智慧整合感控系統概論
Introduction to Cyber-Physical Systems

LAB: NodeRED入門

國立臺北科技大學電子工程系
授課教師：李昭賢 副教授
電子郵件：chlee@ntut.edu.tw
校內分機：2288

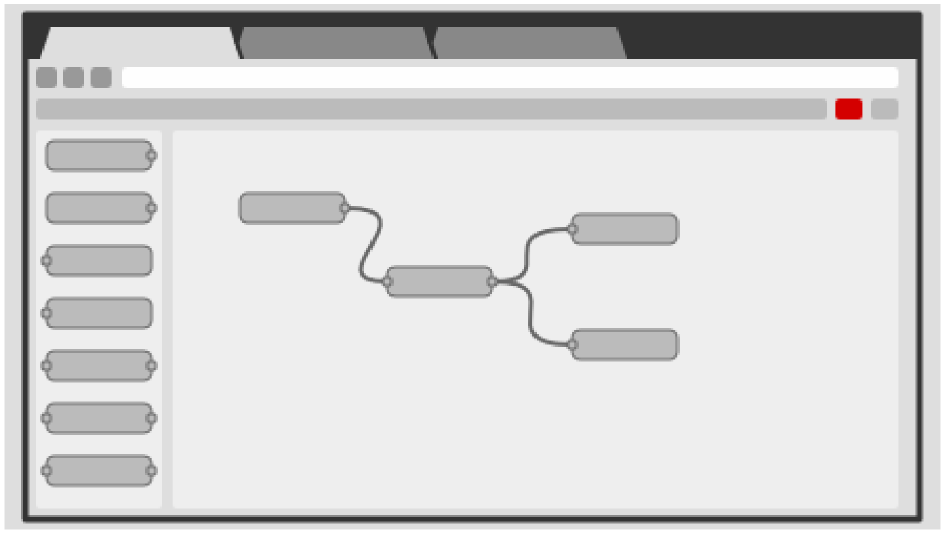
行動寬頻尖端技
術跨校教學聯盟

http://www.cc.ntut.edu.tw/~chlee/

學習目標

1 NodeRED

Node-RED (http://nodered.org)

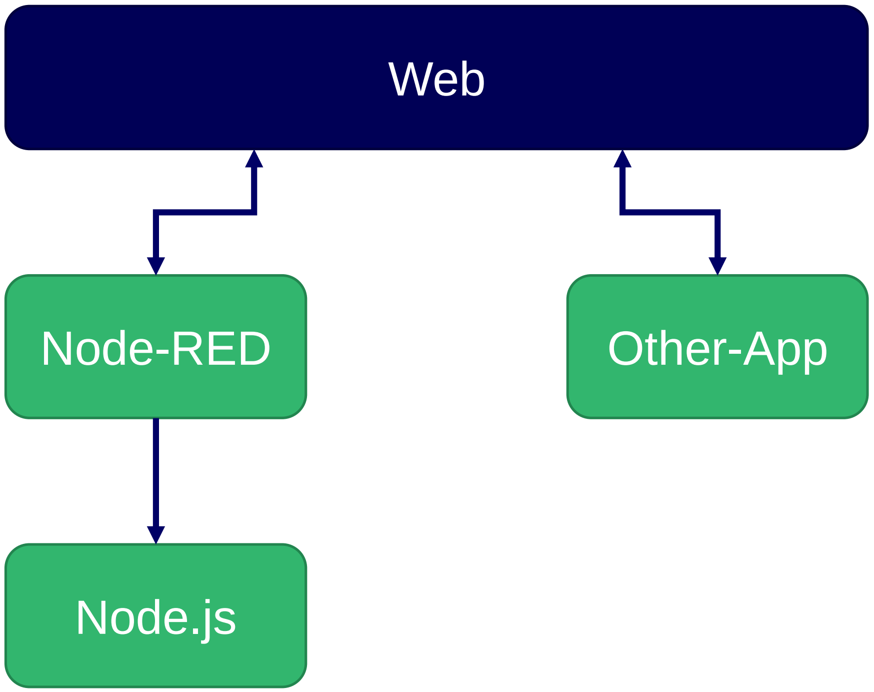


- 一個以 Node.js 為基礎的視覺化 IOT 開發工具
- 讓開發者可以更加容易使用HTTP、MQTT、Twitter進行物聯網應用程式設計

Node-RED支援平台

- 一般系統
  - Max OS X
  - Windows
  - Linux
- 移動式裝置
  - Raspberry Pi
  - BeagleBone Black
  - Interacting with Arduino
  - Android
- 雲端伺服器
  - IBM Bluemix
  - SenseTecnic FRED
  - Amazon Web Services
  - Microsoft Azure

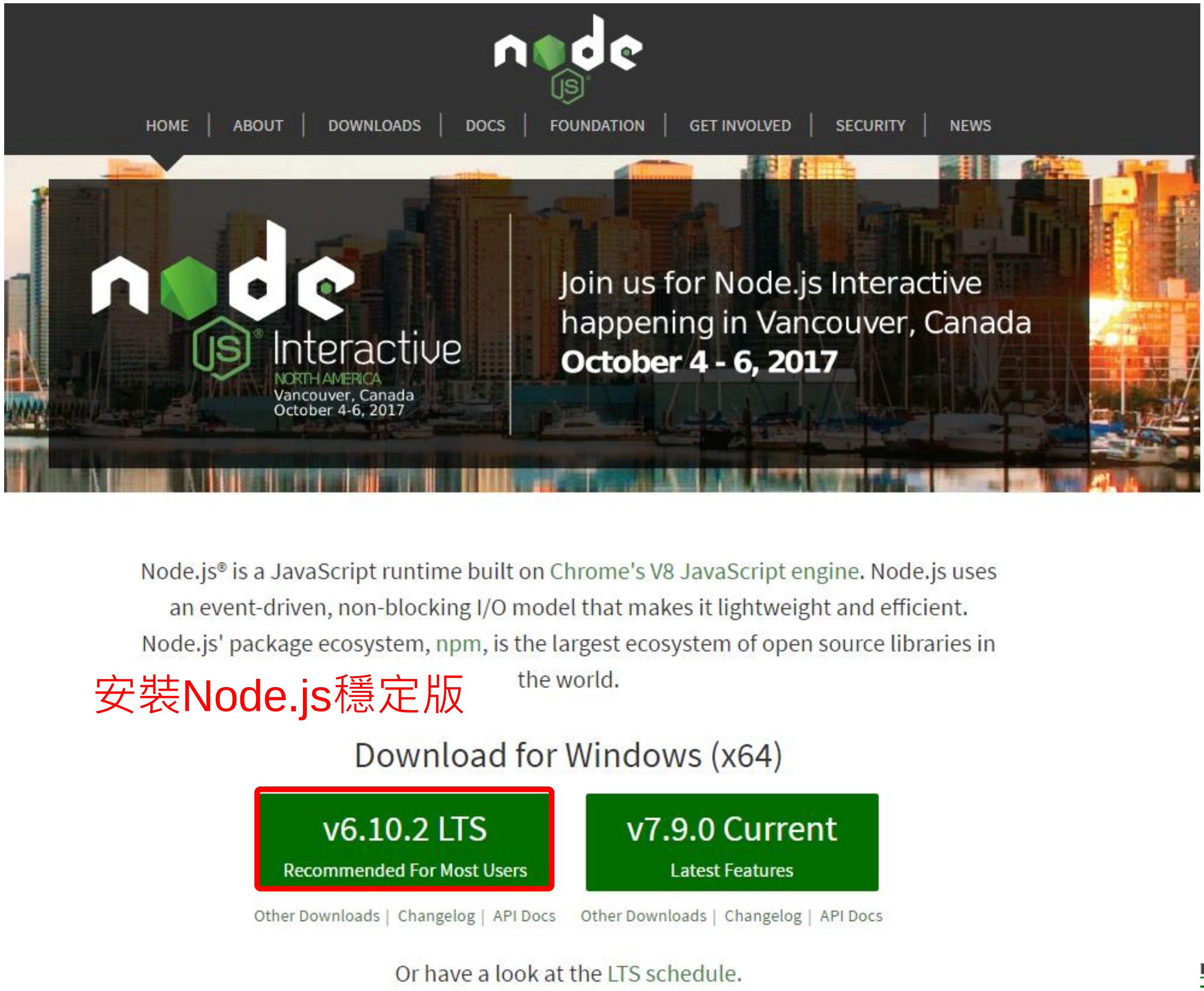
Node-RED架構



Node-RED安裝、執行

- 前往Node.js官網，安裝Node.js (LTS) version
  - https://nodejs.org/en/
- 利用npm安裝Node-RED套件
  - sudo npm install -g --unsafe-perm node-red
  - https://nodered.org/
- 執行Node-RED
  - node-red –p portNum
- 啟動瀏覽器，訪問http://localhost:portNum

Node-RED - Node.js安裝



HOME | ABOUT | DOWNLOADS | DOCS | FOUNDATION | GET INVOLVED | SECURITY | NEWS
node js Interactive
NORTH AMERICA
Vancouver, Canada
October 4-6, 2017

Join us for Node.js Interactive happening in Vancouver, Canada **October 4 - 6, 2017**

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

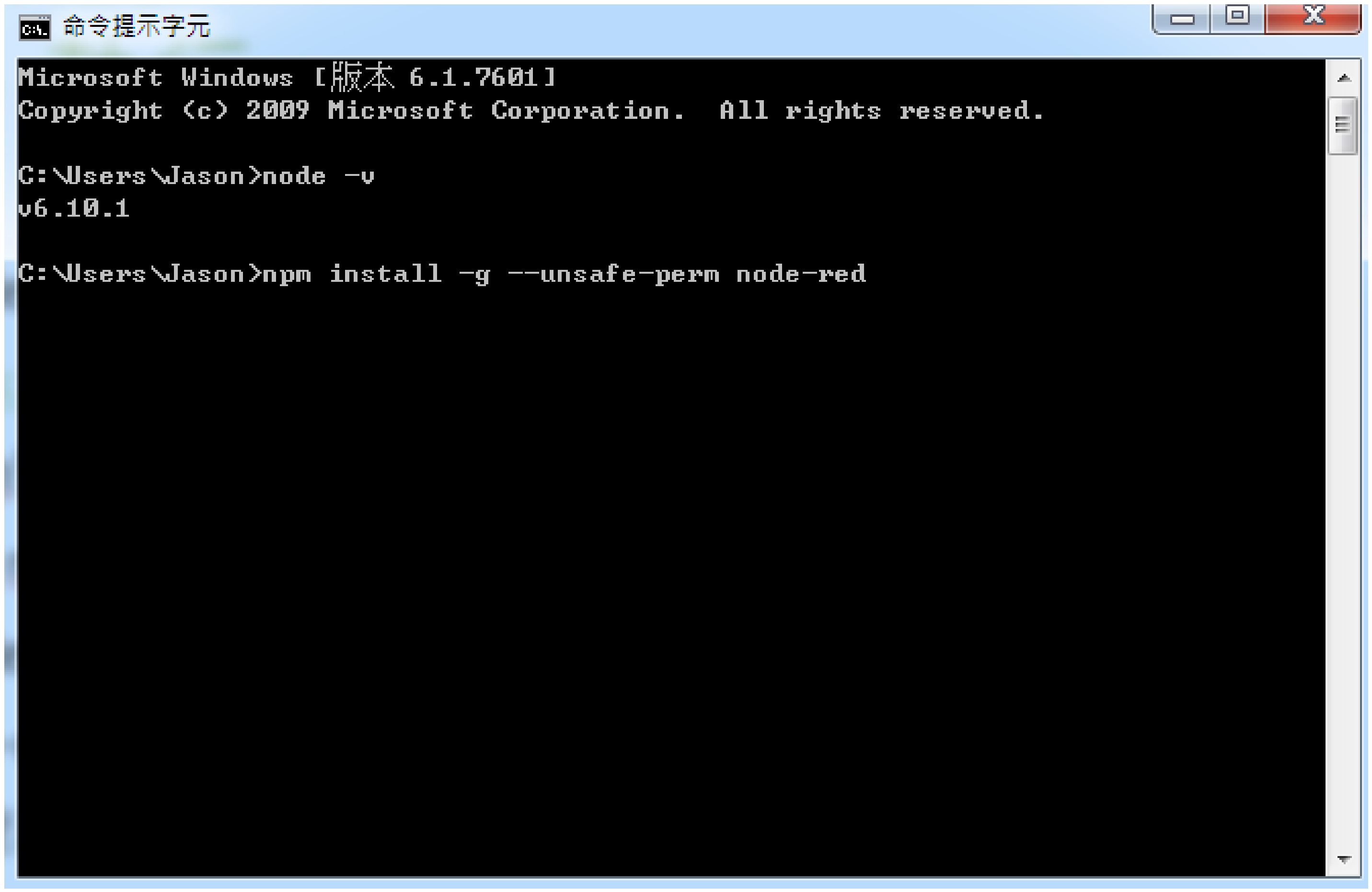
安裝Node.js穩定版

Download for Windows (x64)

v6.10.2 LTS
Recommended For Most Users
v7.9.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

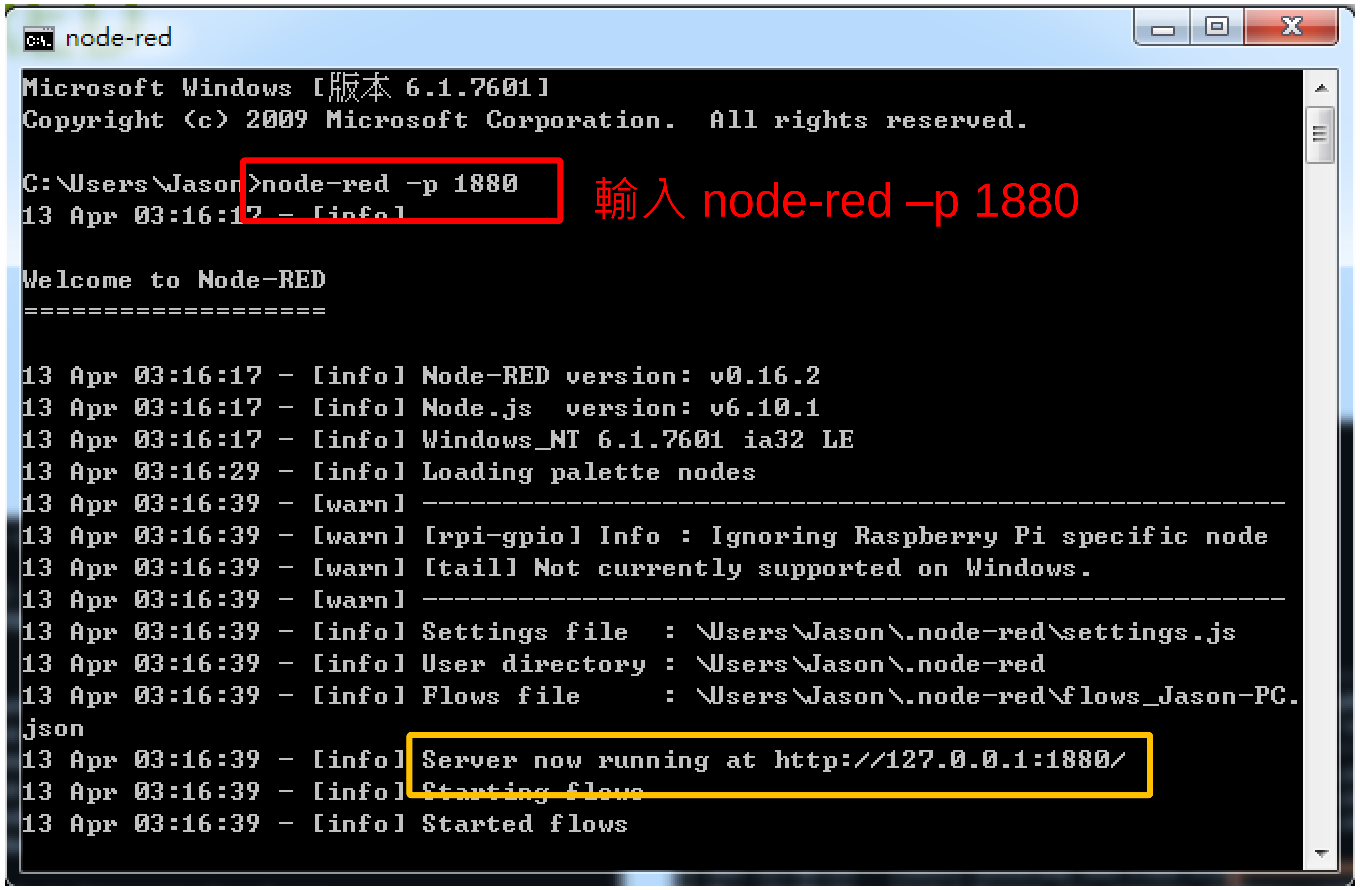
Or have a look at the LTS schedule.

Node-RED - Node-RED安裝



```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jason>node -v
v6.10.1

C:\Users\Jason>npm install -g --unsafe-perm node-red
```

Node-RED執行

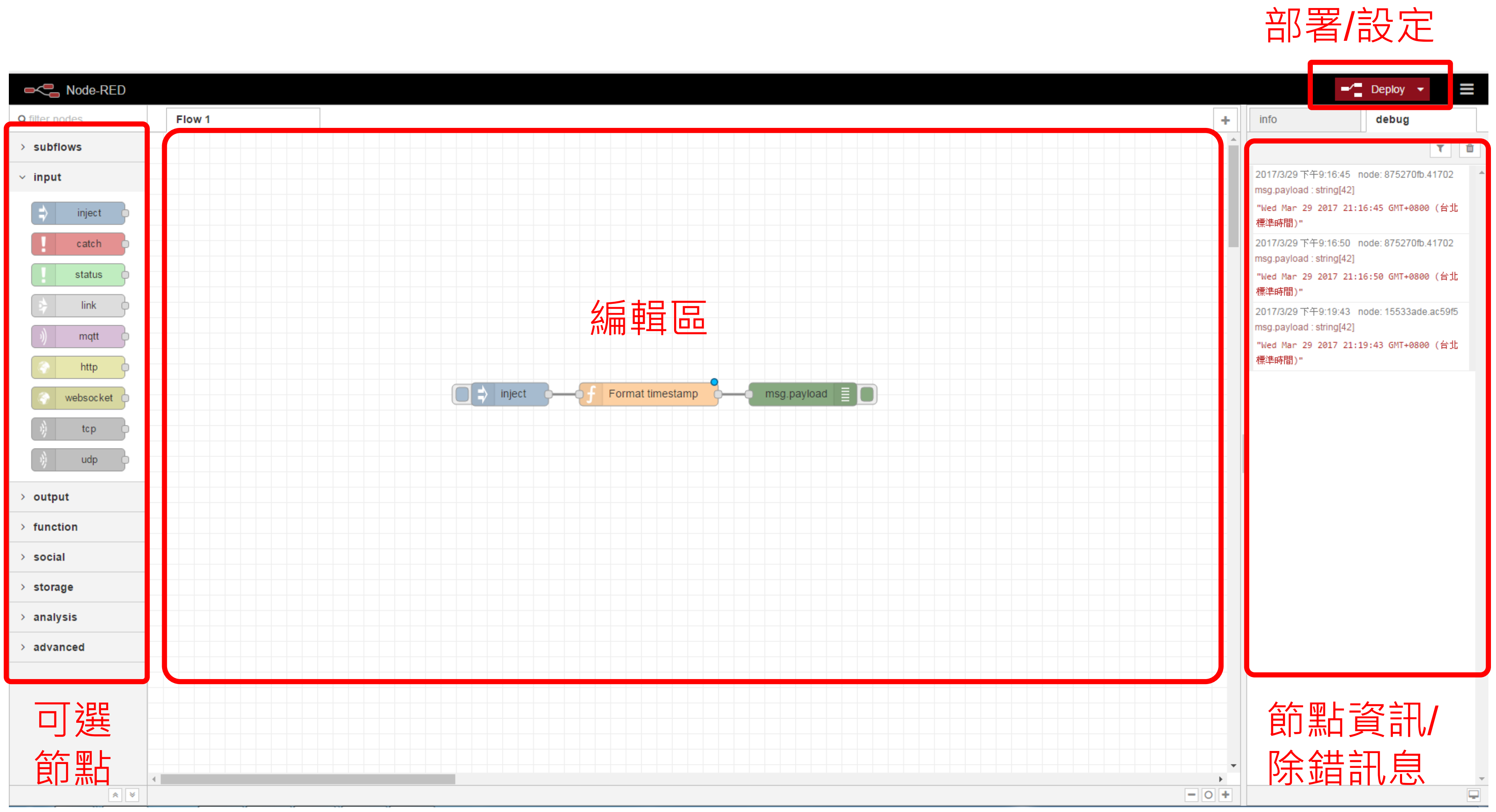


```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jason>node-red -p 1880
13 Apr 03:16:17 - [info]

Welcome to Node-RED
===================

13 Apr 03:16:17 - [info] Node-RED version: v0.16.2
13 Apr 03:16:17 - [info] Node.js  version: v6.10.1
13 Apr 03:16:17 - [info] Windows_NT 6.1.7601 ia32 LE
13 Apr 03:16:29 - [info] Loading palette nodes
13 Apr 03:16:39 - [warn] ------------------------------------------------------
13 Apr 03:16:39 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
13 Apr 03:16:39 - [warn] [tail] Not currently supported on Windows.
13 Apr 03:16:39 - [warn] ------------------------------------------------------
13 Apr 03:16:39 - [info] Settings file  : \Users\Jason\.node-red\settings.js
13 Apr 03:16:39 - [info] User directory : \Users\Jason\.node-red
13 Apr 03:16:39 - [info] Flows file     : \Users\Jason\.node-red\flows_Jason-PC.
json
13 Apr 03:16:39 - [info] Server now running at http://127.0.0.1:1880/
13 Apr 03:16:39 - [info] Starting flows
13 Apr 03:16:39 - [info] Started flows
```

Node-RED (Checking Point 1)

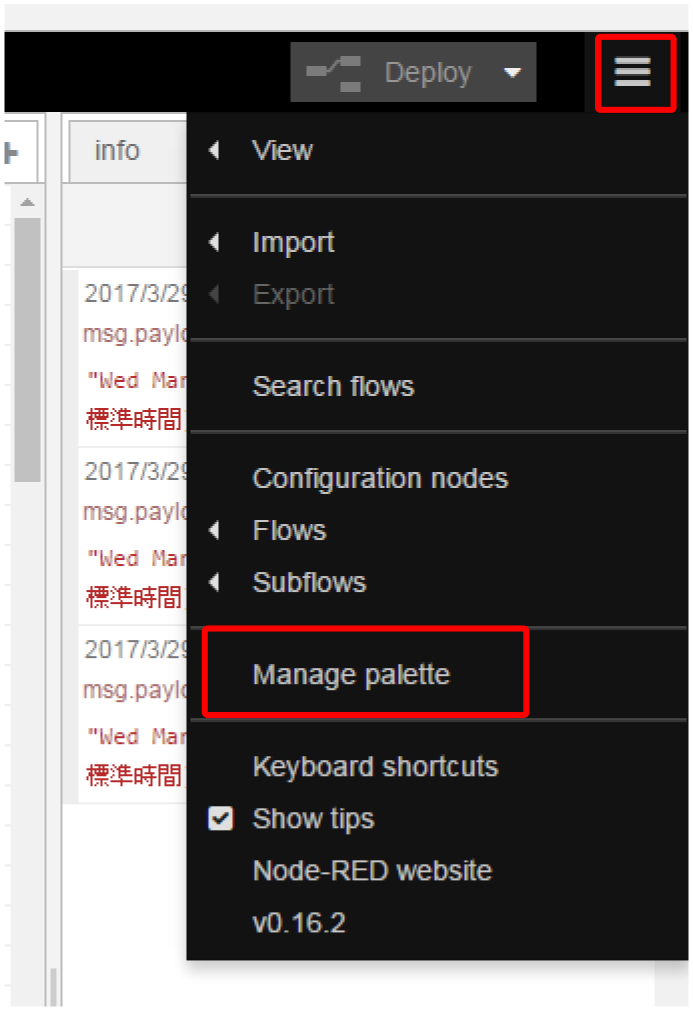
- 完成Node-RED安裝

Node-RED介面



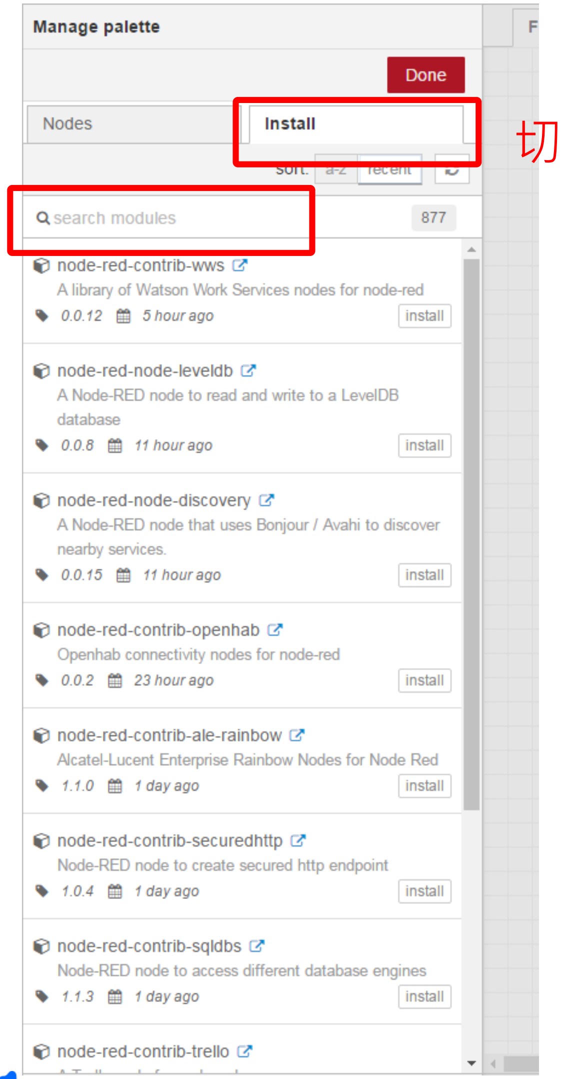
```
2017/3/29 下午9:16:45  node: 875270fb.41702
msg.payload : string[42]
"Wed Mar 29 2017 21:16:45 GMT+0800 (台北標準時間)"
2017/3/29 下午9:16:50  node: 875270fb.41702
msg.payload : string[42]
"Wed Mar 29 2017 21:16:50 GMT+0800 (台北標準時間)"
2017/3/29 下午9:19:43  node: 15533ade.ac59f5
msg.payload : string[42]
"Wed Mar 29 2017 21:19:43 GMT+0800 (台北標準時間)"
```

Node-RED安裝套件方法



設定

Node-RED安裝套件方法



node-red-contrib-wws
A library of Watson Work Services nodes for node-red
0.0.12 5 hour ago install

node-red-node-leveldb
A Node-RED node to read and write to a LevelDB database
0.0.8 11 hour ago install

node-red-node-discovery
A Node-RED node that uses Bonjour / Avahi to discover nearby services.
0.0.15 11 hour ago install

node-red-contrib-openhab
Openhab connectivity nodes for node-red
0.0.2 23 hour ago install

node-red-contrib-ale-rainbow
Alcatel-Lucent Enterprise Rainbow Nodes for Node Red
1.1.0 1 day ago install

node-red-contrib-securedhttp
Node-RED node to create secured http endpoint
1.0.4 1 day ago install

node-red-contrib-sqldbs
Node-RED node to access different database engines
1.1.3 1 day ago install

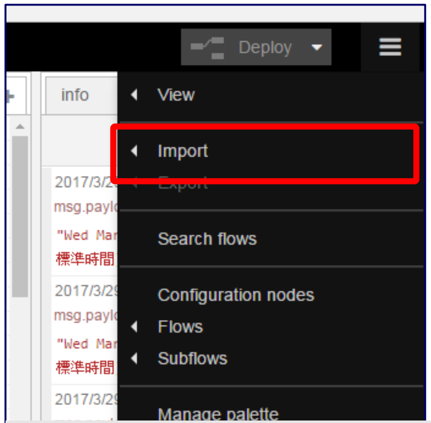
node-red-contrib-trello

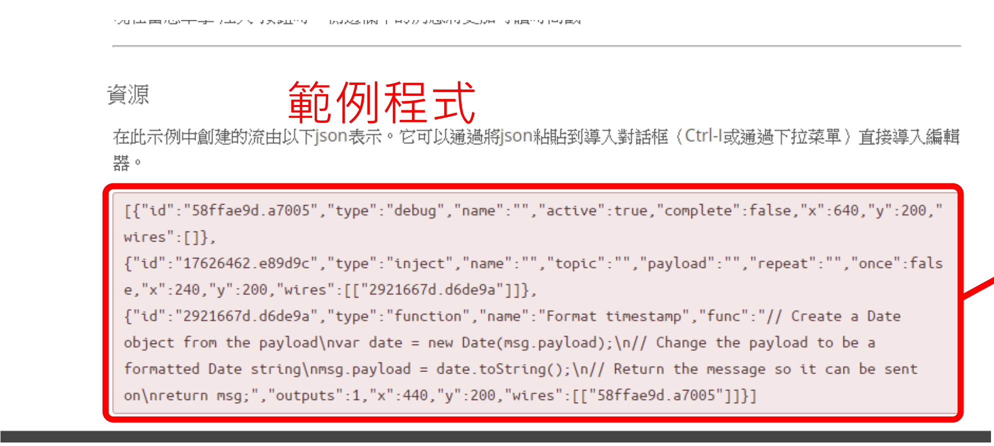
切換至安裝頁面

輸入套件名稱

Node-RED Import

Node-Red官網上有一些簡易的範例程式，可利用Import匯入專案中執行

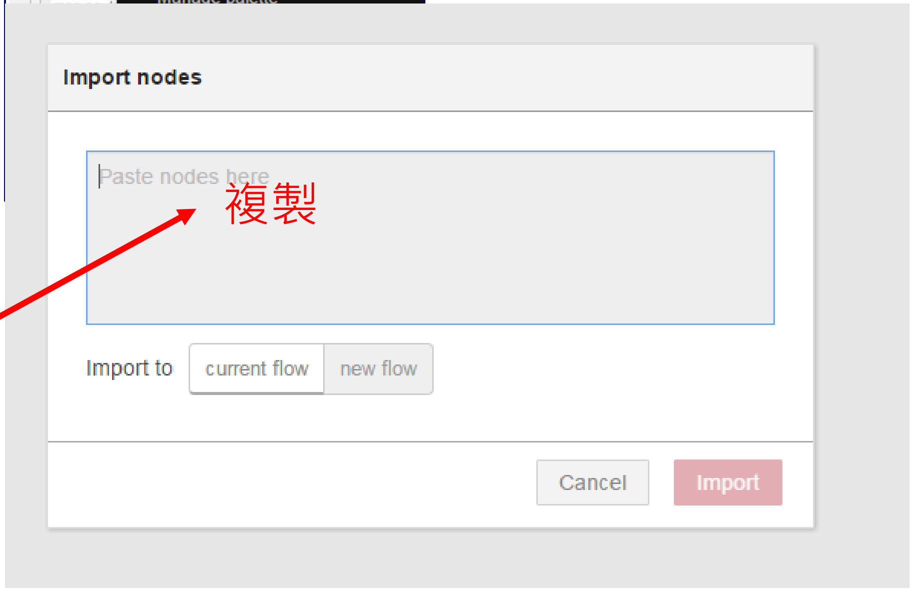



資源

範例程式

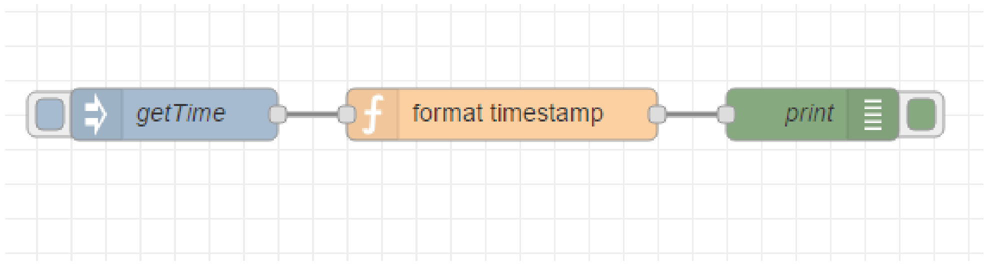
在此示例中創建的流由以下json表示。它可以通過將json粘貼到導入對話框（Ctrl-I或通過下拉菜單）直接導入編輯器。

```
[{"id":"58ffae9d.a7005","type":"debug","name":"","active":true,"complete":false,"x":640,"y":200,"
wires":[]},
{"id":"17626462.e89d9c","type":"inject","name":"","topic":"","payload":"","repeat":"","once":fals
e,"x":240,"y":200,"wires":[["2921667d.d6de9a"]]},
{"id":"2921667d.d6de9a","type":"function","name":"Format timestamp","func":"// Create a Date
object from the payload\nvar date = new Date(msg.payload);\n// Change the payload to be a
formatted Date string\nmsg.payload = date.toString();\n// Return the message so it can be sent
on\nreturn msg;","outputs":1,"x":440,"y":200,"wires":[["58ffae9d.a7005"]]}]
```



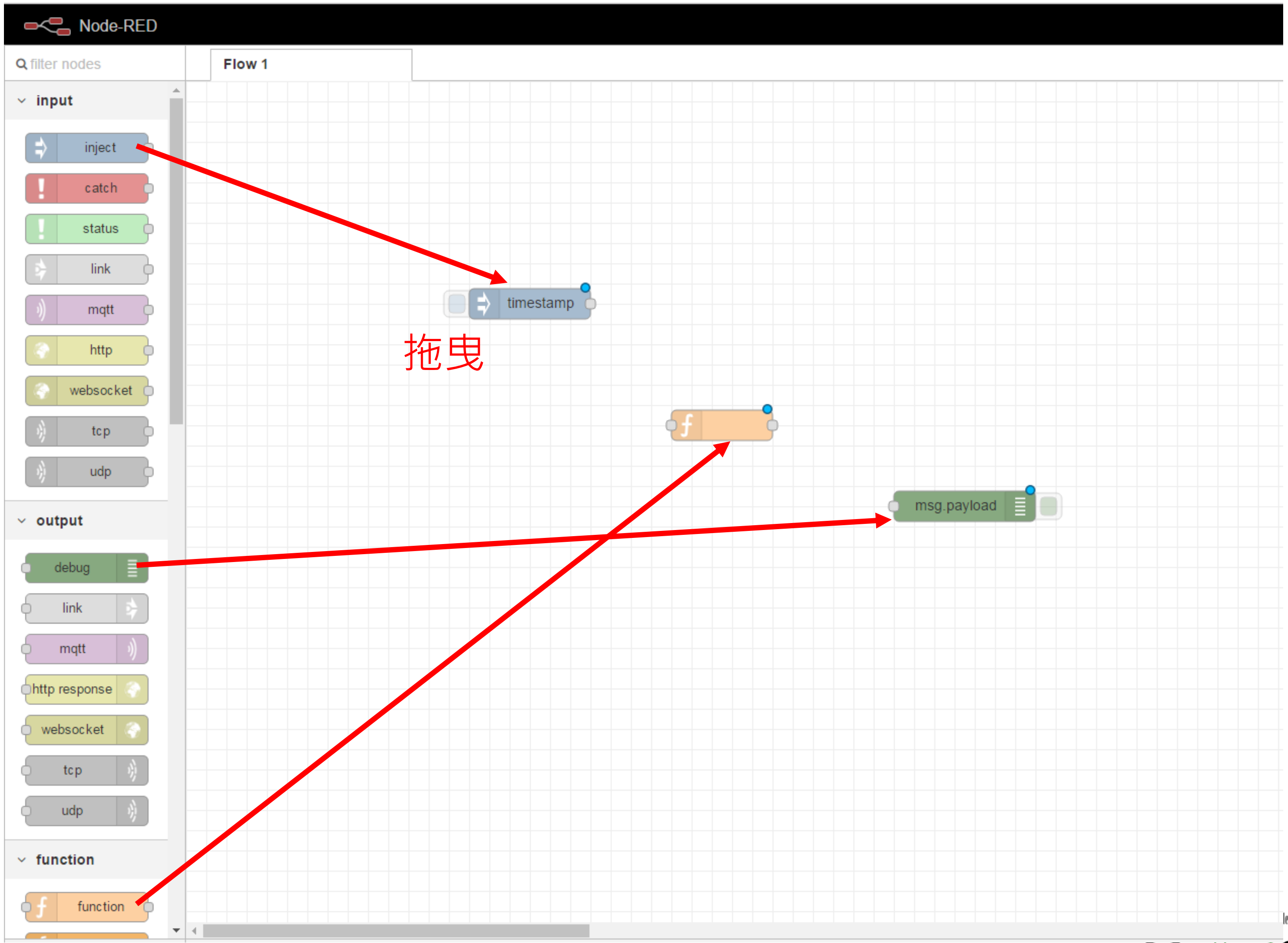
行動寬頻尖端技術跨校教學聯盟

Node-RED 範例1: inject、function、debug

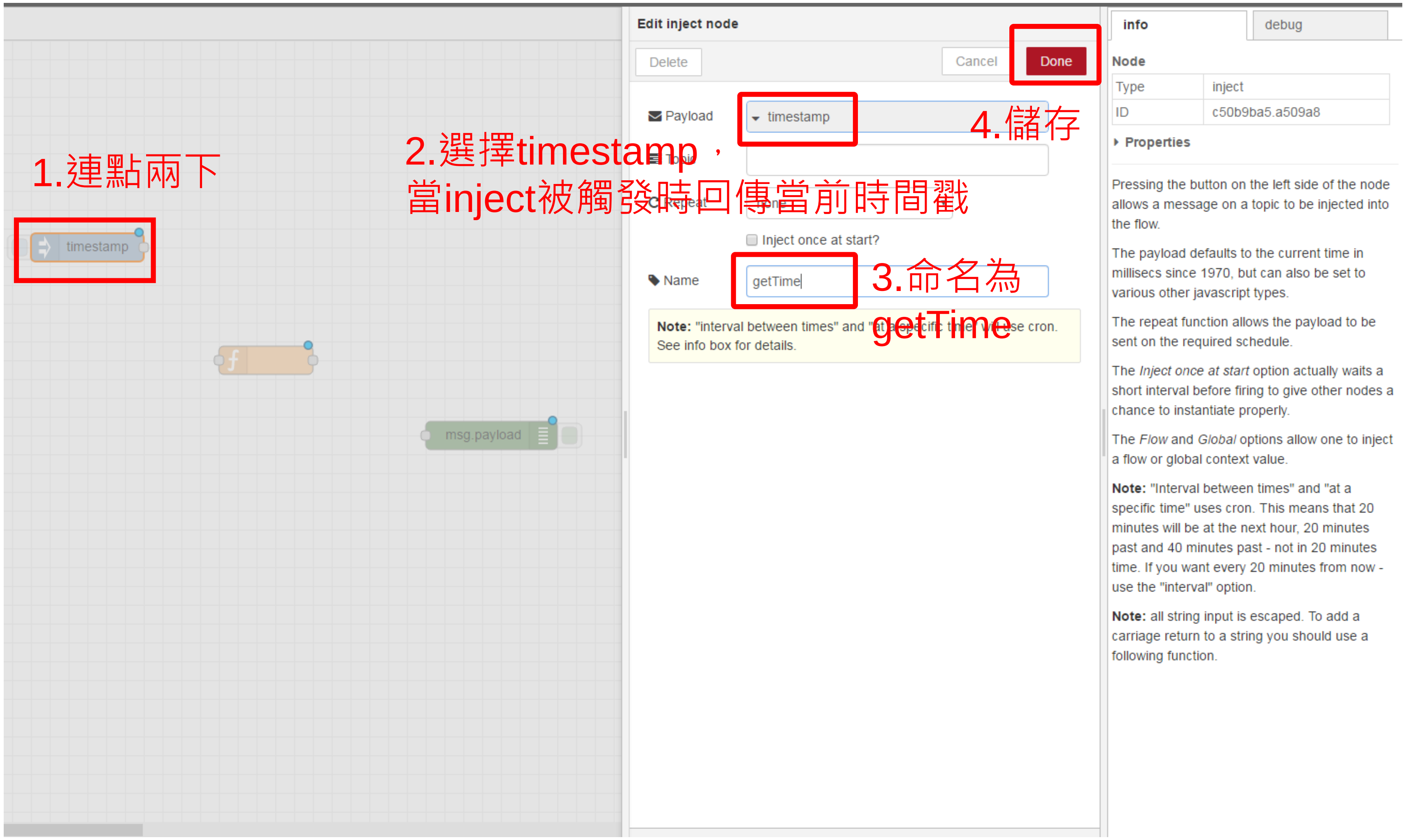


- 範例
  - 編寫一個顯示當前時間的Node-RED程式
- 功能
  - 當觸發inject時
  - 將觸發時間轉換成標準日期和時間格式
  - 進行紀錄

Node-RED 範例1: 新增節點



Node-RED 範例1: inject



1.連點兩下

2.選擇timestamp，當inject被觸發時回傳當前時間戳

3.命名為getTime

4.儲存

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

Node

Type	inject
ID	c50b9ba5.a509a8

▸ **Properties**

Pressing the button on the left side of the node allows a message on a topic to be injected into the flow.

The payload defaults to the current time in millisecs since 1970, but can also be set to various other javascript types.

The repeat function allows the payload to be sent on the required schedule.

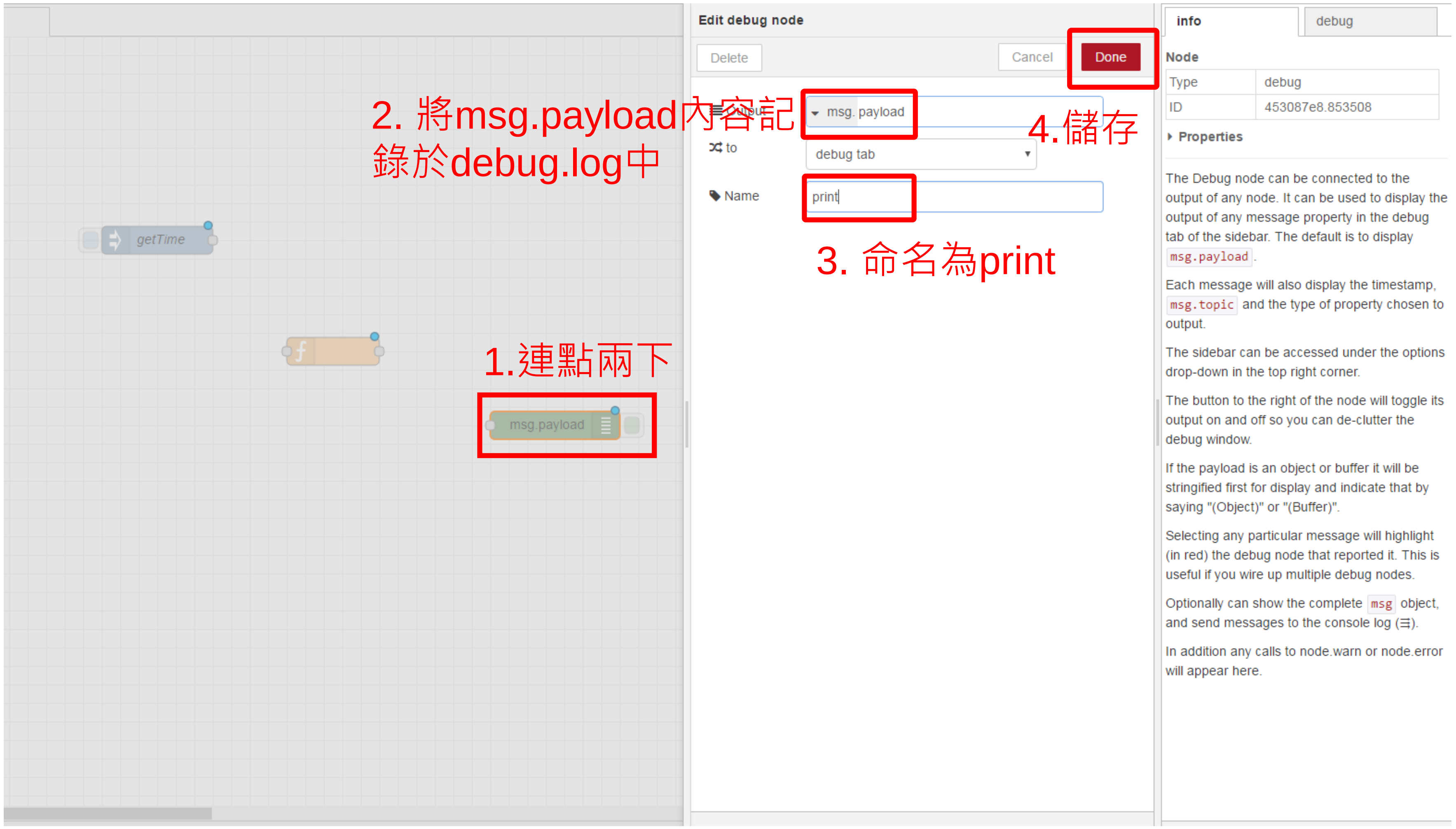
The *Inject once at start* option actually waits a short interval before firing to give other nodes a chance to instantiate properly.

The *Flow* and *Global* options allow one to inject a flow or global context value.

Note: "Interval between times" and "at a specific time" uses cron. This means that 20 minutes will be at the next hour, 20 minutes past and 40 minutes past - not in 20 minutes time. If you want every 20 minutes from now - use the "interval" option.

Note: all string input is escaped. To add a carriage return to a string you should use a following function.

Node-RED 範例1: debug



2. 將msg.payload內容記錄於debug.log中

4.儲存

3. 命名為print

1.連點兩下

Edit debug node

Delete | Cancel | Done

Output: msg. payload

to: debug tab

Name: print

info | debug

Node

Type	debug
ID	453087e8.853508

Properties

The Debug node can be connected to the output of any node. It can be used to display the output of any message property in the debug tab of the sidebar. The default is to display `msg.payload`.

Each message will also display the timestamp, `msg.topic` and the type of property chosen to output.

The sidebar can be accessed under the options drop-down in the top right corner.

The button to the right of the node will toggle its output on and off so you can de-clutter the debug window.

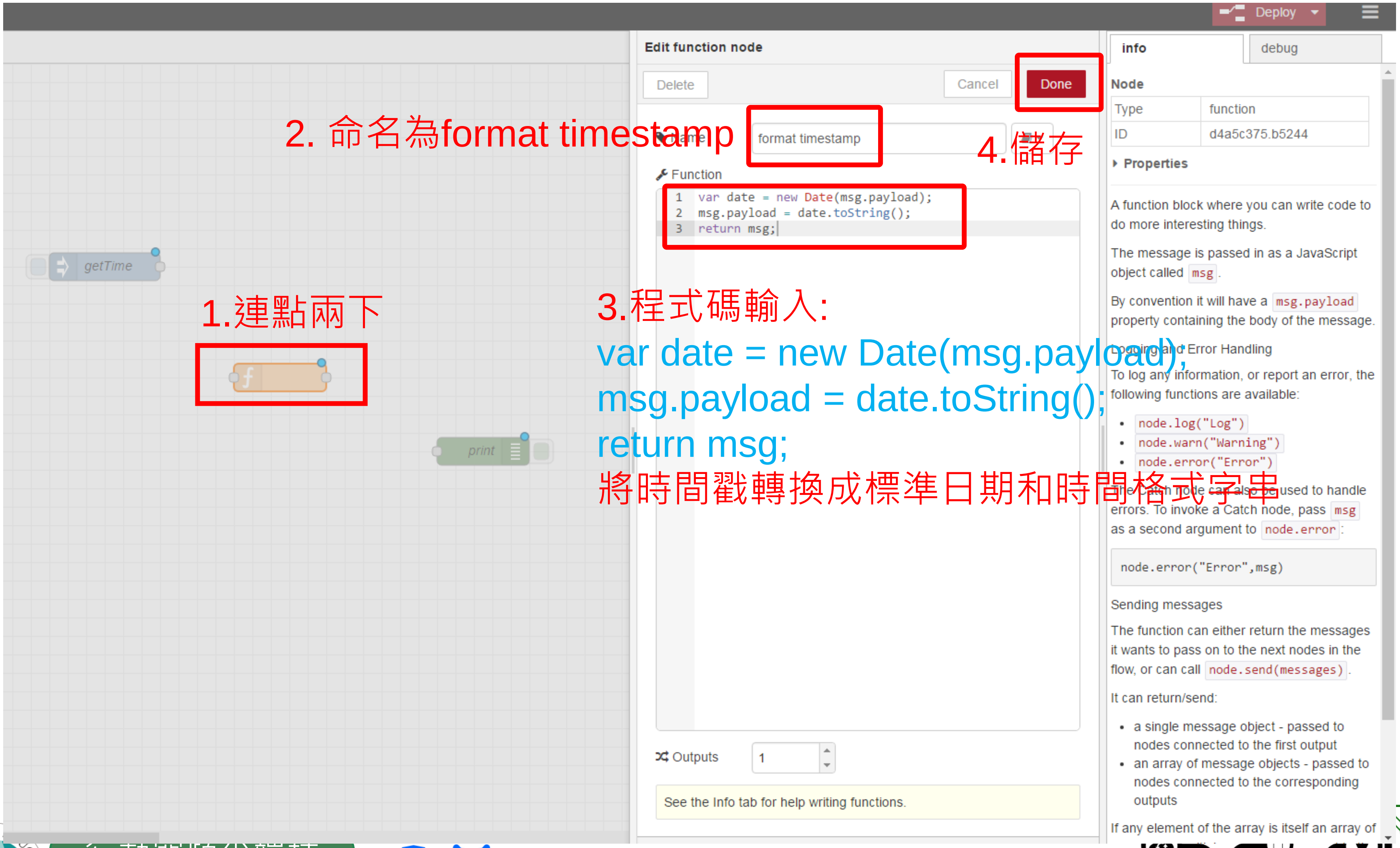
If the payload is an object or buffer it will be stringified first for display and indicate that by saying "(Object)" or "(Buffer)".

Selecting any particular message will highlight (in red) the debug node that reported it. This is useful if you wire up multiple debug nodes.

Optionally can show the complete `msg` object, and send messages to the console log (☰).

In addition any calls to node.warn or node.error will appear here.

Node-RED 範例1: function



2. 命名為format timestamp

format timestamp

4.儲存

Function

```
var date = new Date(msg.payload);
msg.payload = date.toString();
return msg;
```

1.連點兩下

3.程式碼輸入:

var date = new Date(msg.payload);
msg.payload = date.toString();
return msg;
將時間戳轉換成標準日期和時間格式字串

See the Info tab for help writing functions.

Node

Type	function
ID	d4a5c375.b5244

Properties

A function block where you can write code to do more interesting things.

The message is passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

Logging and Error Handling

To log any information, or report an error, the following functions are available:

- `node.log("Log")`
- `node.warn("Warning")`
- `node.error("Error")`

The Catch node can also be used to handle errors. To invoke a Catch node, pass `msg` as a second argument to `node.error`:

```
node.error("Error",msg)
```

Sending messages

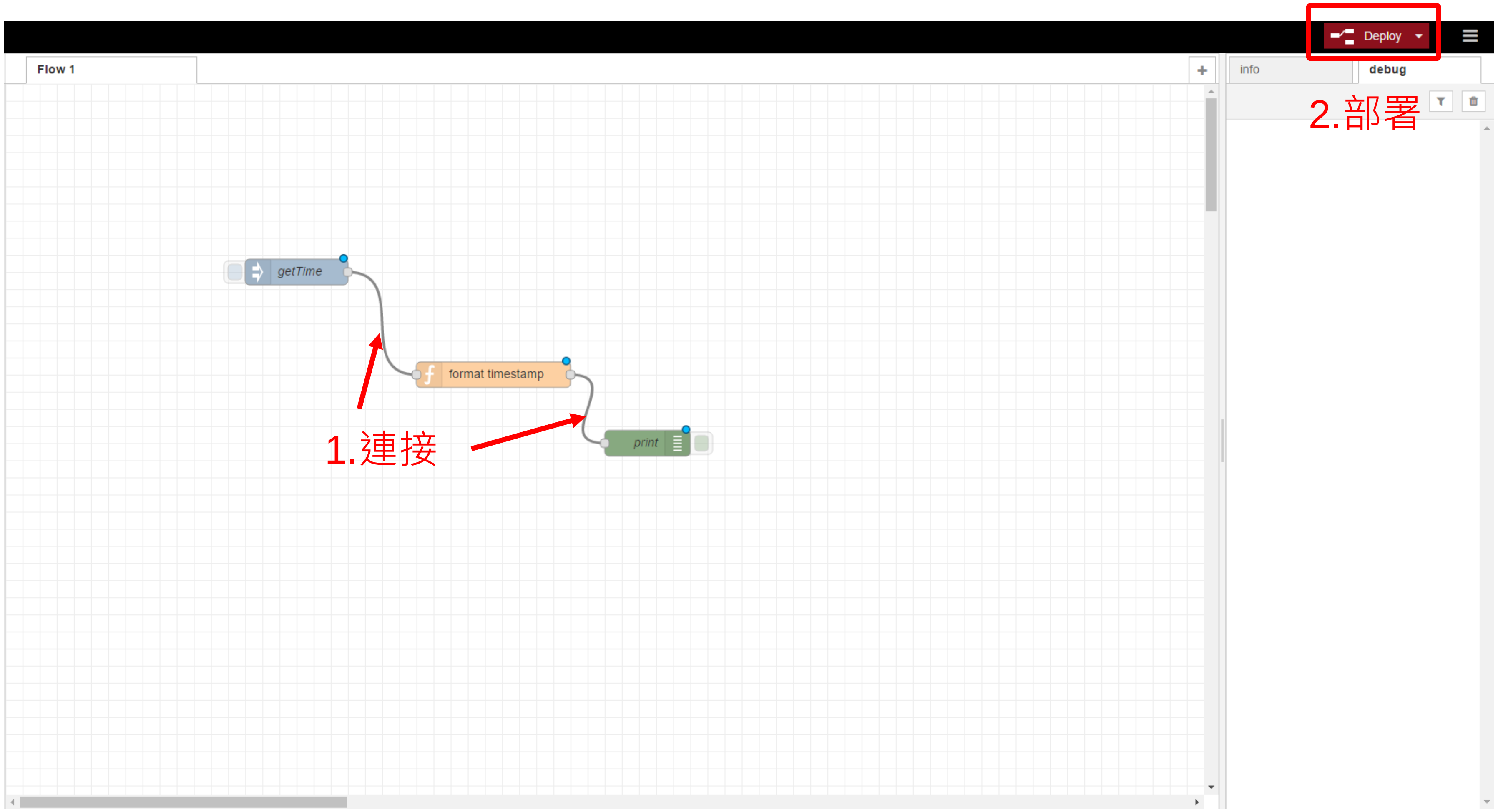
The function can either return the messages it wants to pass on to the next nodes in the flow, or can call `node.send(messages)`.

It can return/send:

- a single message object - passed to nodes connected to the first output
- an array of message objects - passed to nodes connected to the corresponding outputs

If any element of the array is itself an array of

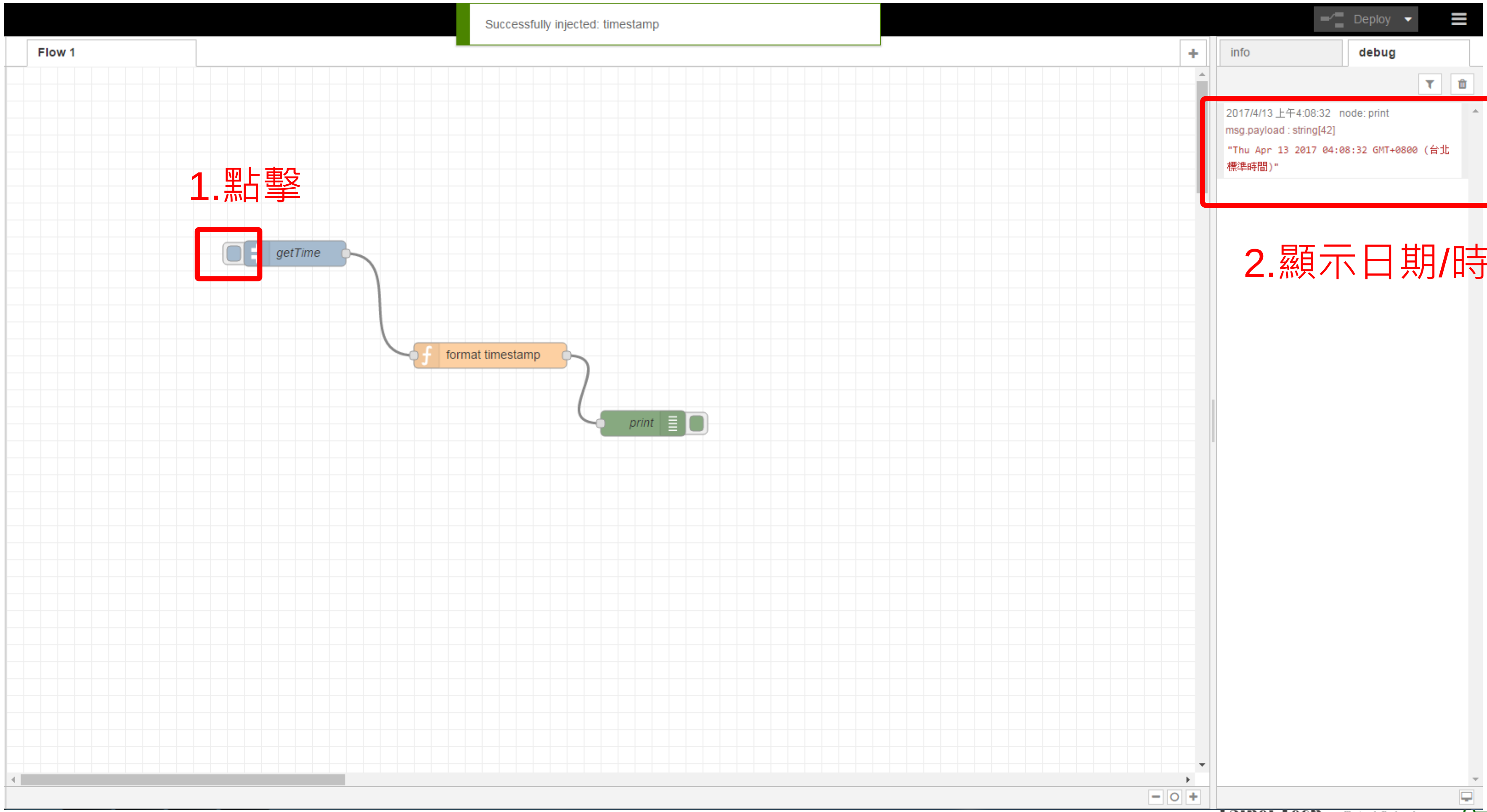
Node-RED 範例1: 連接、部署



行動寬頻尖端技術跨校教學聯盟

文大行動智慧聯網跨校聯盟

Node-RED 範例1: 測試



Successfully injected: timestamp

```
2017/4/13 上午4:08:32  node: print
msg.payload : string[42]
"Thu Apr 13 2017 04:08:32 GMT+0800 (台北標準時間)"
```

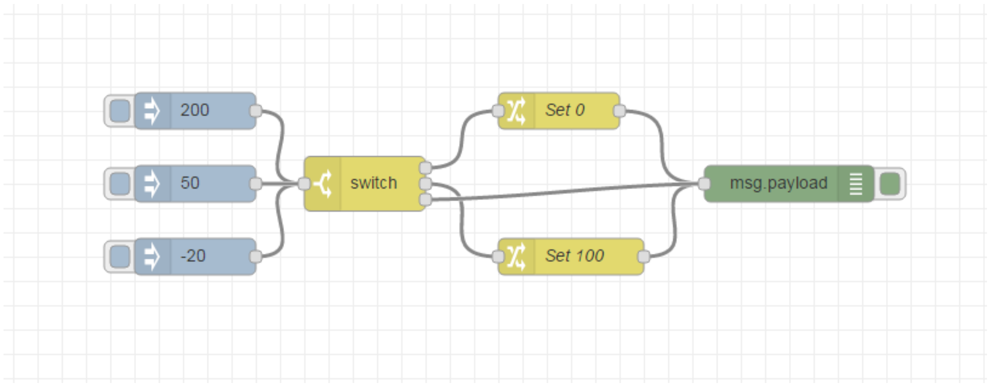
1.點擊

2.顯示日期/時間

Node-RED (Checking Point 2)

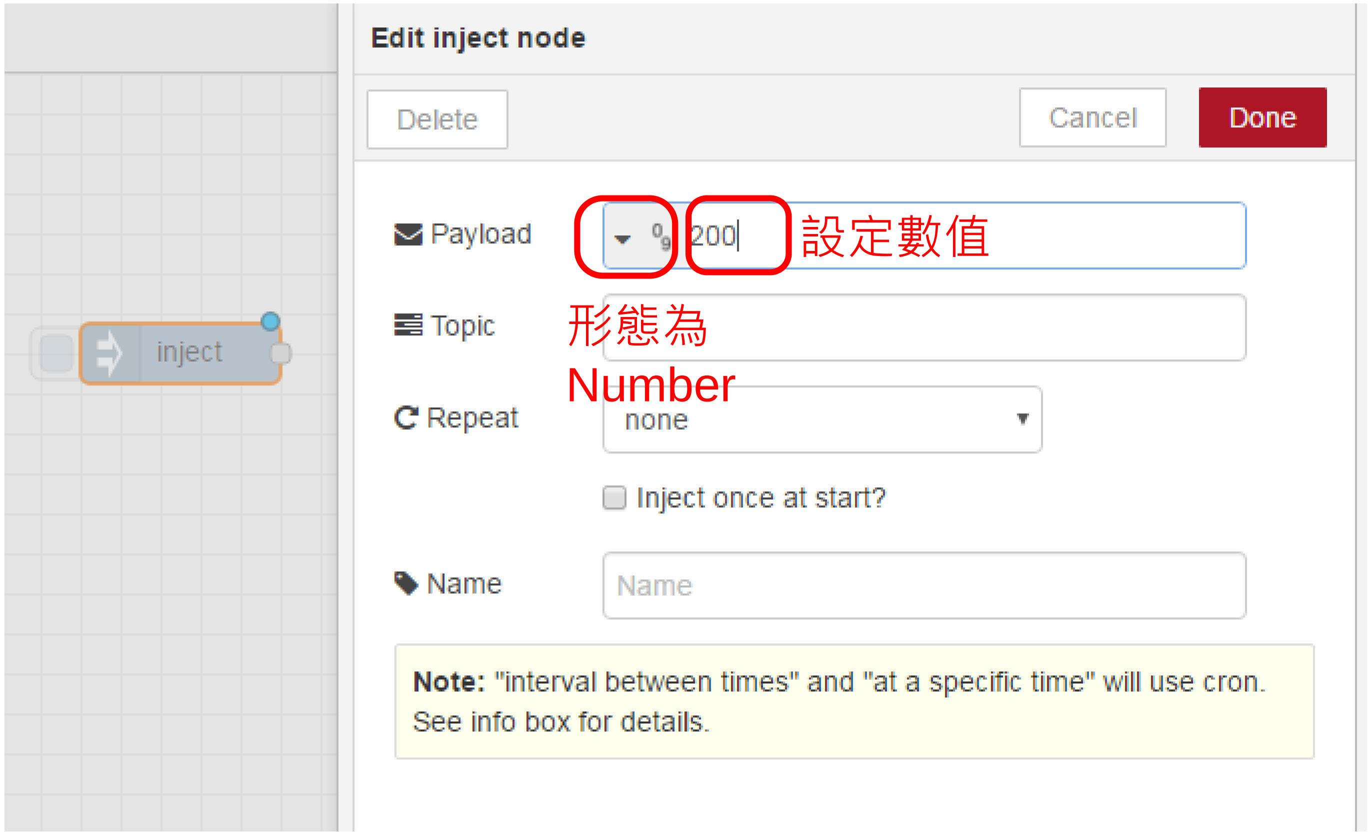
- 完成範例1。

Node-RED 範例2: switch、change



- 範例
  - 利用switch針對msg狀況進行不同的動作，並使用change設定msg
- 功能
  - 利用條件判斷進行數值調整
    - 若輸入>100則輸出100
    - 若輸入<0則輸出0
    - 其他情況不進行調整

Node-RED 範例2: inject



Edit inject node

Payload: 200 設定數值

形態為 Number

Topic:

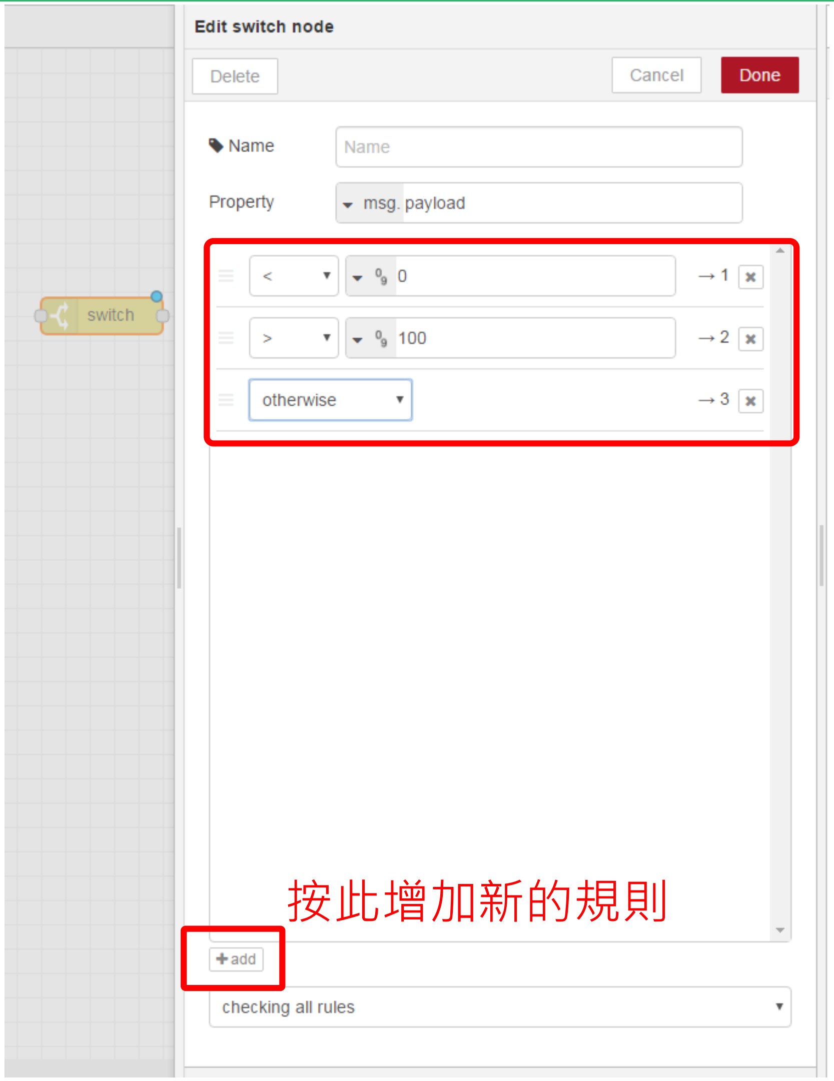
Repeat: none

☐ Inject once at start?

Name: Name

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

Node-RED 範例2: switch



Edit switch node

Delete | Cancel | Done

Name: Name

Property: msg. payload

```
<   0     → 1
>   100   → 2
otherwise → 3
```

+add

checking all rules

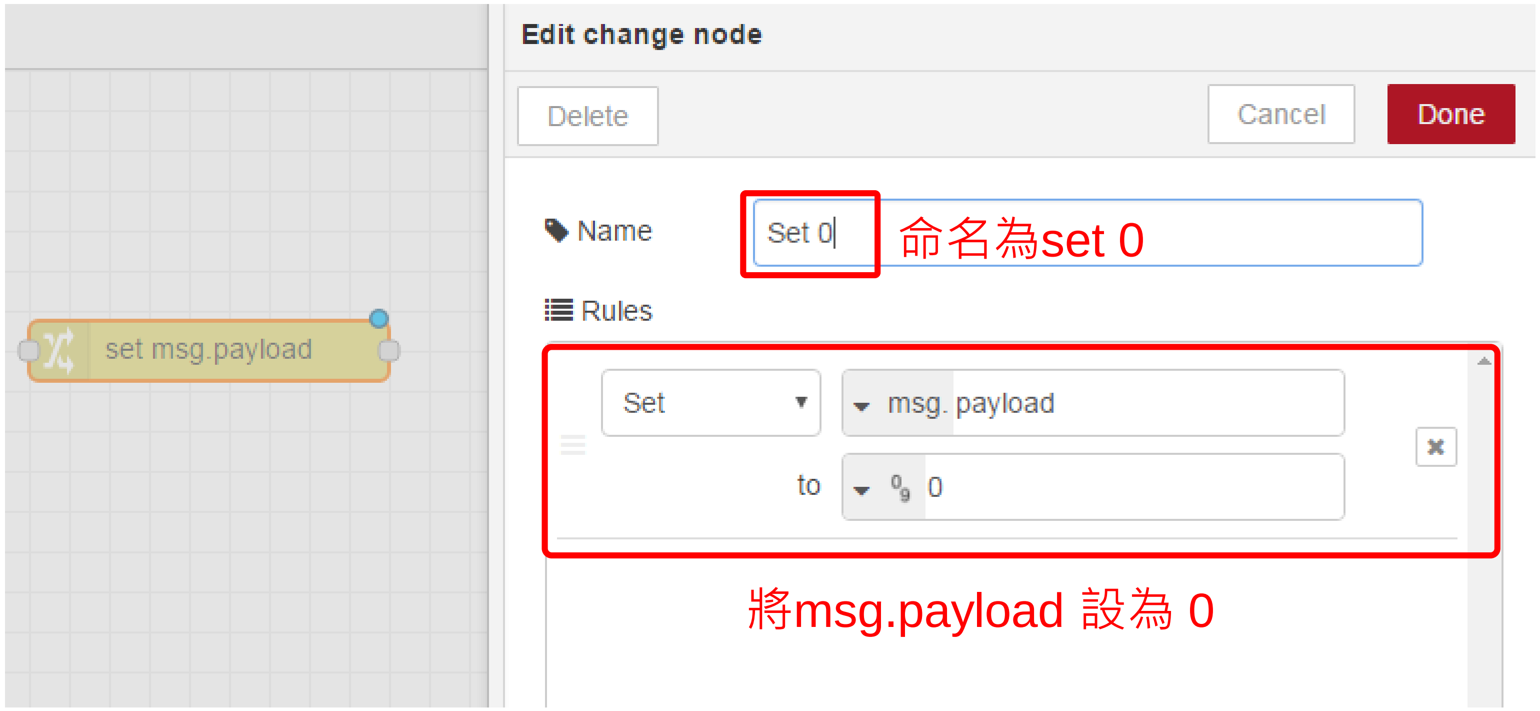
按此增加新的規則

分歧1: < 0
分歧2: > 100
分歧3: otherwise

switch利用條件
執行不同流程塊

行動寬頻尖端技
術跨校教學聯盟

Node-RED 範例2: change



Edit change node

Delete | Cancel | Done

Name: Set 0

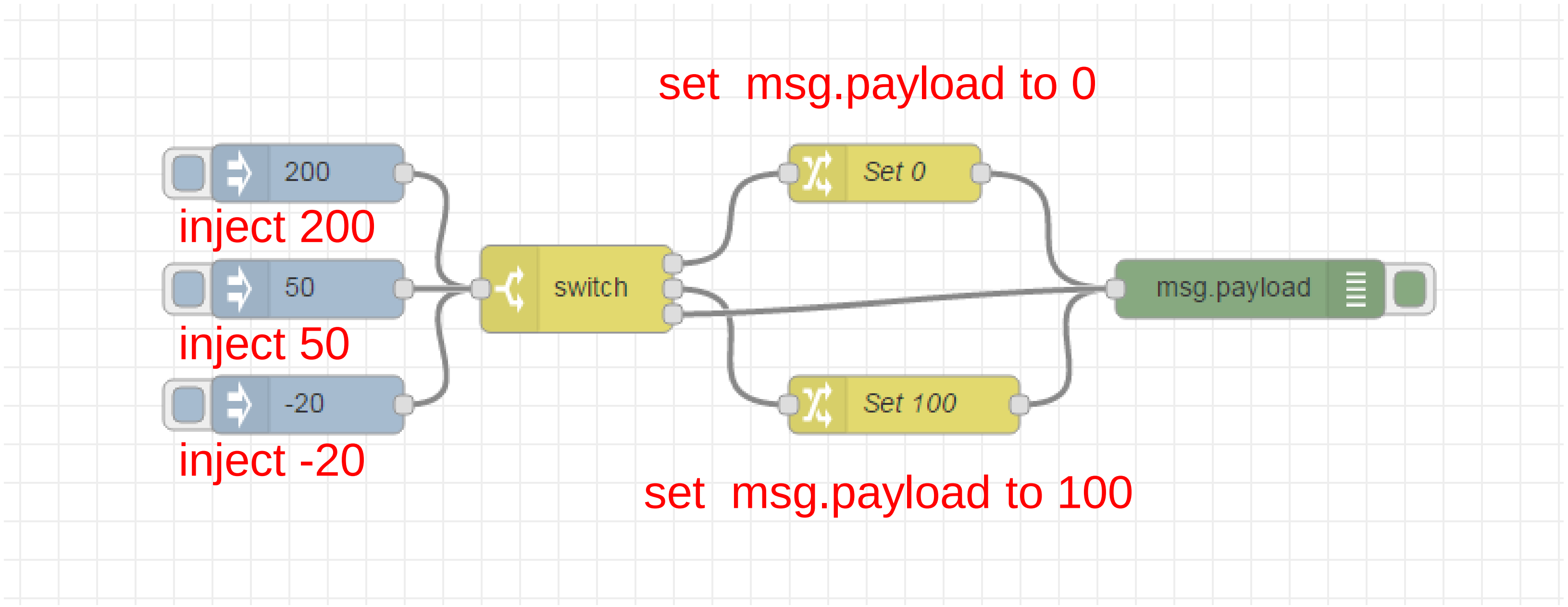
命名為set 0

Rules

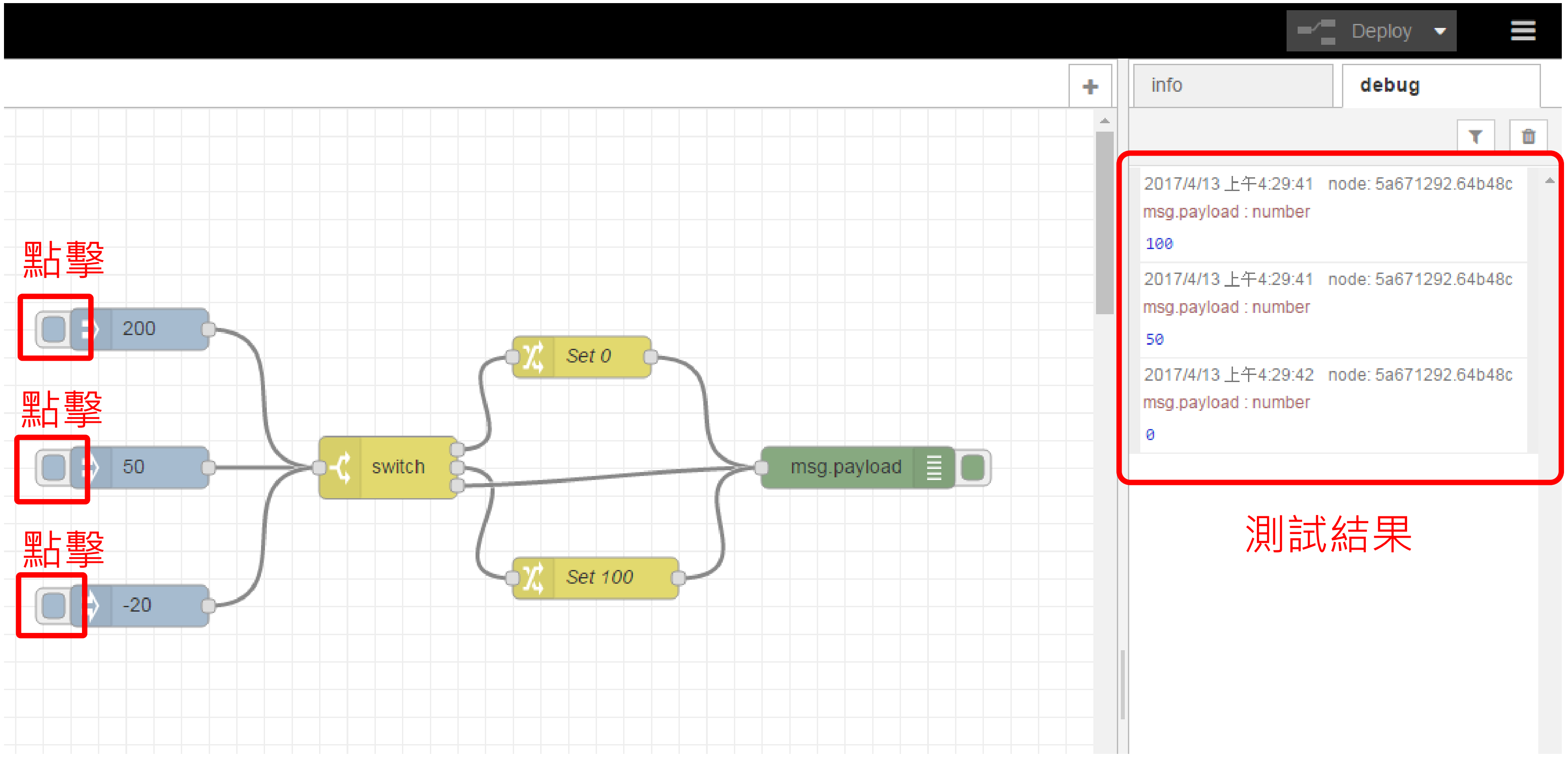
Set msg.payload to 0

將msg.payload 設為 0

Node-RED 範例2: 連接



Node-RED 範例2: 測試



```
2017/4/13 上午4:29:41   node: 5a671292.64b48c
msg.payload : number
100
2017/4/13 上午4:29:41   node: 5a671292.64b48c
msg.payload : number
50
2017/4/13 上午4:29:42   node: 5a671292.64b48c
msg.payload : number
0
```

Node-RED (Checking Point 3)

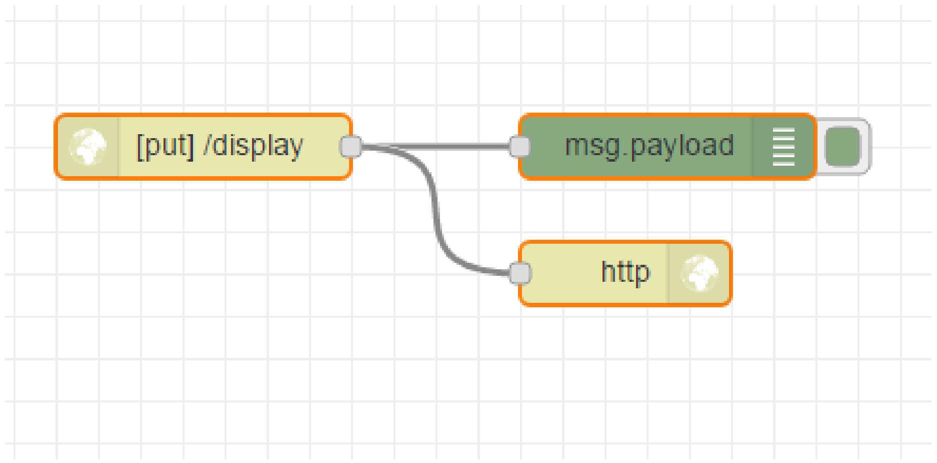
- 完成範例2

Node-RED & Postman整合測試

- Node-RED
  - 產生一Web Service

- Postman
  - 產生HTTP Request存取Web Service

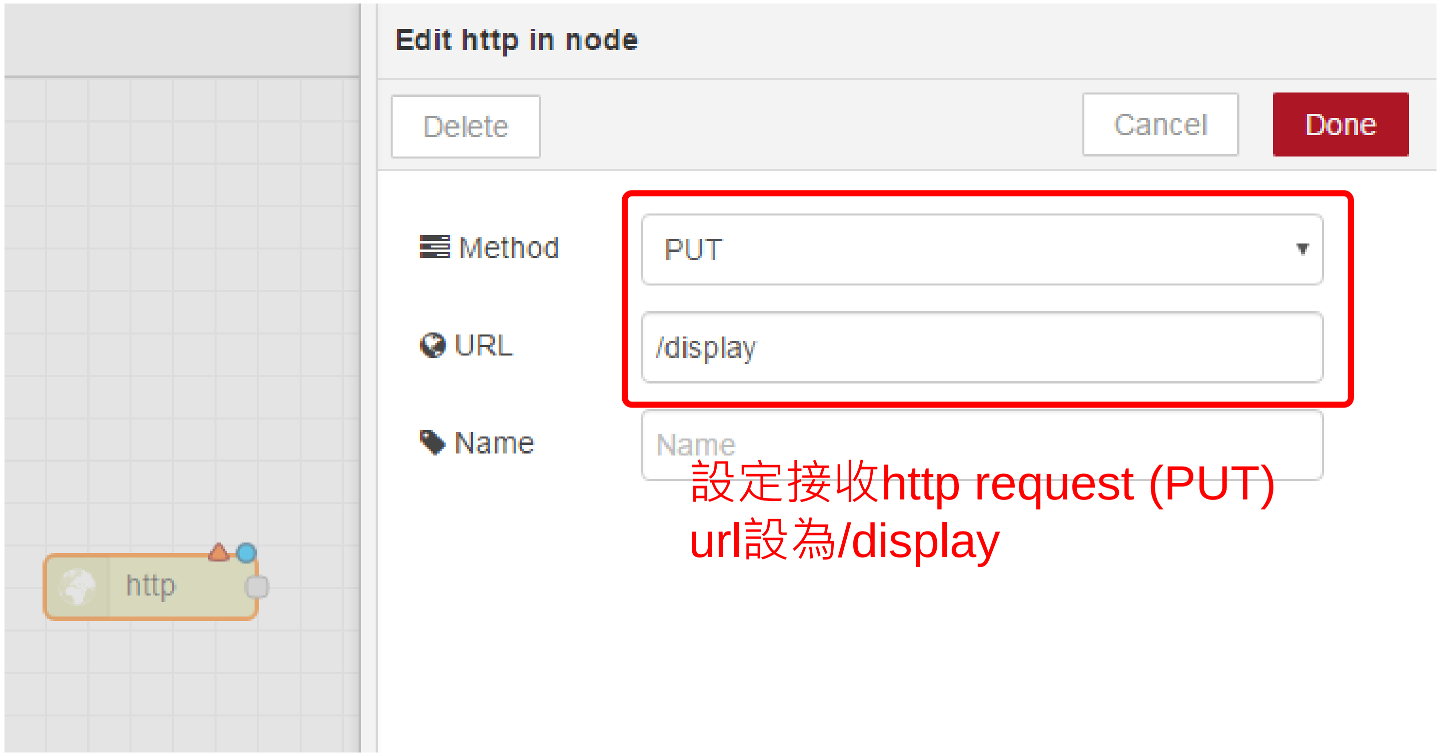
Node-RED 範例3: HTTP Web Service



❖ 範例
- 建構一簡單的Web Service
- 測試此Web Service是否正常運作

❖ 功能
- 監聽/display的http request (Put)
- 記錄於debug.log中
- 返回http response

Node-RED 範例3: HTTP



Edit http in node

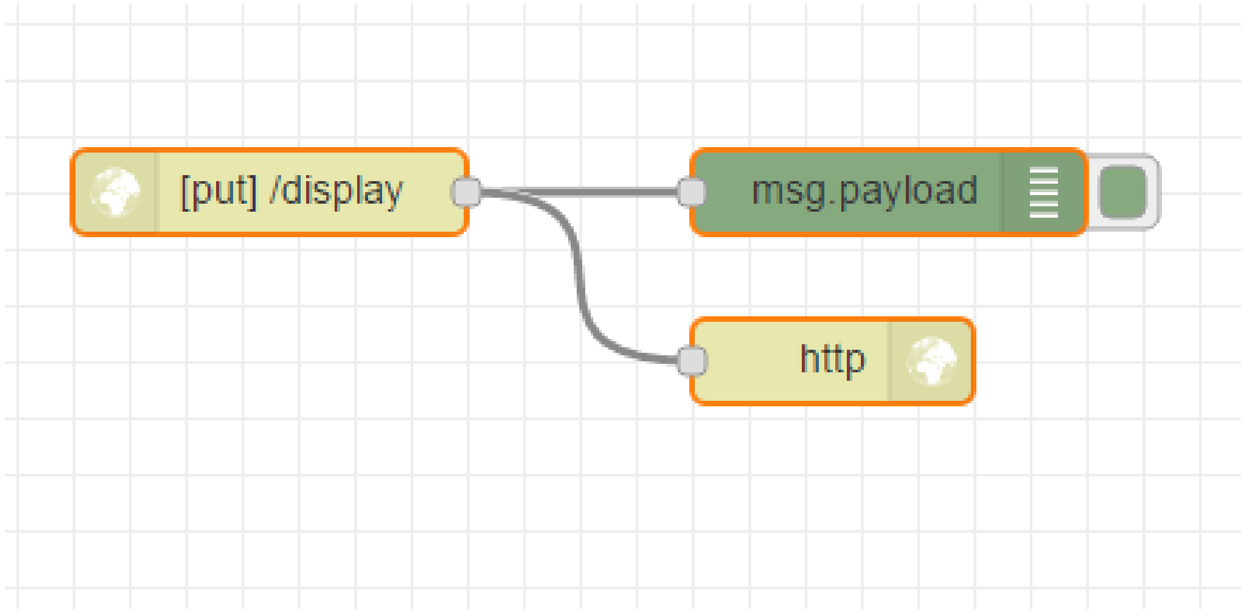
Method: PUT

URL: /display

Name: Name

設定接收http request (PUT)
url設為/display

Node-RED 範例3: 連接

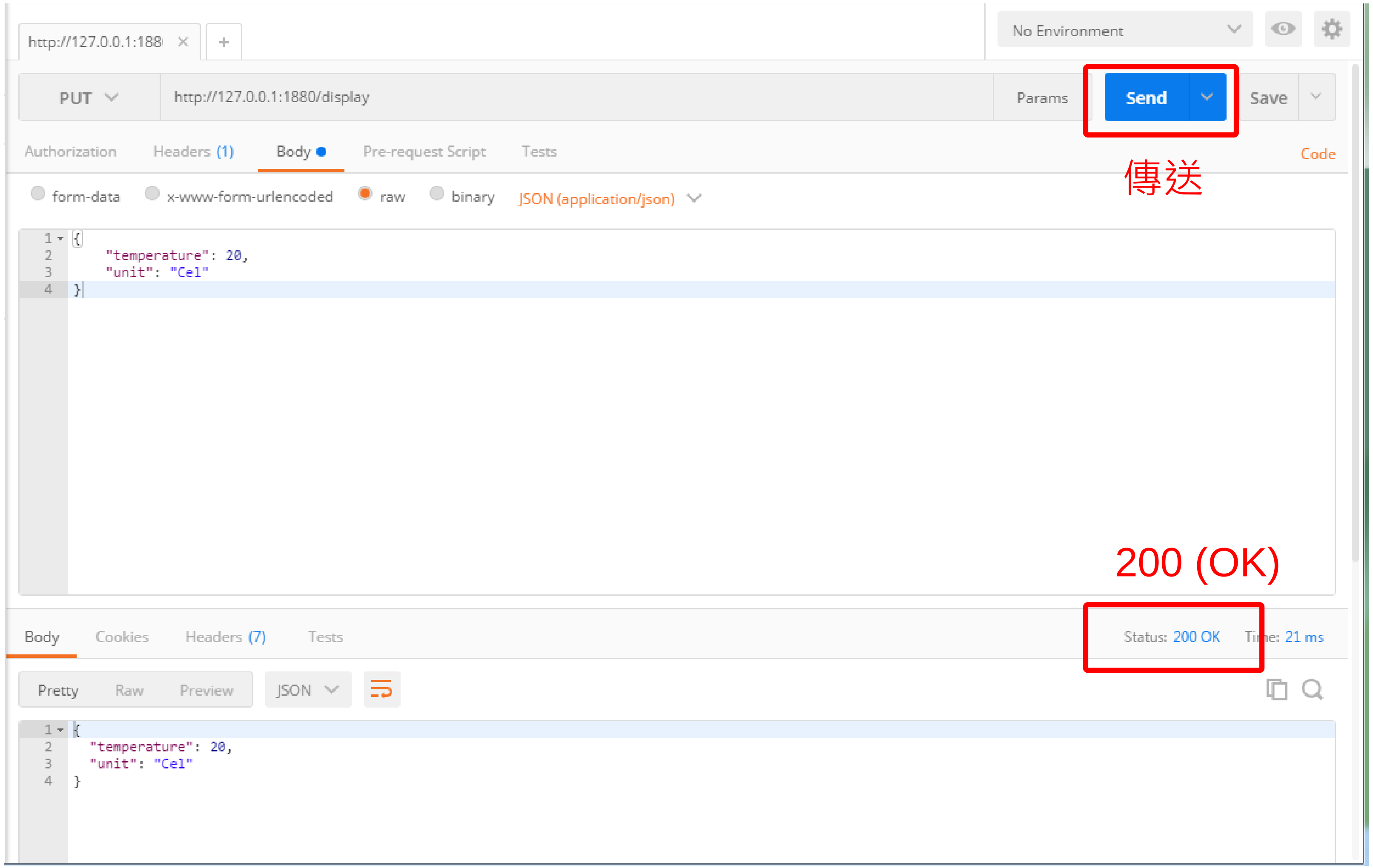


PostMan 範例3: 參數設定

- 設定Postman各項參數，如下
  - Method：Put
  - URI：http://127.0.0.1:1800/display
  - Headers
    - Content-Type: application/json
  - Body

```
{
    "temperature": 20,
    "unit": "Cel"
}
```

PostMan 範例3: 測試



PUT http://127.0.0.1:1880/display

form-data   x-www-form-urlencoded   [✓ raw]   binary   JSON (application/json)

```
{
    "temperature": 20,
    "unit": "Cel"
}
```

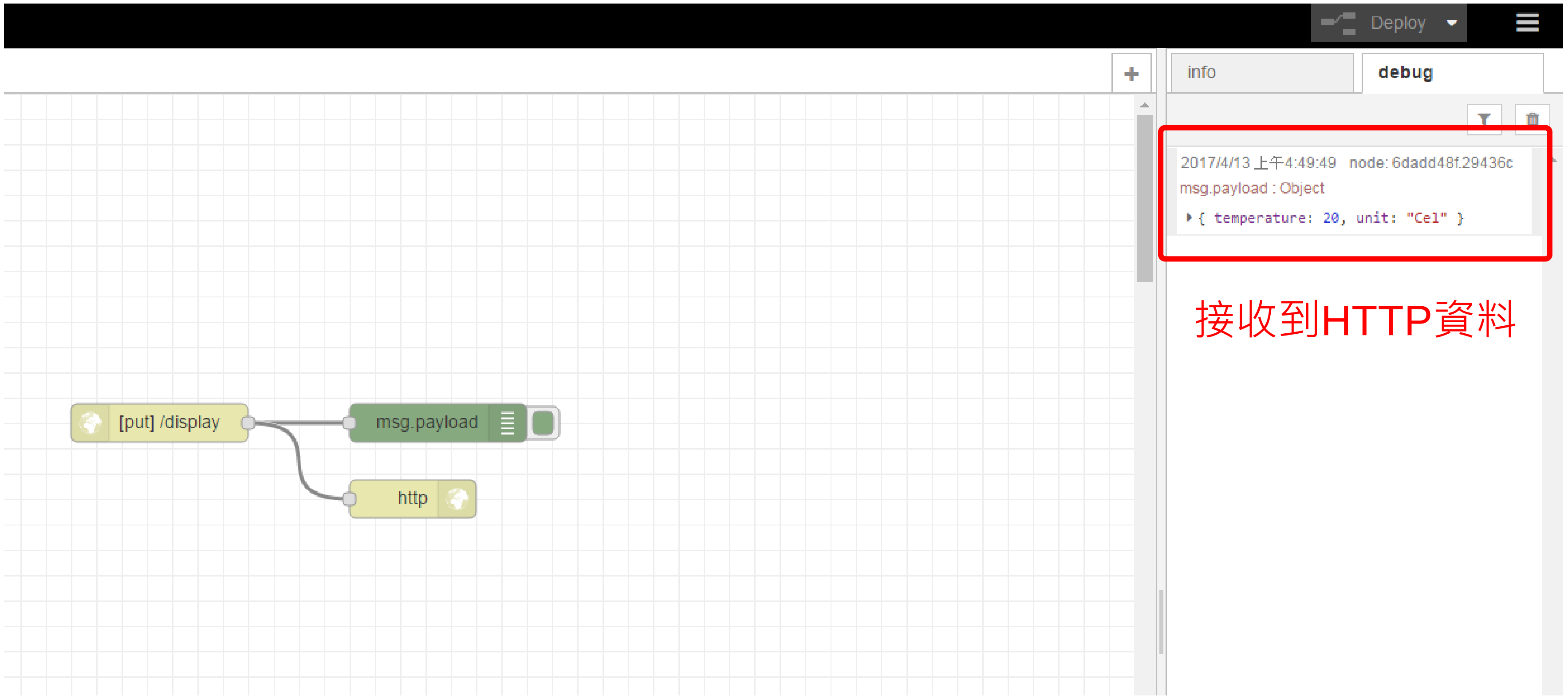
傳送

200 (OK)

Status: 200 OK   Time: 21 ms

```
{
  "temperature": 20,
  "unit": "Cel"
}
```

Node-RED 範例3: 測試



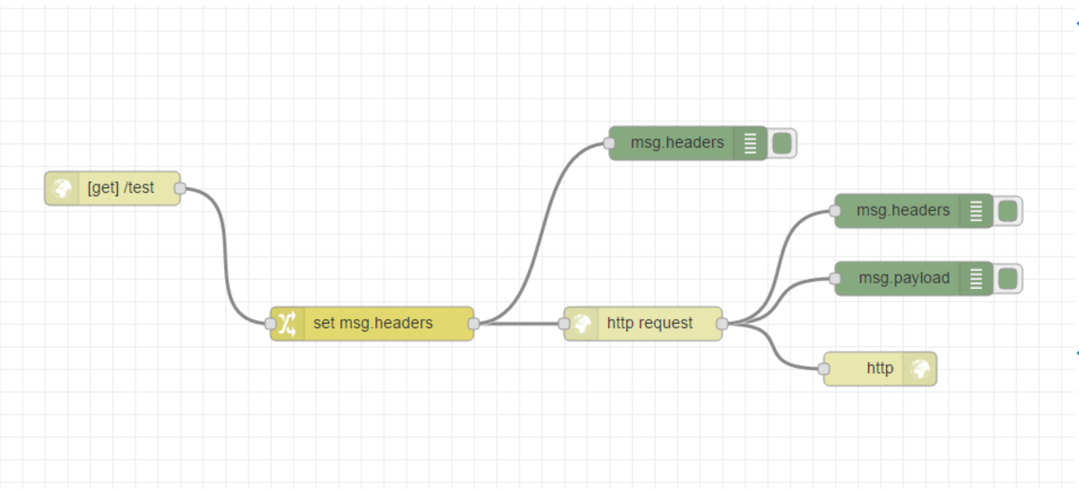
```
2017/4/13 上午4:49:49   node: 6dadd48f.29436c
msg.payload : Object
▸{ temperature: 20, unit: "Cel" }
```

接收到HTTP資料

Node-RED (Checking Point 4)

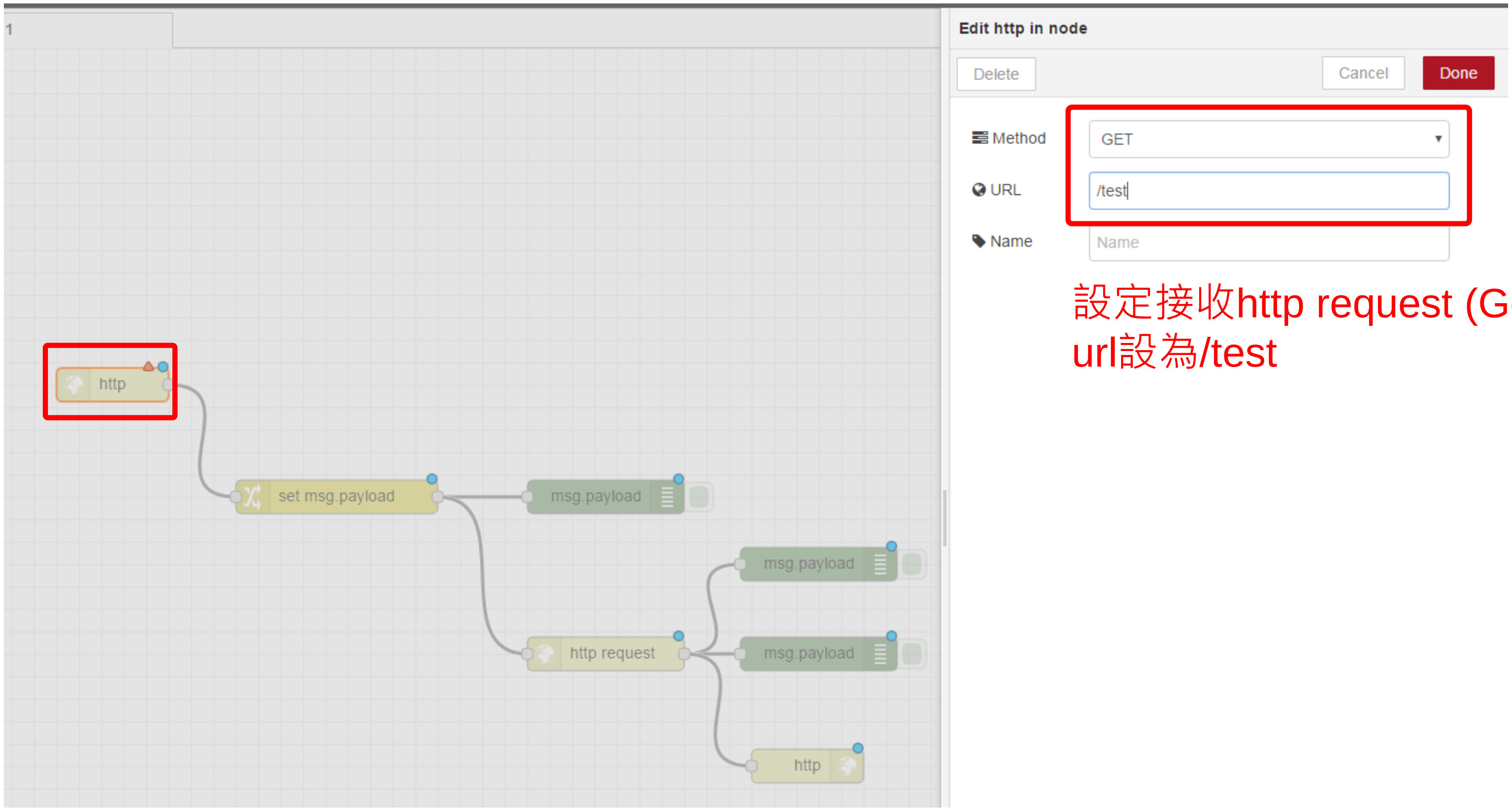
❖ 完成範例3

Node-RED 範例4: HTTP Request



- 範例
  - 建構一簡單的代理伺服器(proxy)
  - 測試是否正常運作
- 功能
  - 監聽/test的http request (Get)
  - 設定Headers
  - 發送http request至Google Map API
  - 取得地理資訊
  - 返回http response

Node-RED 範例4: HTTP Request



Edit http in node

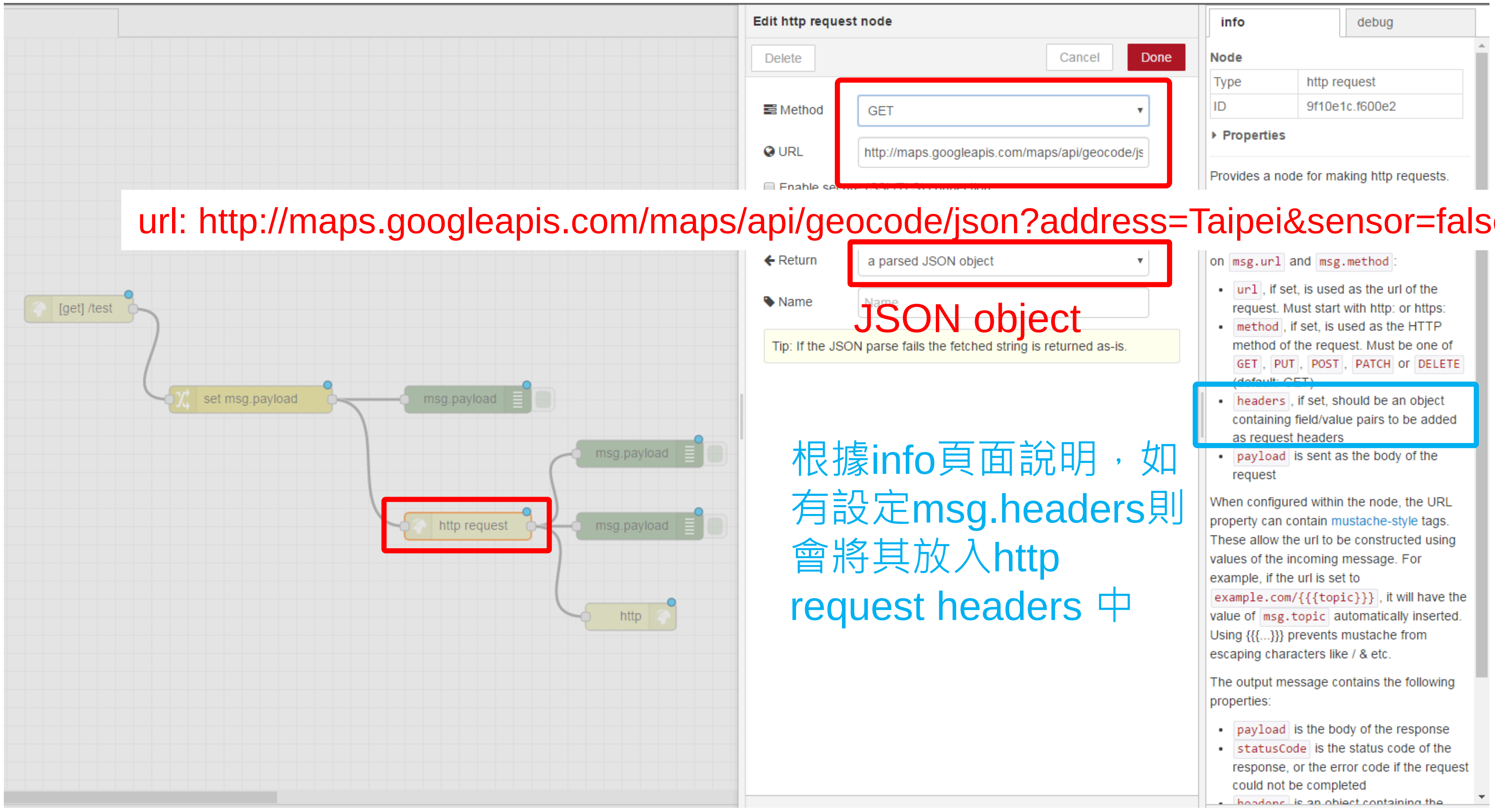
Method: GET

URL: /test

Name: Name

設定接收http request (Get)
url設為/test

Node-RED 範例4: HTTP Request



Edit http request node

Delete | Cancel | Done

Method: GET

URL: http://maps.googleapis.com/maps/api/geocode/js

Enable se

url: http://maps.googleapis.com/maps/api/geocode/json?address=Taipei&sensor=false

Return: a parsed JSON object

Name: Name

JSON object

Tip: If the JSON parse fails the fetched string is returned as-is.

根據info頁面說明，如有設定msg.headers則會將其放入http request headers 中

info | debug

Node

Type	http request
ID	9f10e1c.f600e2

▸ **Properties**

Provides a node for making http requests.

on `msg.url` and `msg.method`:

- `url`, if set, is used as the url of the request. Must start with http: or https:
- `method`, if set, is used as the HTTP method of the request. Must be one of `GET`, `PUT`, `POST`, `PATCH` or `DELETE` (default: GET)
- `headers`, if set, should be an object containing field/value pairs to be added as request headers
- `payload` is sent as the body of the request

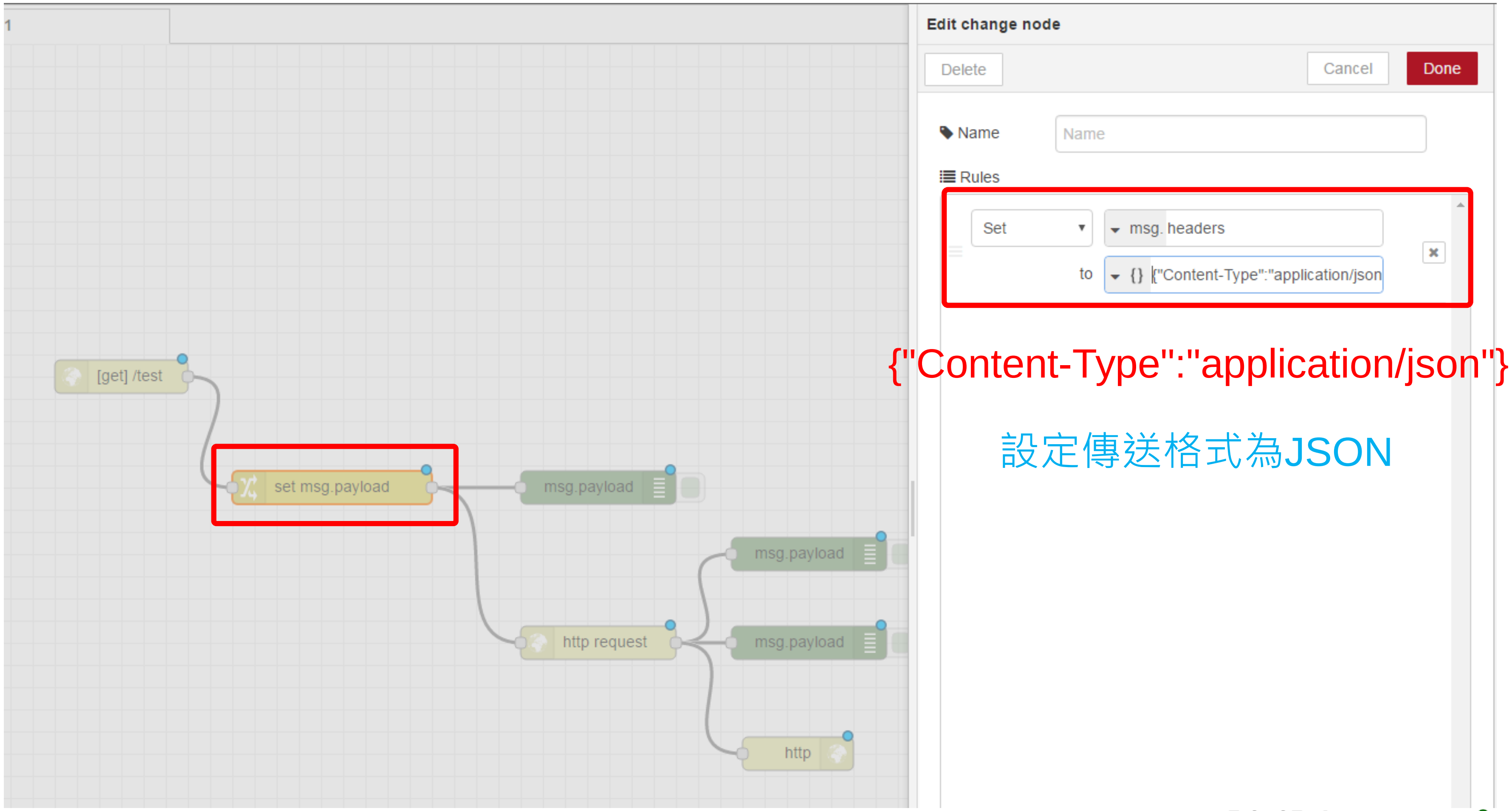
When configured within the node, the URL property can contain mustache-style tags. These allow the url to be constructed using values of the incoming message. For example, if the url is set to `example.com/{{{topic}}}`, it will have the value of `msg.topic` automatically inserted. Using {{{...}}} prevents mustache from escaping characters like / & etc.

The output message contains the following properties:

- `payload` is the body of the response
- `statusCode` is the status code of the response, or the error code if the request could not be completed
- `headers` is an object containing the

行動寬頻尖端技術跨校教學聯盟

Node-RED 範例4: change



Edit change node

Name: Name

Rules

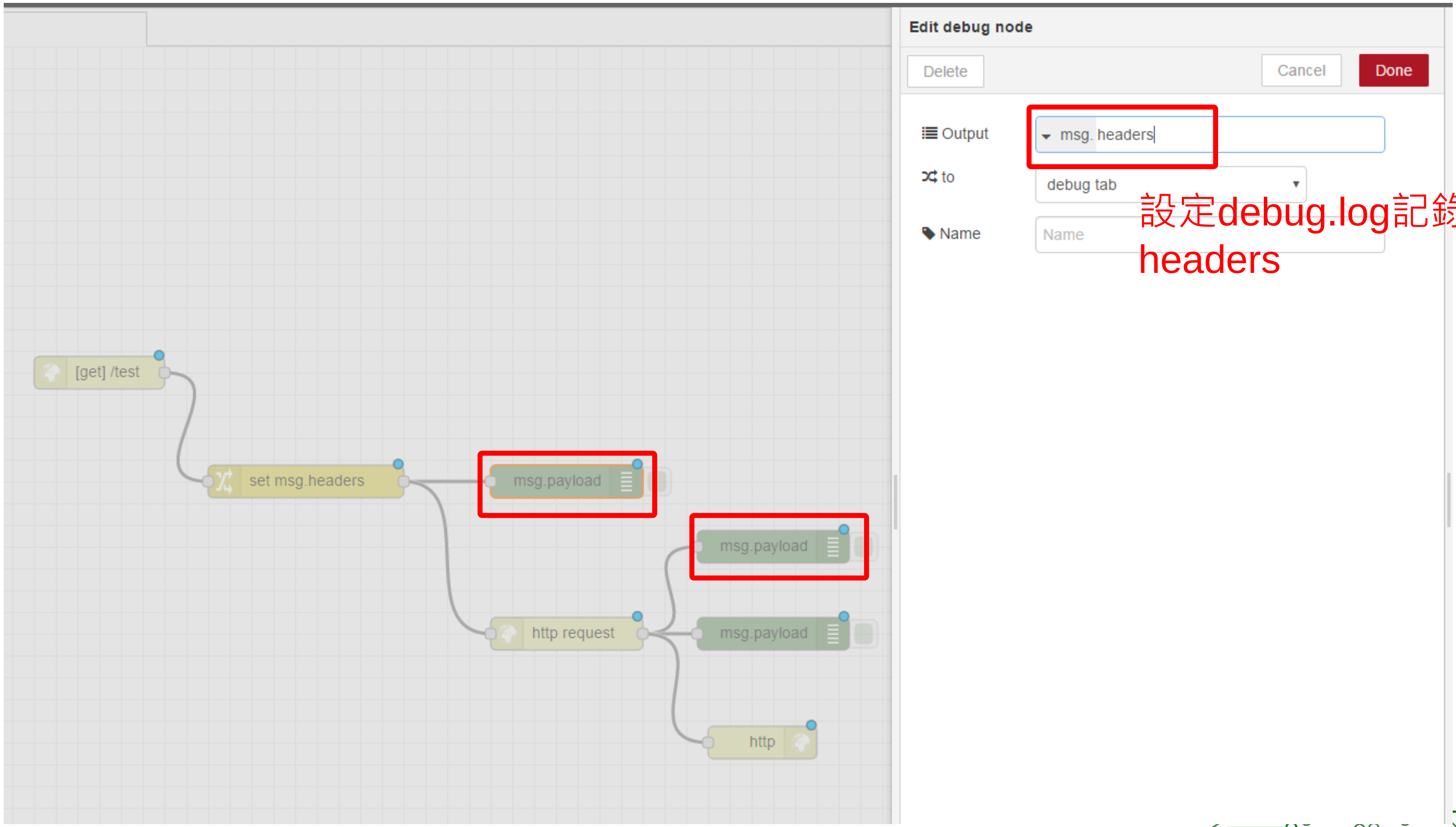
Set msg. headers

to {} {"Content-Type":"application/json

{"Content-Type":"application/json"}

設定傳送格式為JSON

Node-RED 範例4: debug



Edit debug node

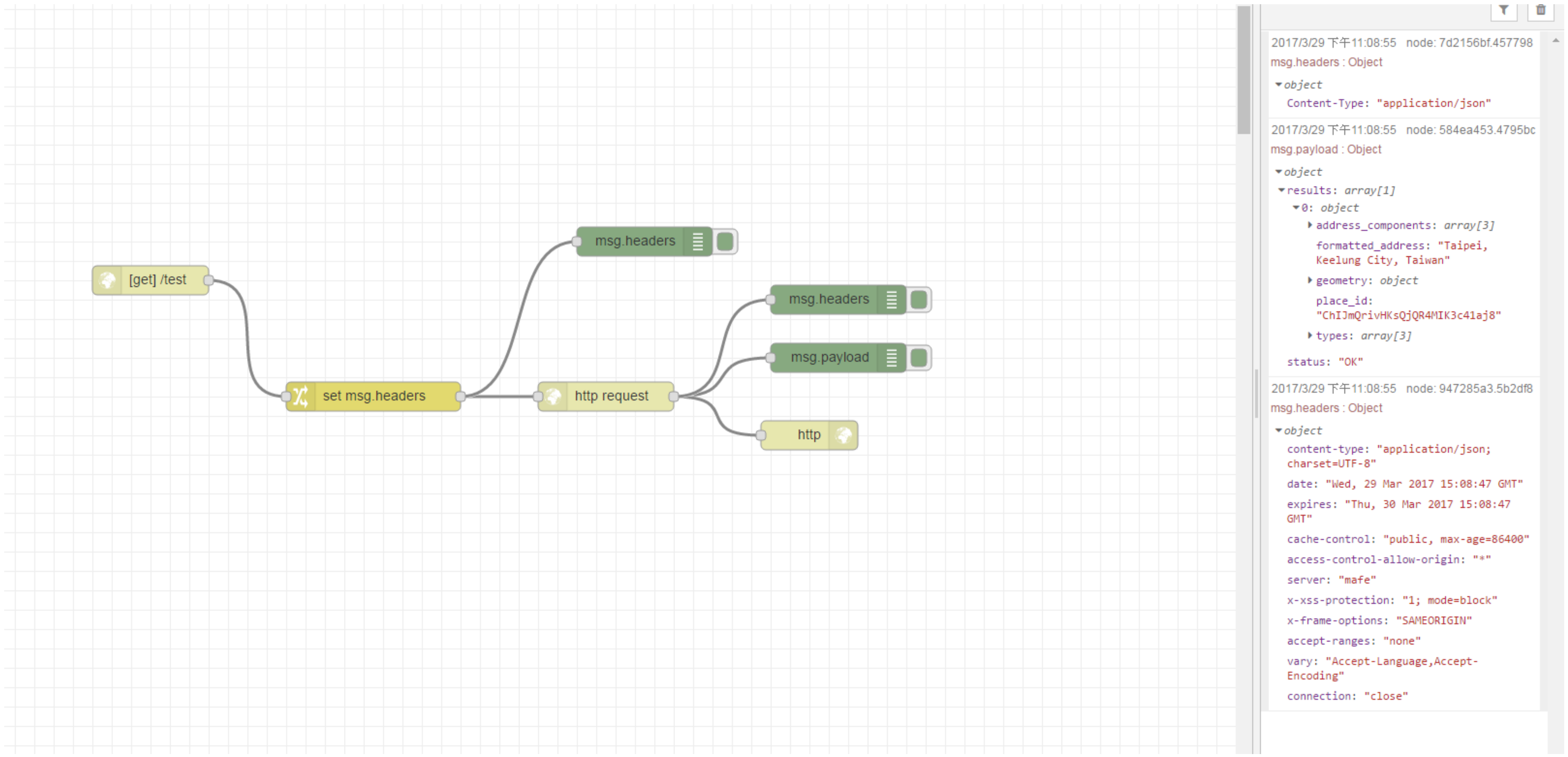
Output: msg. headers

to: debug tab

Name: Name

設定debug.log記錄headers

Node-RED 範例4: 連接

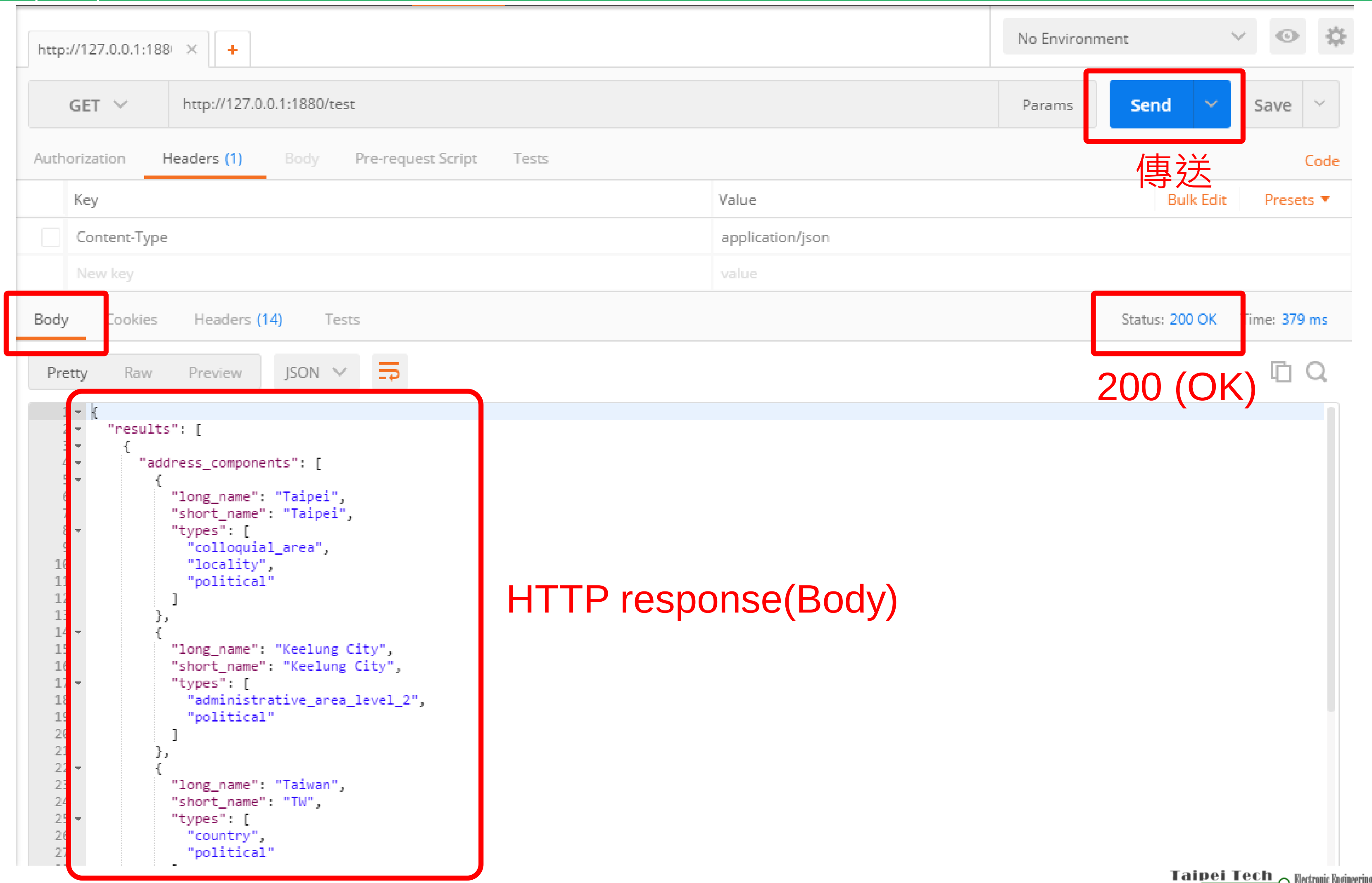


```
2017/3/29 下午11:08:55  node: 7d2156bf.457798
msg.headers : Object
▾object
  Content-Type: "application/json"

2017/3/29 下午11:08:55  node: 584ea453.4795bc
msg.payload : Object
▾object
 ▾results: array[1]
   ▾0: object
     ▸address_components: array[3]
      formatted_address: "Taipei,
      Keelung City, Taiwan"
     ▸geometry: object
      place_id:
      "ChIJmQrivHKsQjQR4MIK3c41aj8"
     ▸types: array[3]
  status: "OK"

2017/3/29 下午11:08:55  node: 947285a3.5b2df8
msg.headers : Object
▾object
  content-type: "application/json;
  charset=UTF-8"
  date: "Wed, 29 Mar 2017 15:08:47 GMT"
  expires: "Thu, 30 Mar 2017 15:08:47
  GMT"
  cache-control: "public, max-age=86400"
  access-control-allow-origin: "*"
  server: "mafe"
  x-xss-protection: "1; mode=block"
  x-frame-options: "SAMEORIGIN"
  accept-ranges: "none"
  vary: "Accept-Language,Accept-
  Encoding"
  connection: "close"
```

PostMan 範例4: 測試

- 設定Postman各項參數，如下
  - Method：Get
  - URL：http://127.0.0.1:1800/test

PostMan 範例4: 測試



http://127.0.0.1:1880/test

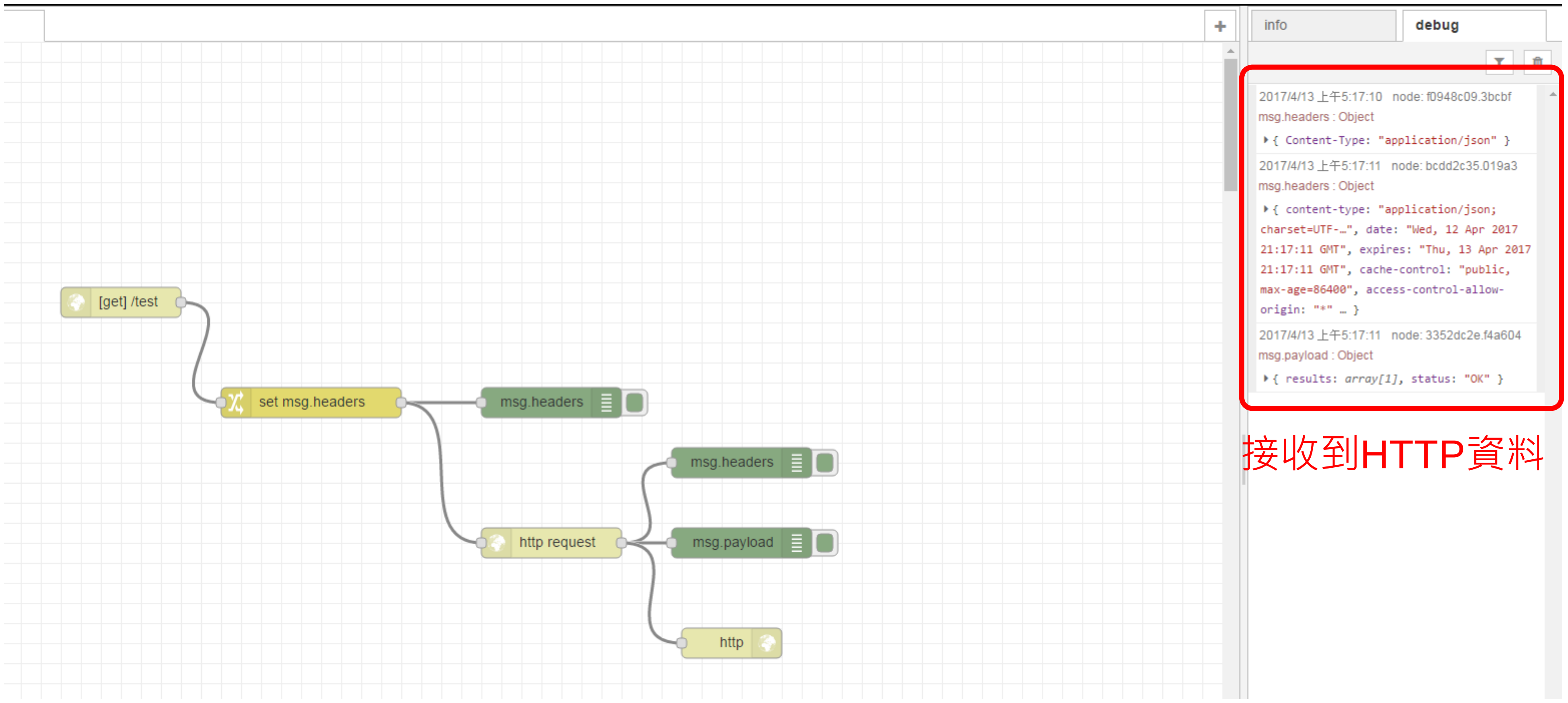
Key	Value
Content-Type	application/json
New key	value

Status: 200 OK　Time: 379 ms

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "Taipei",
          "short_name": "Taipei",
          "types": [
            "colloquial_area",
            "locality",
            "political"
          ]
        },
        {
          "long_name": "Keelung City",
          "short_name": "Keelung City",
          "types": [
            "administrative_area_level_2",
            "political"
          ]
        },
        {
          "long_name": "Taiwan",
          "short_name": "TW",
          "types": [
            "country",
            "political"
```

行動寬頻尖端技術跨校教學聯盟

Node-RED 範例4: 測試



```
2017/4/13 上午5:17:10   node: f0948c09.3bcbf
msg.headers : Object
▸{ Content-Type: "application/json" }
2017/4/13 上午5:17:11   node: bcdd2c35.019a3
msg.headers : Object
▸{ content-type: "application/json;
charset=UTF-…", date: "Wed, 12 Apr 2017
21:17:11 GMT", expires: "Thu, 13 Apr 2017
21:17:11 GMT", cache-control: "public,
max-age=86400", access-control-allow-
origin: "*" … }
2017/4/13 上午5:17:11   node: 3352dc2e.f4a604
msg.payload : Object
▸{ results: array[1], status: "OK" }
```

接收到HTTP資料

行動寬頻尖端技術跨校教學聯盟

文大行動智慧聯網跨校聯盟

Node-RED (Checking Point 5)

- 完成範例4

總結

- Node-RED是一個視覺化之程式開發工具
  - 以Node.js為基礎
  - 支援多平台(如：Raspberry Pi、Arduino、Android…)、多系統(如：Windows、Linux…)
  - 可增加套件，擴充支援的功能