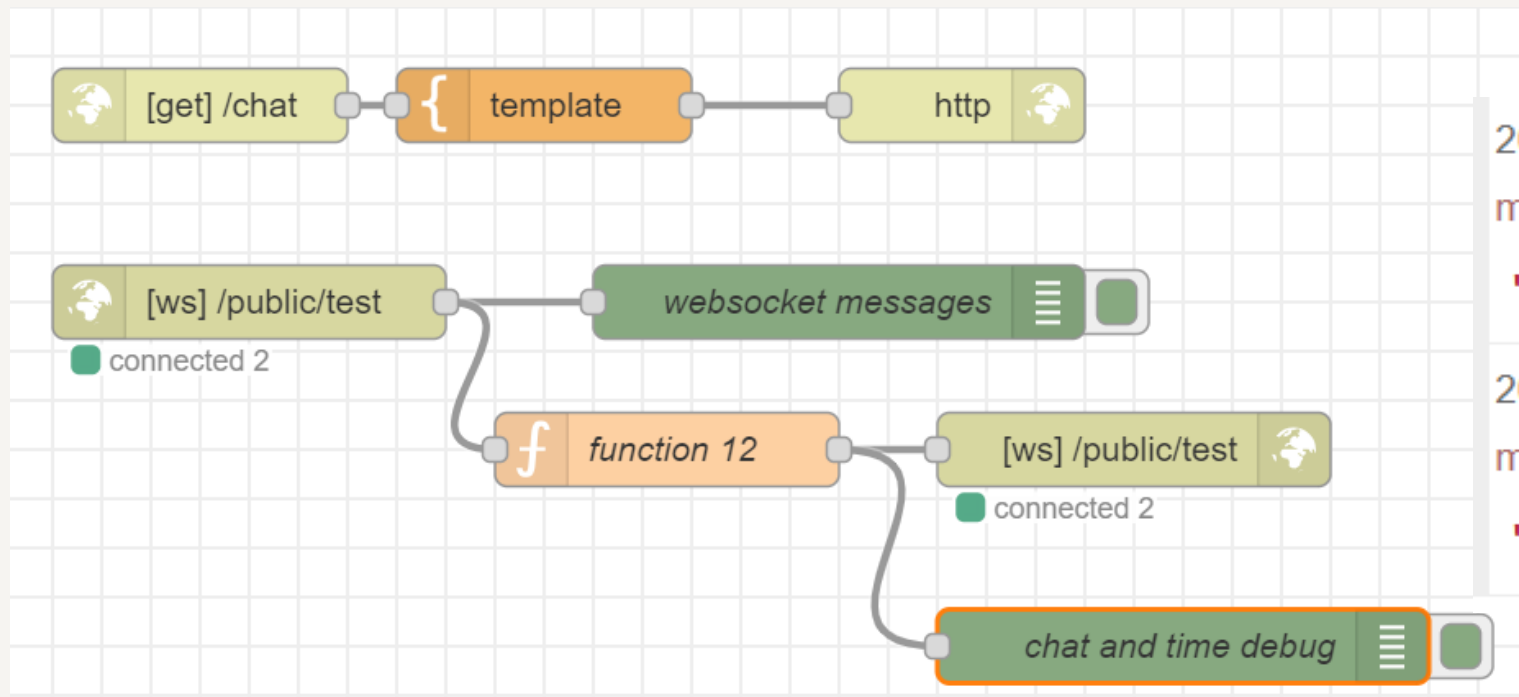
☒ debug window

☐ system console

☐ node status (32 characters)

Name

chat and time debug



2023/10/18 下午4:16:44 node: websocket messages

msg.payload : string[2]

"hi"

2023/10/18 下午4:16:44 node: chat and time debug

msg.payload : string[9]

"16:16: hi"

chat and time message

chat room

hi

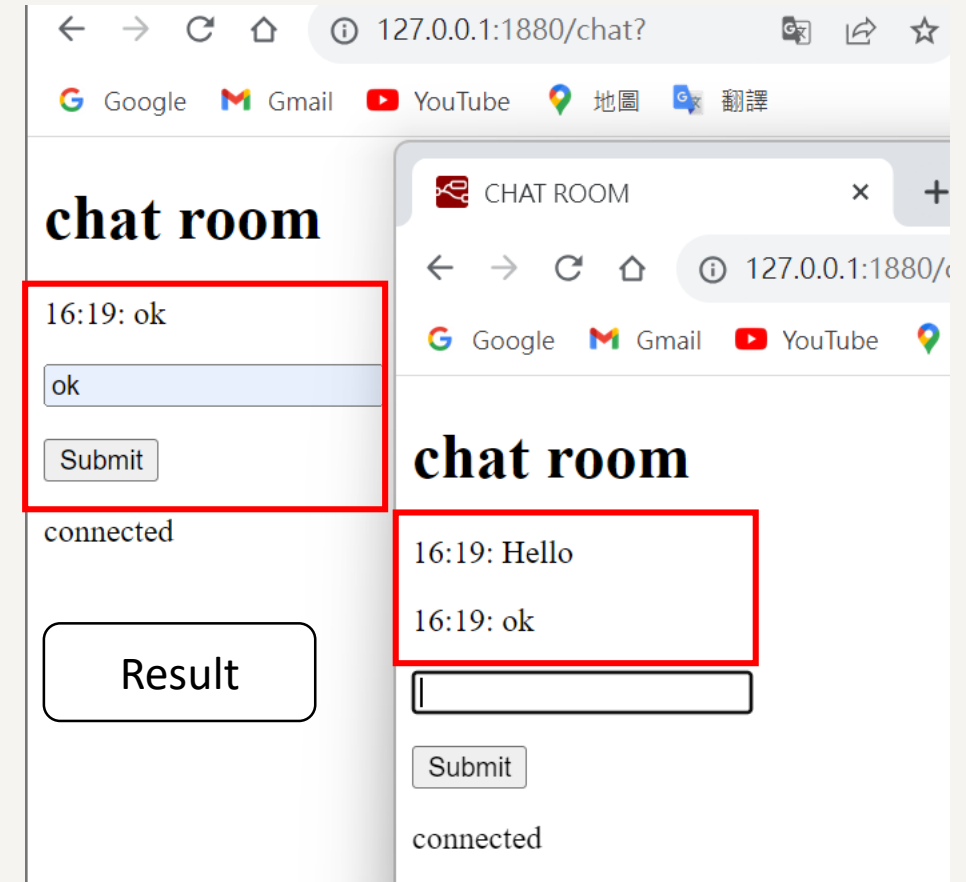
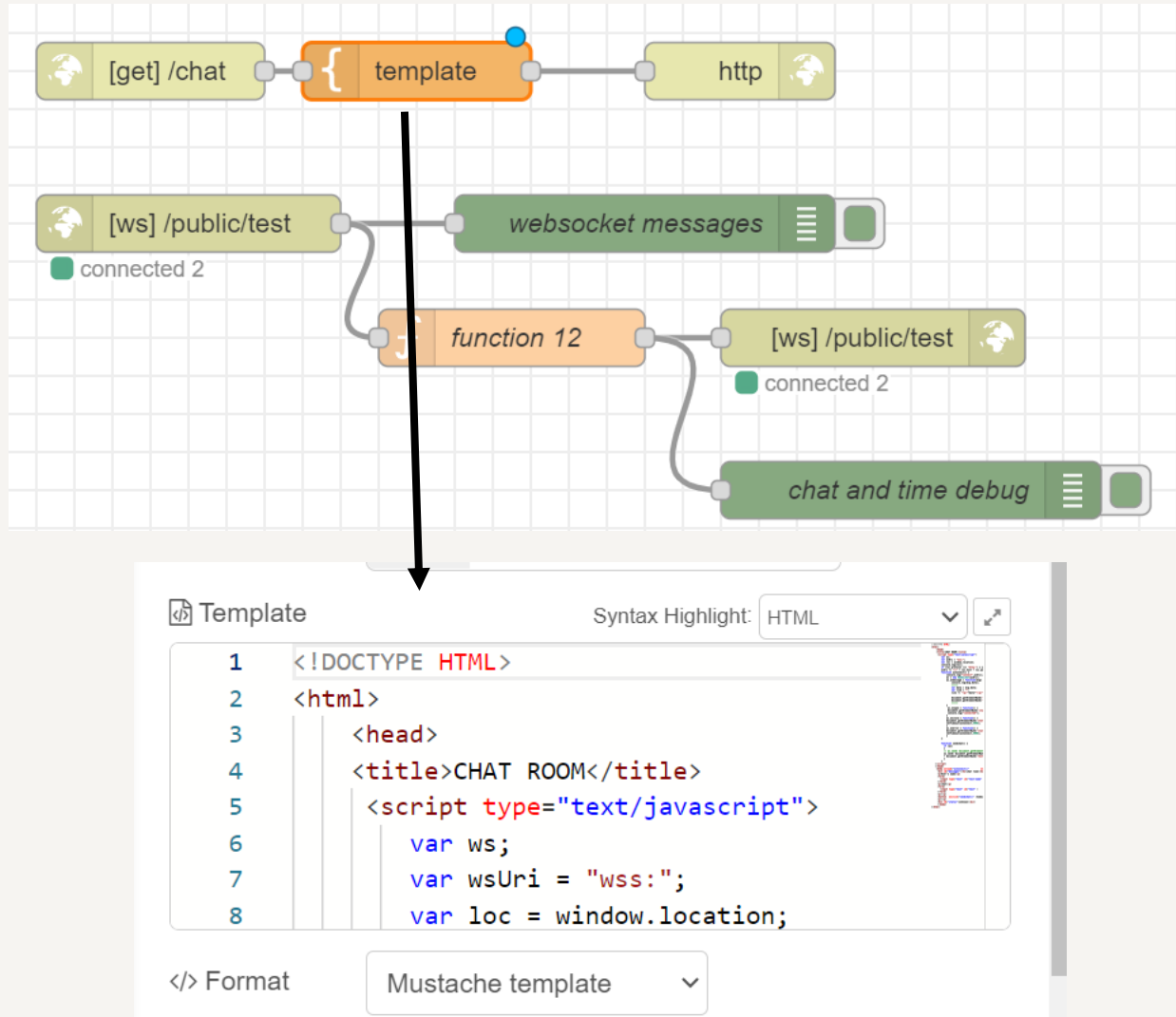
1.Hi

Submit

2.Submit

connected

Step 15: Modify HTML



Homework 6-1

- 請加上一個可以輸入用戶端代號的文字表單，讓聊天室網頁呈現進行聊天者的代號、聊天內容與聊天時間，如下圖所示。 Please add another input for user's name. When you receive or send a message, show the time, the user's name and the chat messag

chat room 18:27: Mary:Hello 18:28: Tom:Hi User's name <input type="text" value="Mary"/> chat <input type="text" value="Hello"/> <input type="button" value="Submit"/> connected	chat room 18:27: Mary:Hello 18:28: Tom:Hi User's name <input type="text" value="Tom"/> chat <input type="text" value="Hi"/> <input type="button" value="Submit"/> connected
--	--

修改程式碼

增加使用者名字顯示跟一個冒號將人名文字區隔開來

```
40 function sendchat()
41 {
42     if (ws)
43     {
44         // ws.send( document.getElementById('text').value);
45         ws.send( document.getElementById('text-name').value + ":"
46                 + document.getElementById('text').value);
47     }
48 }
```



```
48 </script>
49 </head>
50 <body onload="wsConnect()" onunload="ws.onclose()" >
51     <div id="messages"><h1>chat room</h1> </div>
52     <p>User's name</p>
53     <form>
54         <input type="text" id="text-name" >
55     </form>
56     <p>chat</p>
57     <form>
58         <input type="text" id="text" >
59     </form>
60     <p></p>
61     <button onclick="sendchat()" >Send</button>
62     <p></p>
63     <div id="status">unknown</div>
64 </body>
65 </html>
```

增加一個User's name跟chat的文字以及一個輸入框並給其一個id(text-name)用來記錄輸入的人名

Template

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>CHAT ROOM</title>
5     <script type="text/javascript">
6       var ws;
7       var wsUri = "wss:";
8       var loc = window.location;
```

Syntax Highlight: HTML

Format: Mustache template

Result

M 11 | M i-l | M i-l | M i-l | M

← → ↻ 🏠 ⓘ 127.0.0.1:18

Google Gmail YouTube

chat room

17:2: Anna:hi

17:2: Linda:hi

17:3: Anna:what do we eat?

17:3: Linda:chicker

17:3: Anna:only have one!

17:4: Linda:no~

User's name

Linda

chat

no~

Submit

CHAT ROOM

← → ↻ 🏠 ⓘ 127.0.0

Google Gmail YouT

chat room

17:2: Anna:hi

17:2: Linda:hi

17:3: Anna:what do we eat?

17:3: Linda:chicker

17:3: Anna:only have one!

17:4: Linda:no~

User's name

Anna

chat

only have one!

Submit

connected

Linda and Anna chat records