物聯網實務 HW2

電機碩一 11278008 林佳慧

日期:2023/09/20

Exercise 2-1

Step 1:Install Node.js

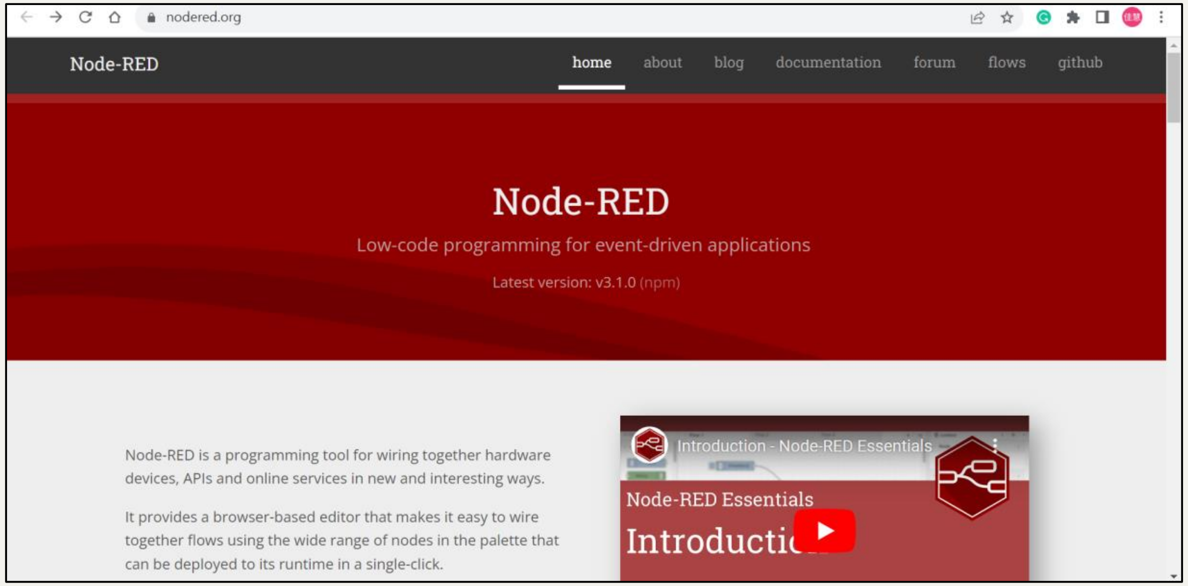
Step 2:Install Node-RED

- Open Node.js command
- npm install -g --unsafe-perm node-red

Step 3: Run Node-RED

- node-red

1. 到Node-RED網站https://nodered.org/



Node-RED

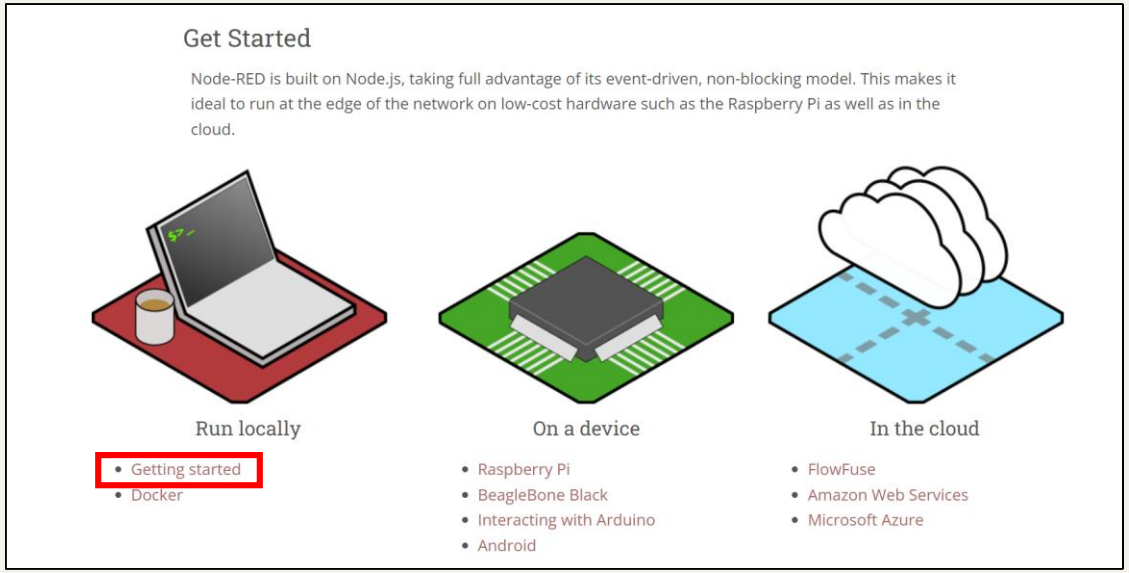
Low-code programming for event-driven applications

Latest version: v3.1.0 (npm)

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

2. 下滑點選Getting started



Get Started

Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

Run locally

- Getting started
- Docker

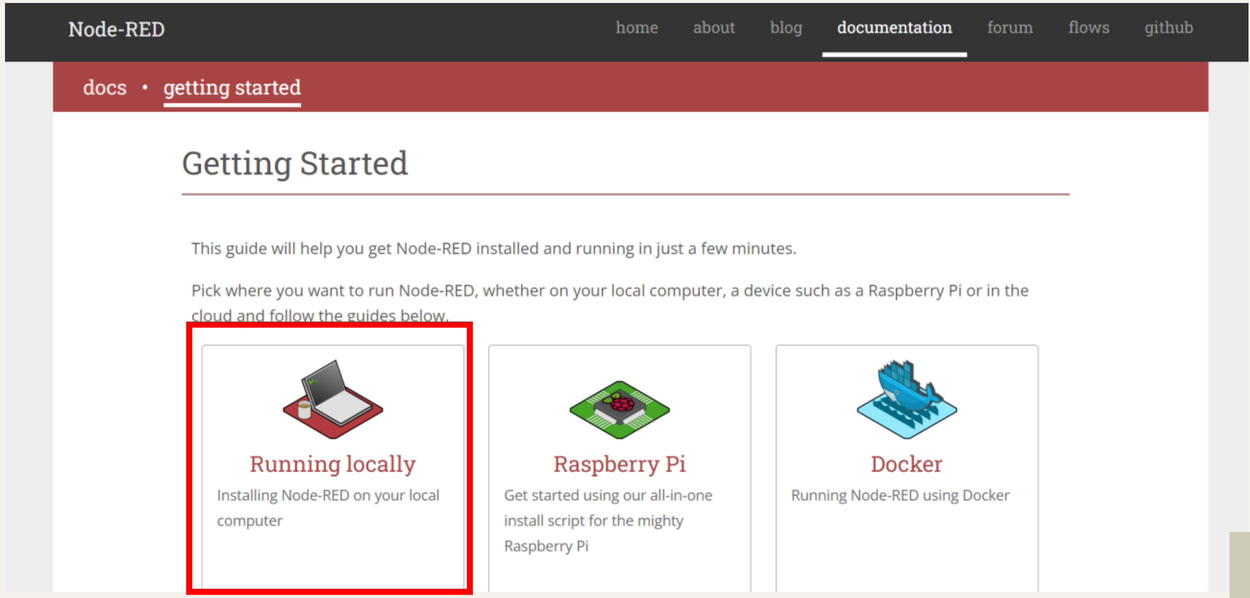
On a device

- Raspberry Pi
- BeagleBone Black
- Interacting with Arduino
- Android

In the cloud

- FlowFuse
- Amazon Web Services
- Microsoft Azure

3. 點選Running locally



Getting Started

This guide will help you get Node-RED installed and running in just a few minutes.

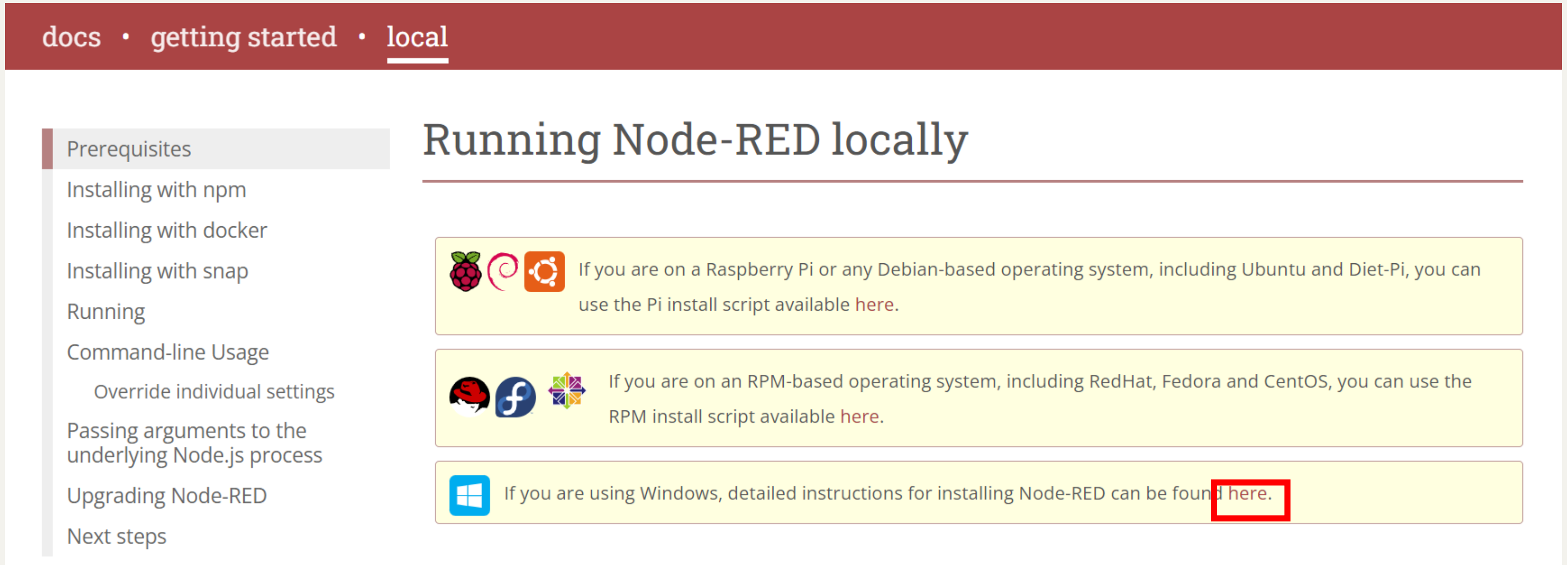
Pick where you want to run Node-RED, whether on your local computer, a device such as a Raspberry Pi or in the cloud and follow the guides below.

Running locally
Installing Node-RED on your local computer

Raspberry Pi
Get started using our all-in-one install script for the mighty Raspberry Pi

Docker
Running Node-RED using Docker

4. 點選here



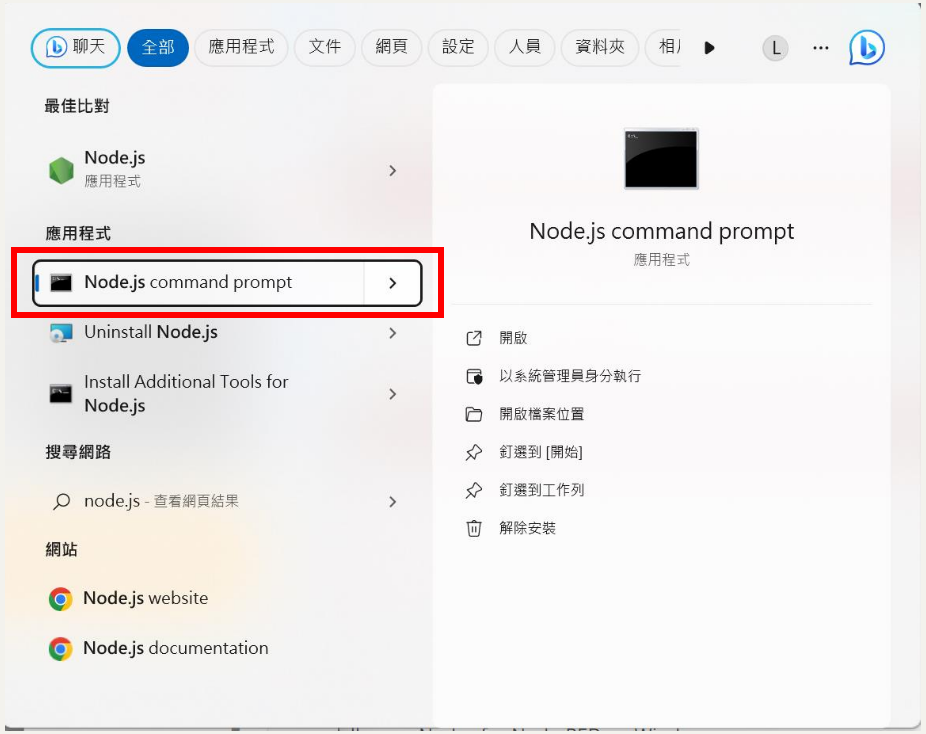
Running Node-RED locally

If you are on a Raspberry Pi or any Debian-based operating system, including Ubuntu and Diet-Pi, you can use the Pi install script available here.

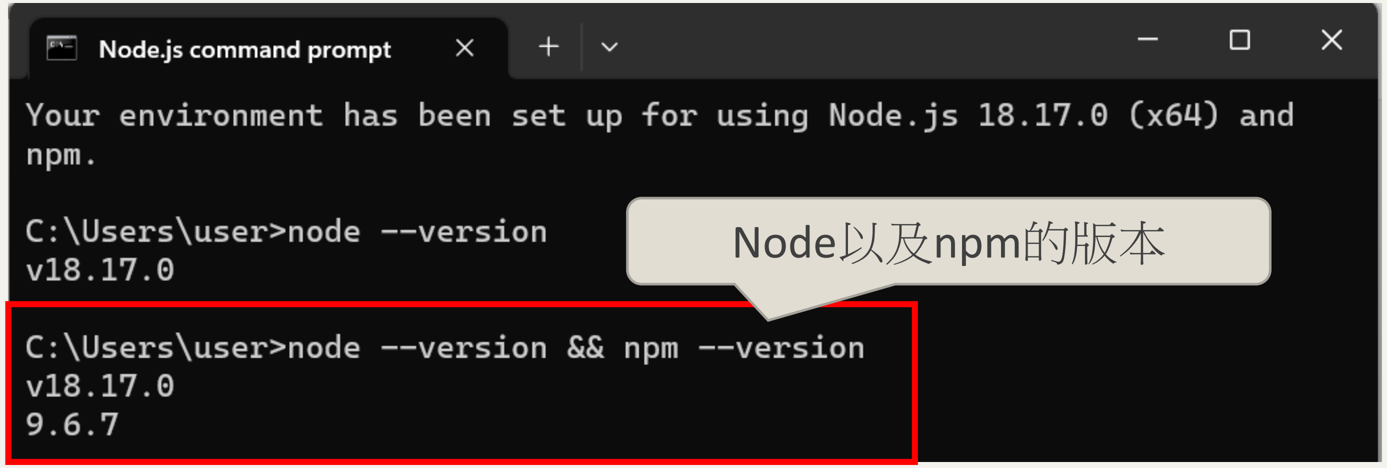
If you are on an RPM-based operating system, including RedHat, Fedora and CentOS, you can use the RPM install script available here.

If you are using Windows, detailed instructions for installing Node-RED can be found here.

1. 因我之前有裝node.js，可在搜尋找到，點選node.js command prompt



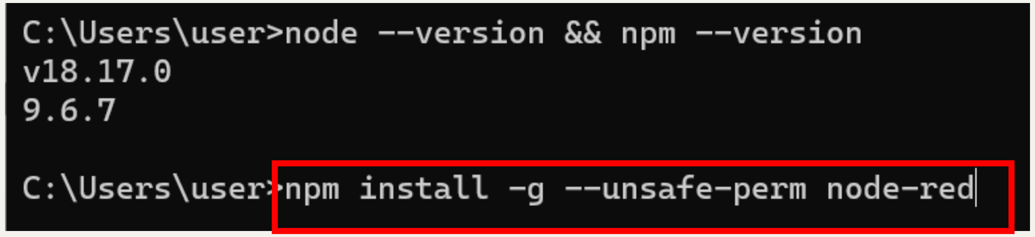
2. Ensure Node.js and npm are installed correctly，輸入指令(node --version && npm --version)



```
Your environment has been set up for using Node.js 18.17.0 (x64) and
npm.

C:\Users\user>node --version
v18.17.0

C:\Users\user>node --version && npm --version
v18.17.0
9.6.7
```

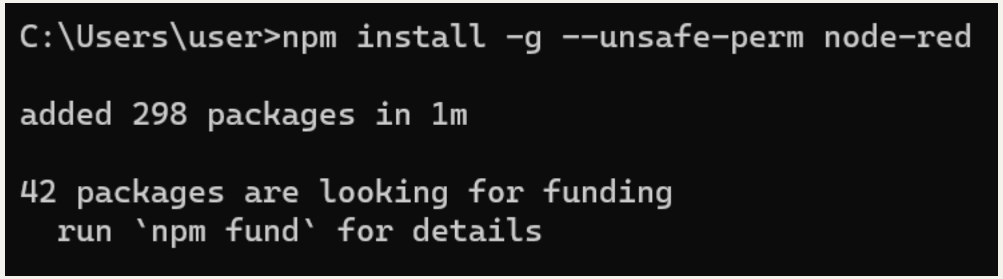
Node以及npm的版本

3. Install Node-RED，輸入指令(npm install -g --unsafe-perm node-red)



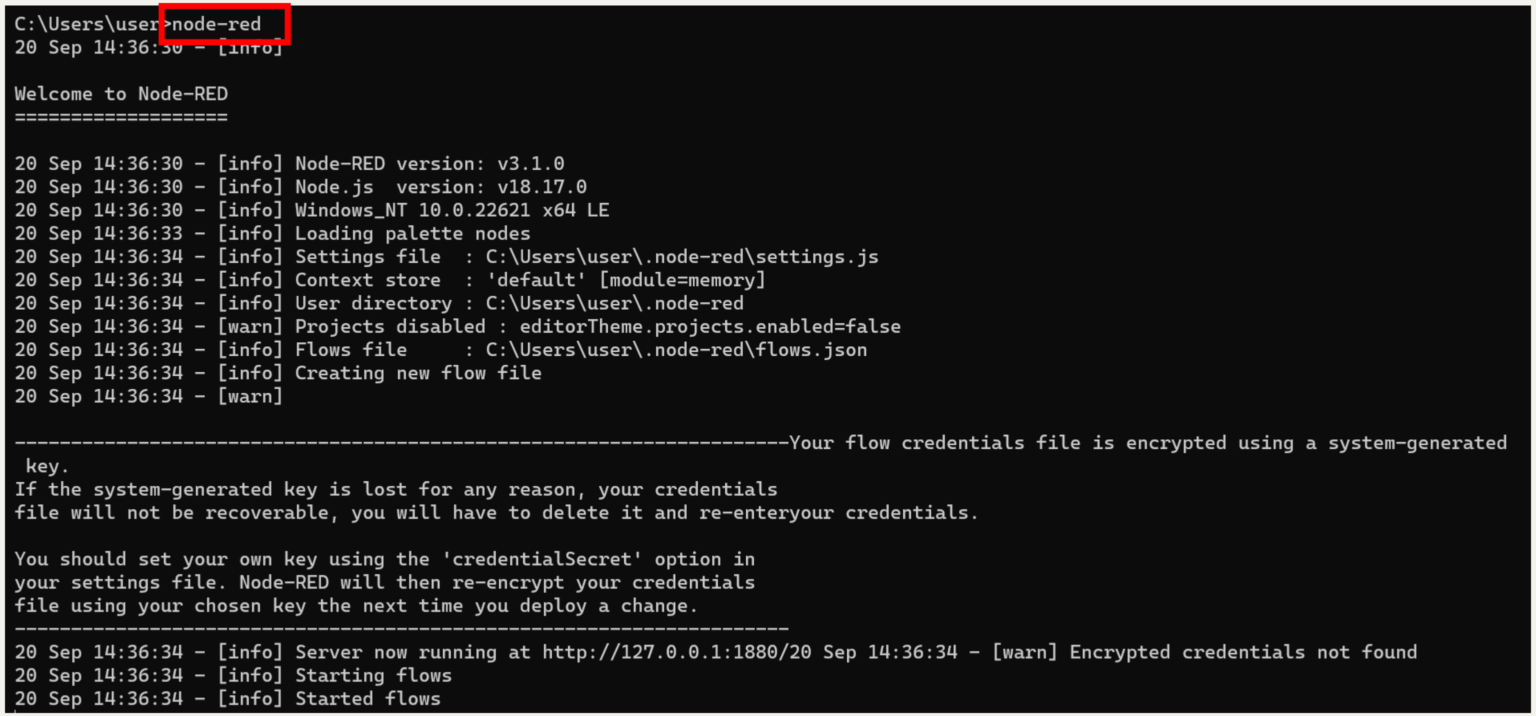
```
C:\Users\user>node --version && npm --version
v18.17.0
9.6.7

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

4. 安裝完成



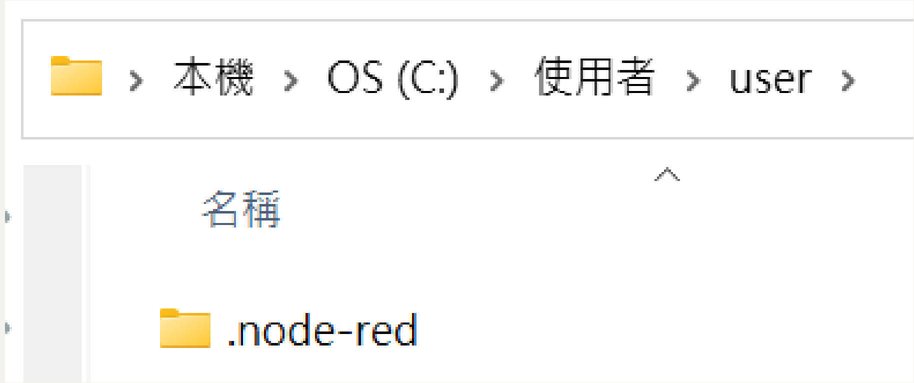
```
C:\Users\user>npm install -g --unsafe-perm node-red

added 298 packages in 1m

42 packages are looking for funding
  run `npm fund` for details
```

5. Run Node-RED輸入指令(node-red)



```
C:\Users\user>node-red
20 Sep 14:36:30 - [info]

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

20 Sep 14:36:30 - [info] Node-RED version: v3.1.0
20 Sep 14:36:30 - [info] Node.js  version: v18.17.0
20 Sep 14:36:30 - [info] Windows_NT 10.0.22621 x64 LE
20 Sep 14:36:33 - [info] Loading palette nodes
20 Sep 14:36:34 - [info] Settings file  : C:\Users\user\.node-red\settings.js
20 Sep 14:36:34 - [info] Context store  : 'default' [module=memory]
20 Sep 14:36:34 - [info] User directory : C:\Users\user\.node-red
20 Sep 14:36:34 - [warn] Projects disabled : editorTheme.projects.enabled=false
20 Sep 14:36:34 - [info] Flows file     : C:\Users\user\.node-red\flows.json
20 Sep 14:36:34 - [info] Creating new flow file
20 Sep 14:36:34 - [warn]

---------------------------------------------------------------------Your flow credentials file is encrypted using a system-generated
 key.
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enteryour credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------
20 Sep 14:36:34 - [info] Server now running at http://127.0.0.1:1880/20 Sep 14:36:34 - [warn] Encrypted credentials not found
20 Sep 14:36:34 - [info] Starting flows
20 Sep 14:36:34 - [info] Started flows
```

6. Check your userDir folder



Exercise 2-2

1. Access the editor

2. Add an Inject node

3. Add a Debug node

4. Wire the two together

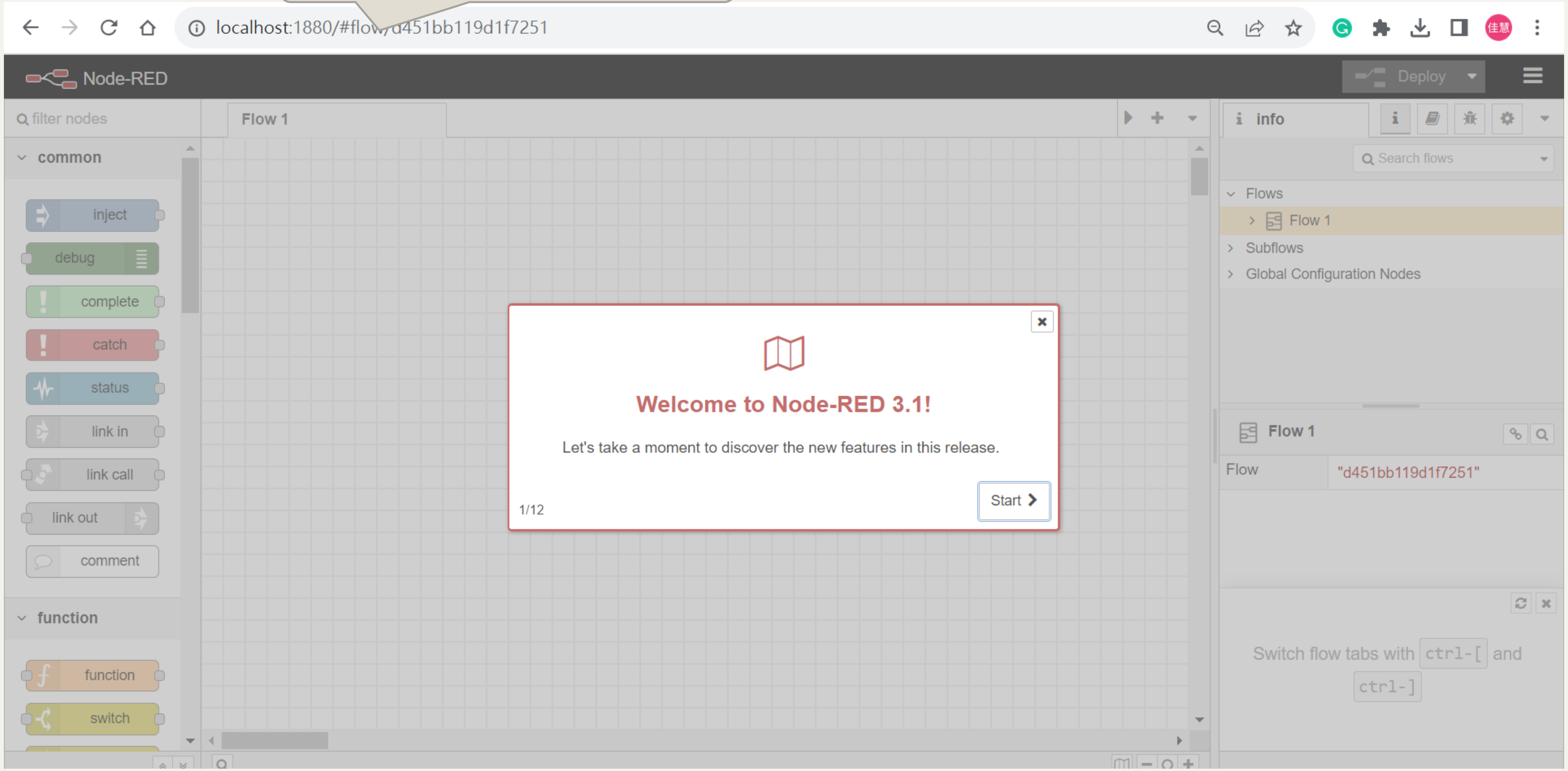
5. Deploy

6. Inject

7. Add a Function node

Open the editor in a web browser

http://localhost:1880



localhost:1880/#flow/d451bb119d1f7251

Welcome to Node-RED 3.1!

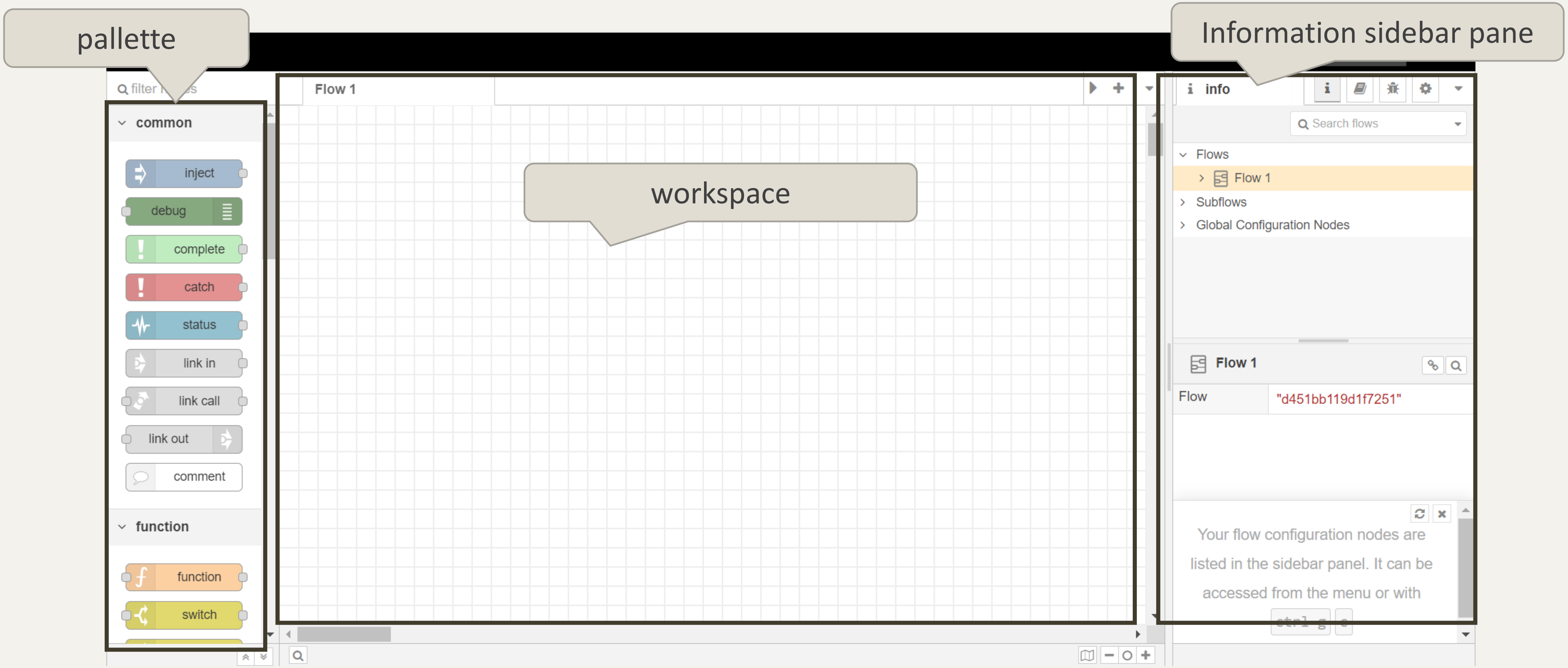
Let's take a moment to discover the new features in this release.

1/12

Start

Flow	"d451bb119d1f7251"

Switch flow tabs with `ctrl-[` and `ctrl-]`



pallette

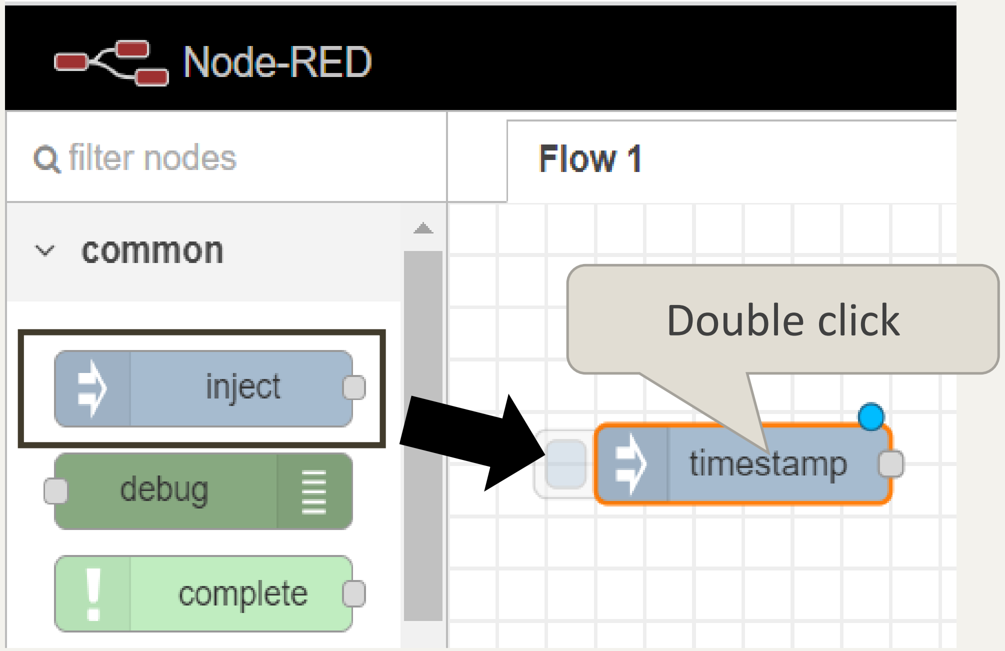
workspace

Information sidebar pane

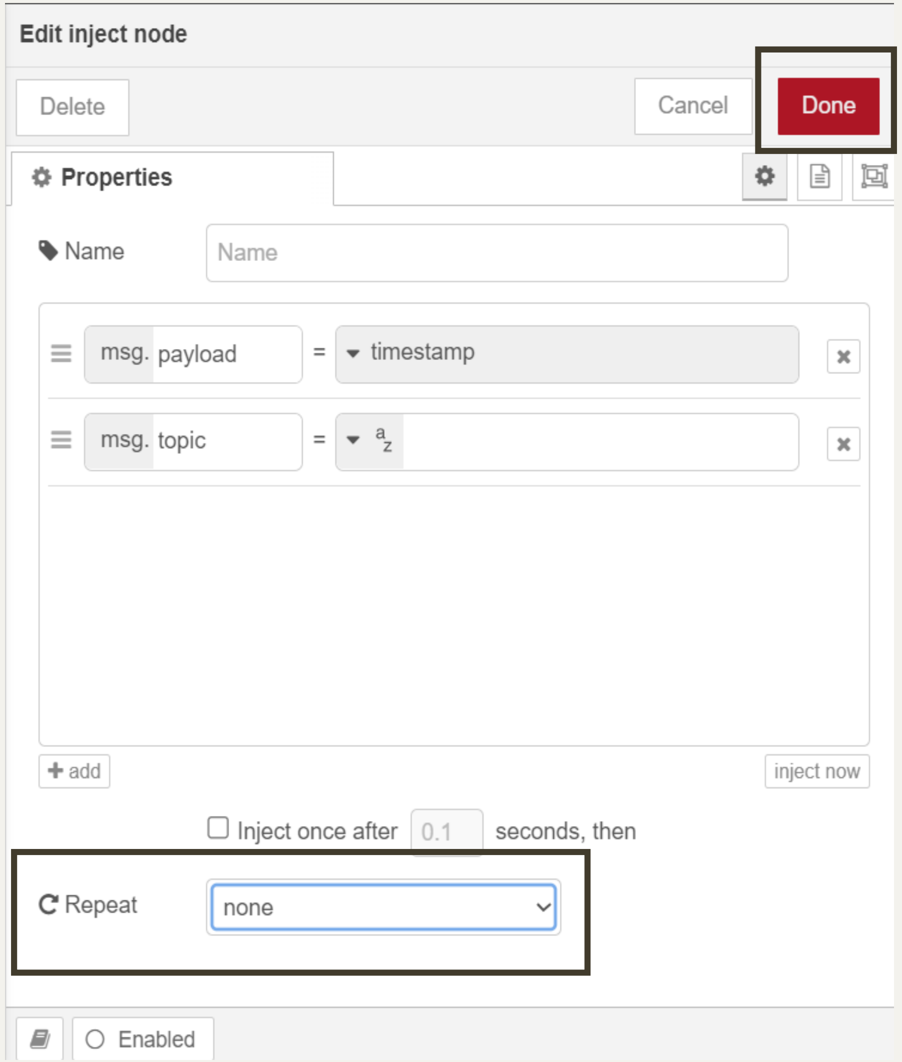
Flow "d451bb119d1f7251"

Your flow configuration nodes are listed in the sidebar panel. It can be accessed from the menu or with ctrl g c

Add an Inject node



Double click



Name: Name

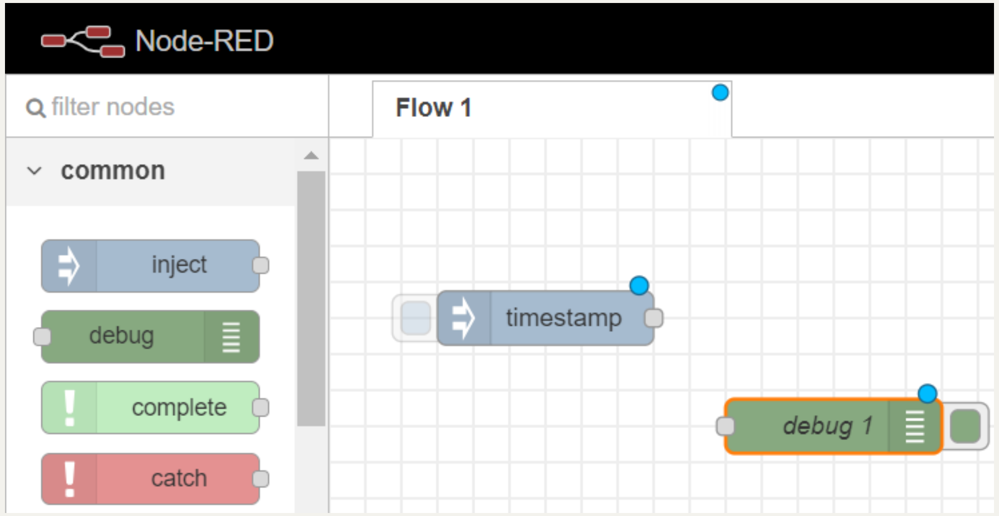
msg. payload = timestamp

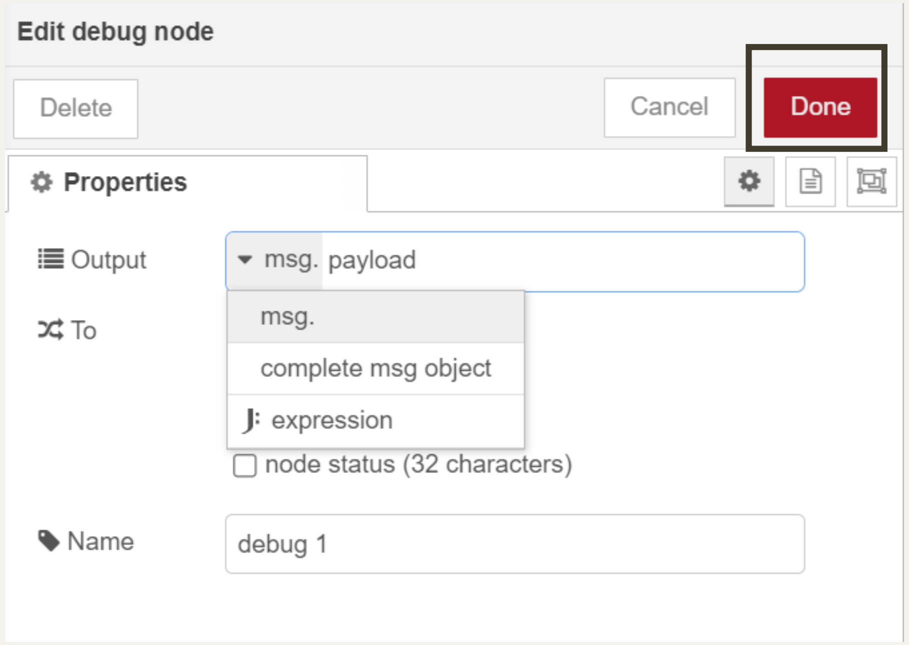
msg. topic =

☐ Inject once after 0.1 seconds, then

Repeat: none

Add a Debug node





Edit debug node

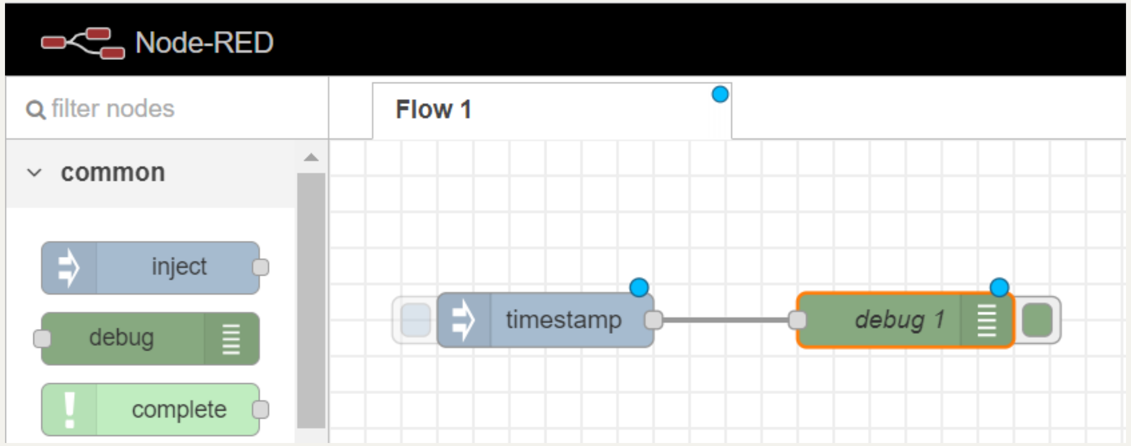
Output: msg. payload

To

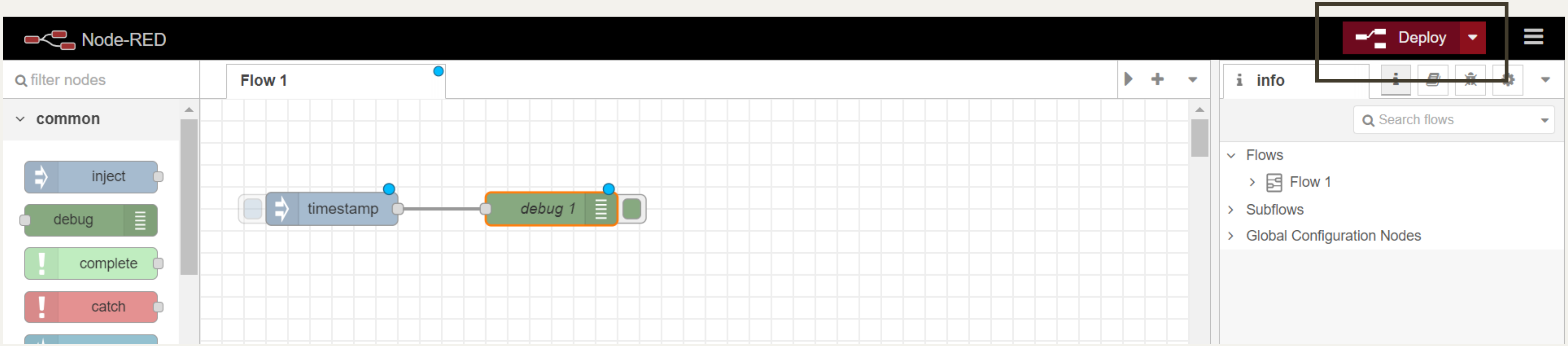
☐ node status (32 characters)

Name: debug 1

Wire the two together

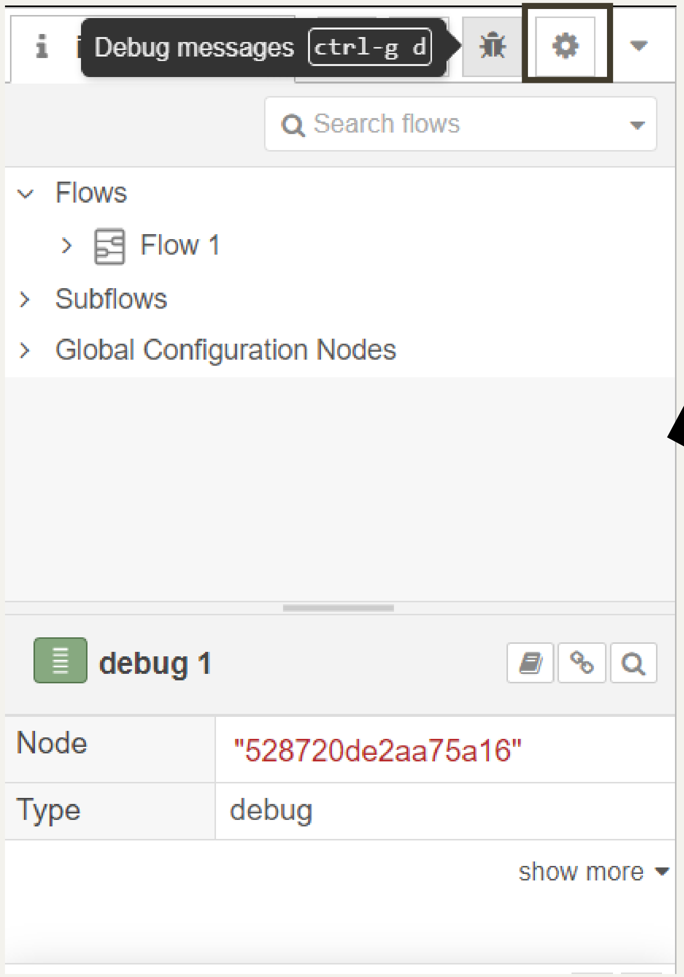


Deploy

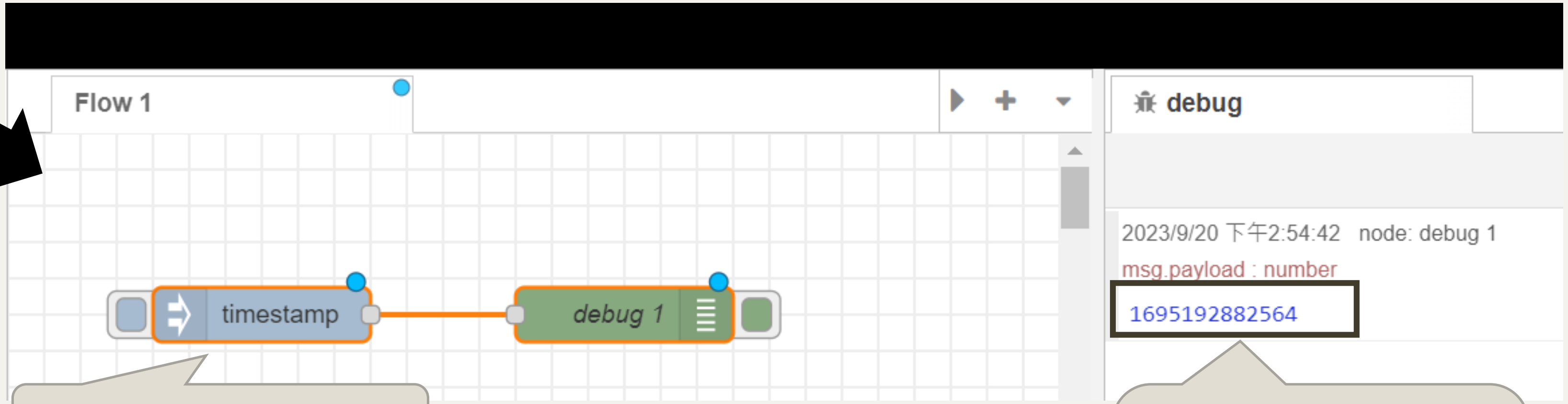

Node-RED
Deploy
filter nodes
common
inject
debug
complete
catch
Flow 1
timestamp
debug 1
info
Search flows
Flows
Flow 1
Subflows
Global Configuration Nodes

Inject

1. Debug messages



Node	"528720de2aa75a16"
Type	debug

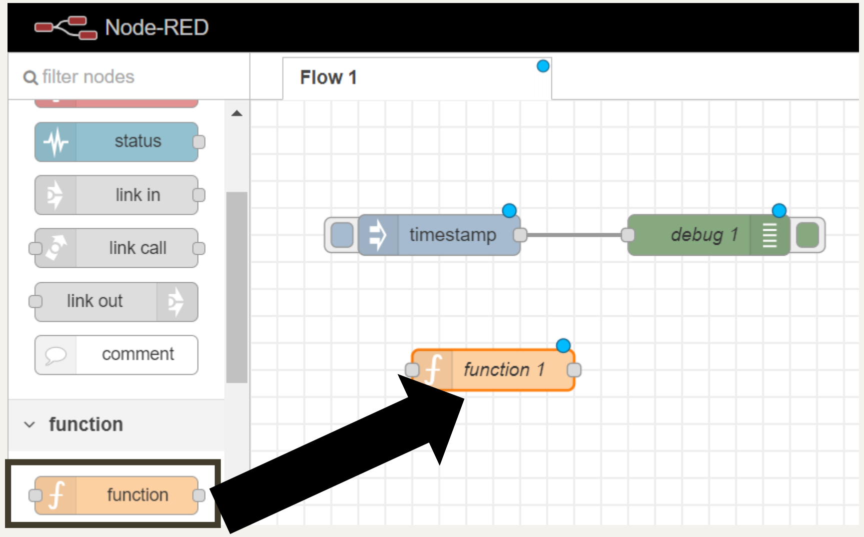


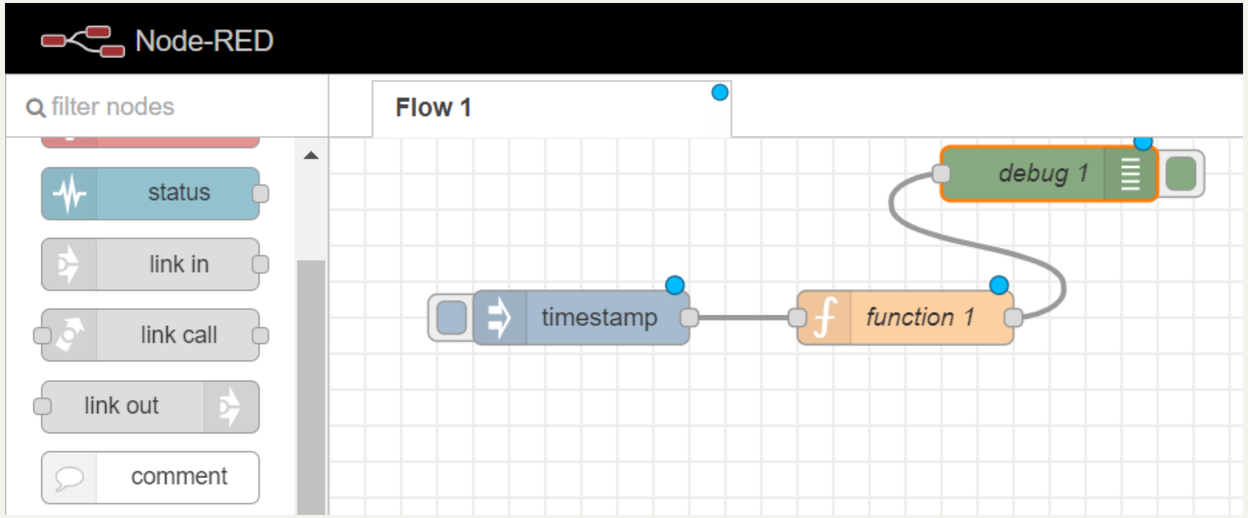
```
2023/9/20 下午2:54:42   node: debug 1
msg.payload : number
1695192882564
```

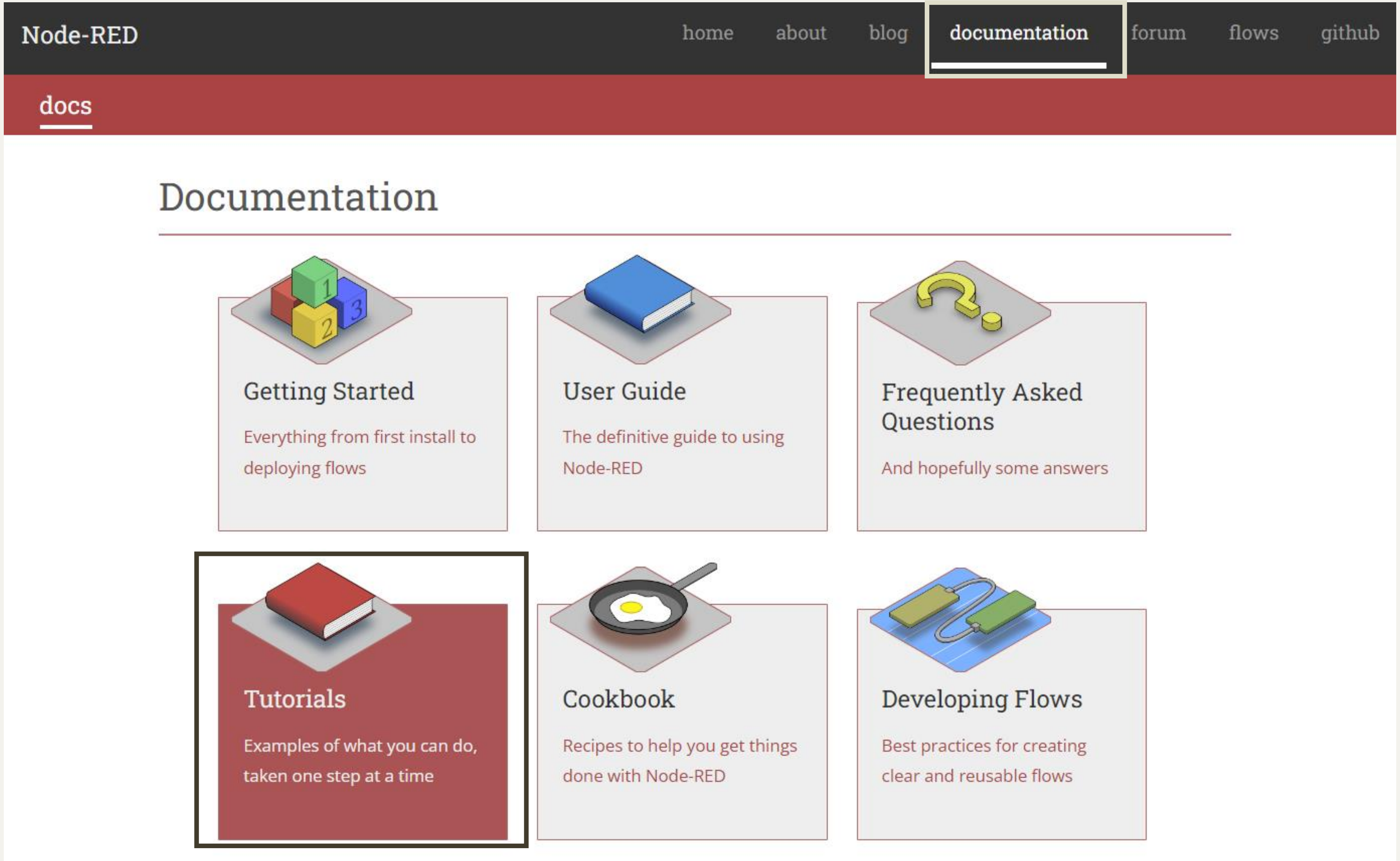
2. Click

The milliseconds since January 1st, 1970

Add a Function node







Documentation

Getting Started

Everything from first install to deploying flows

User Guide

The definitive guide to using Node-RED

Frequently Asked Questions

And hopefully some answers

Tutorials

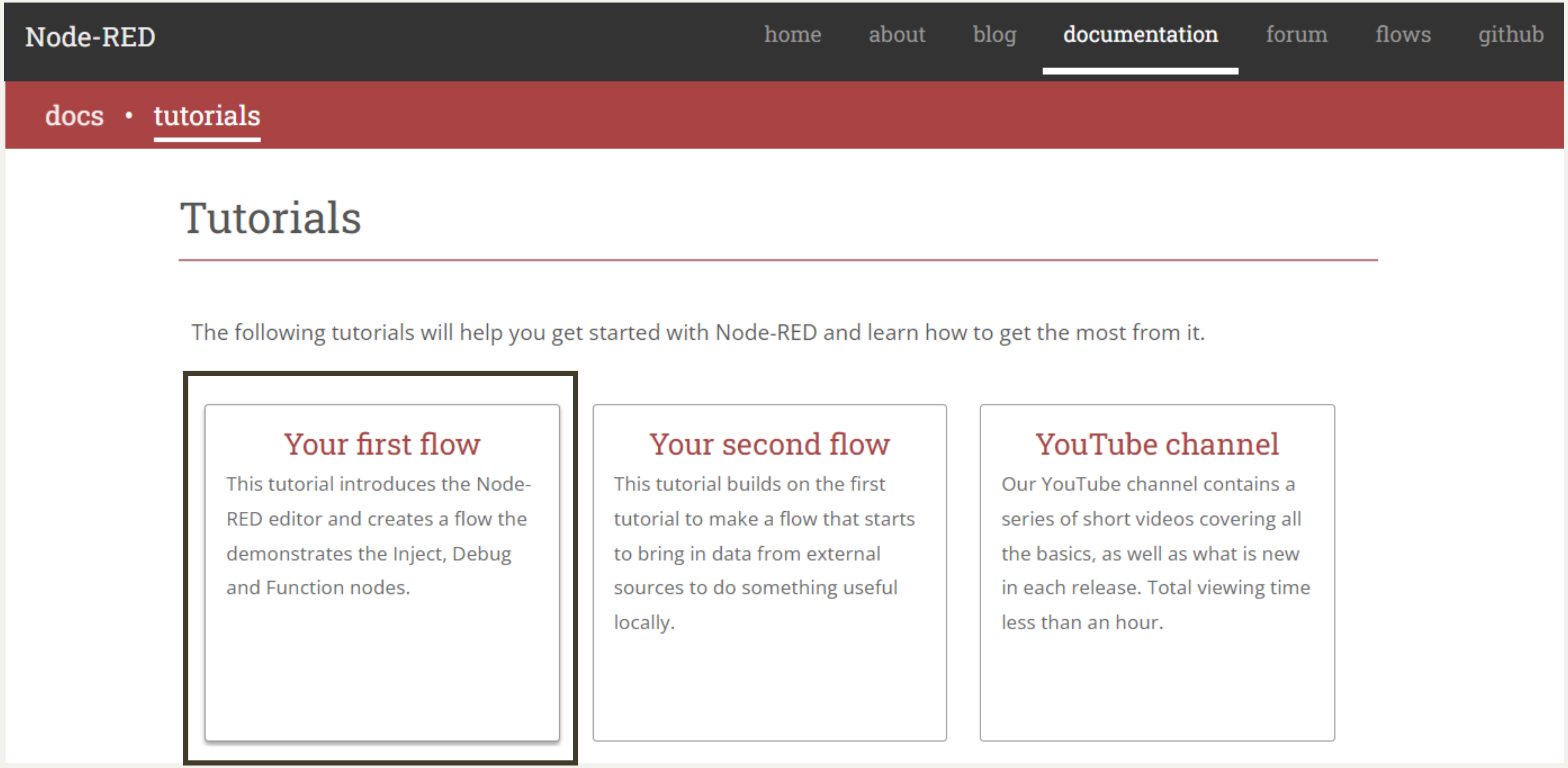
Examples of what you can do, taken one step at a time

Cookbook

Recipes to help you get things done with Node-RED

Developing Flows

Best practices for creating clear and reusable flows



Tutorials

The following tutorials will help you get started with Node-RED and learn how to get the most from it.

Your first flow

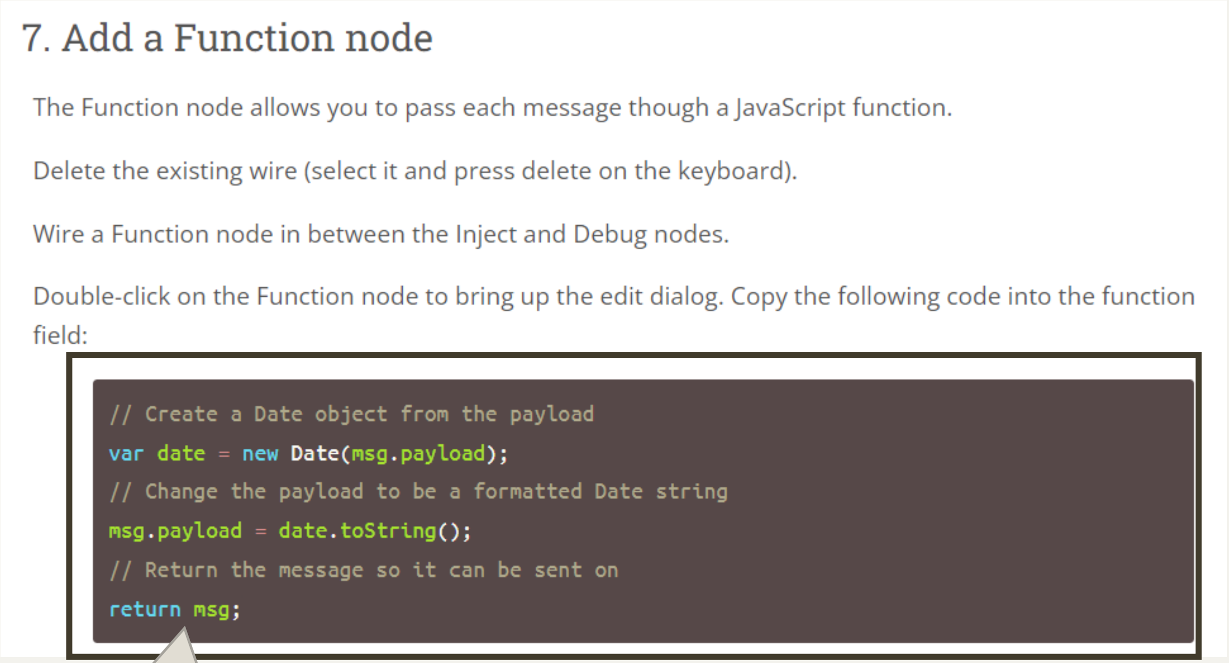
This tutorial introduces the Node-RED editor and creates a flow the demonstrates the Inject, Debug and Function nodes.

Your second flow

This tutorial builds on the first tutorial to make a flow that starts to bring in data from external sources to do something useful locally.

YouTube channel

Our YouTube channel contains a series of short videos covering all the basics, as well as what is new in each release. Total viewing time less than an hour.



7. Add a Function node

The Function node allows you to pass each message though a JavaScript function.

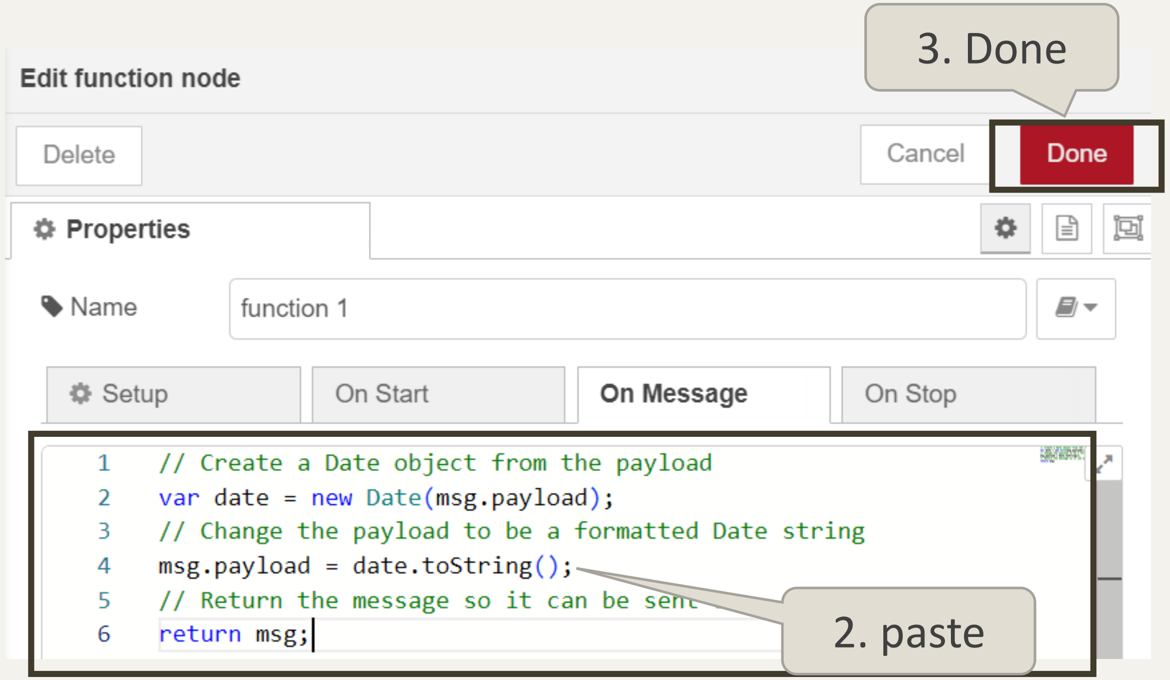
Delete the existing wire (select it and press delete on the keyboard).

Wire a Function node in between the Inject and Debug nodes.

Double-click on the Function node to bring up the edit dialog. Copy the following code into the function field:

```
// Create a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent on
return msg;
```

copy



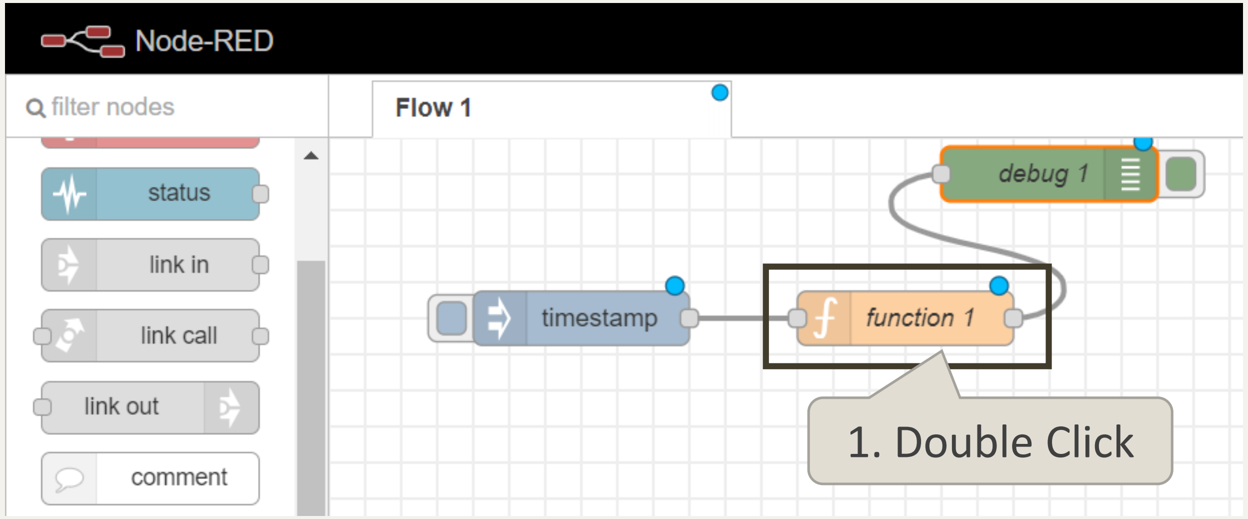
3. Done

Edit function node

Name: function 1

```
// Create a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent
return msg;
```

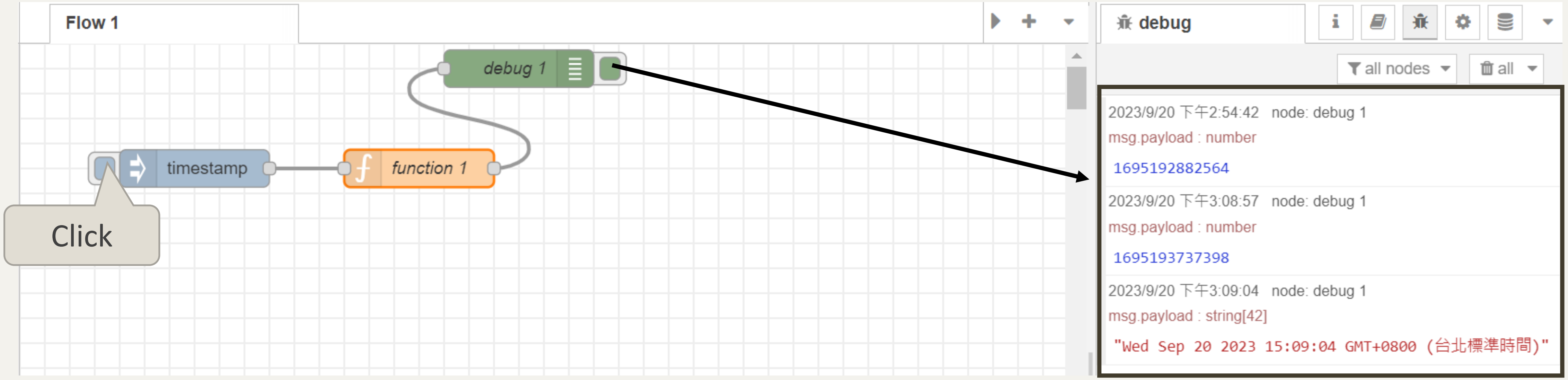
2. paste



1. Double Click

4. Deploy

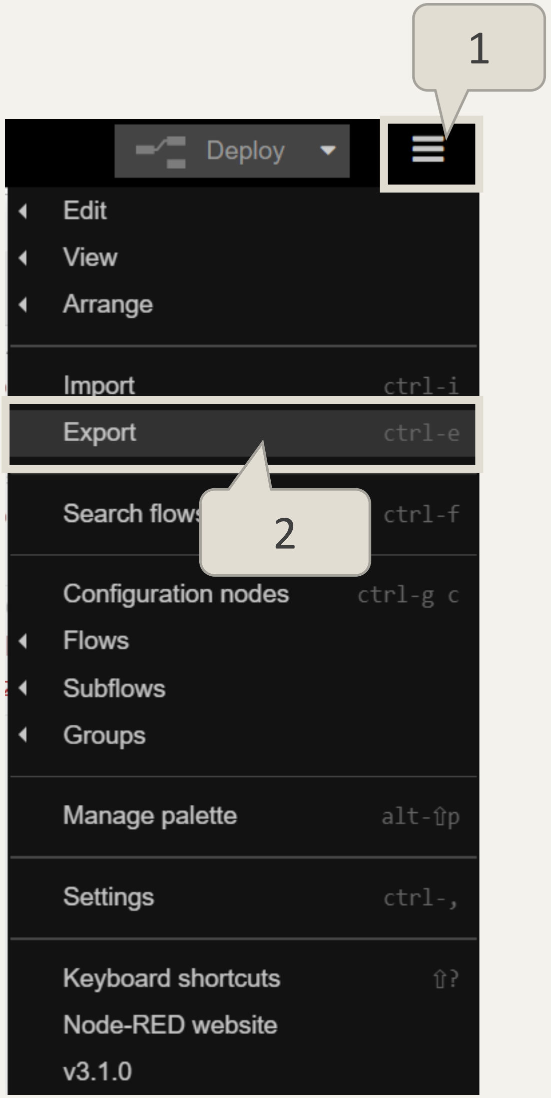
Trigger a flow

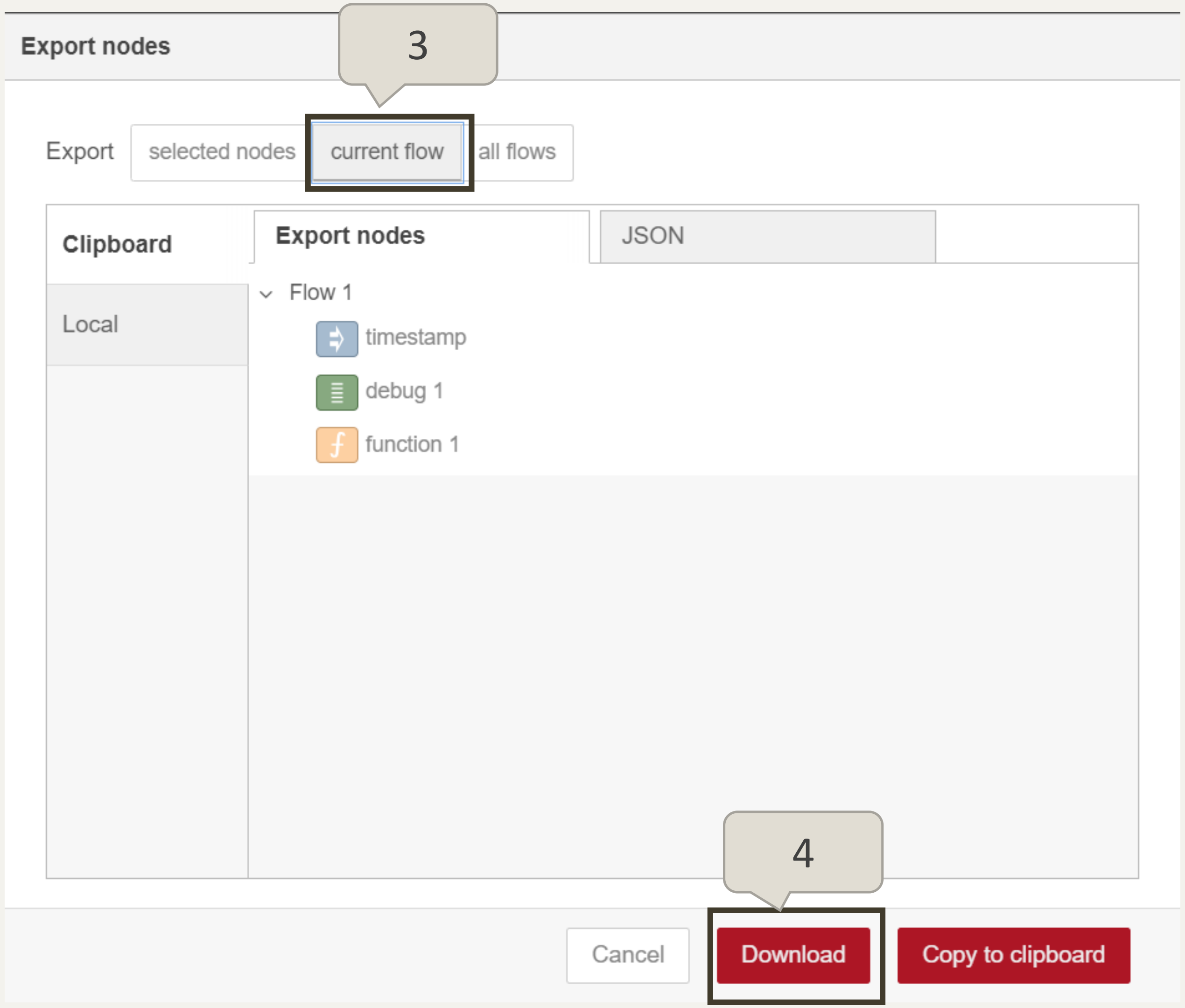


```
2023/9/20 下午2:54:42  node: debug 1
msg.payload : number
 1695192882564
2023/9/20 下午3:08:57  node: debug 1
msg.payload : number
 1695193737398
2023/9/20 下午3:09:04  node: debug 1
msg.payload : string[42]
 "Wed Sep 20 2023 15:09:04 GMT+0800 (台北標準時間)"
```

Exercise 2-3

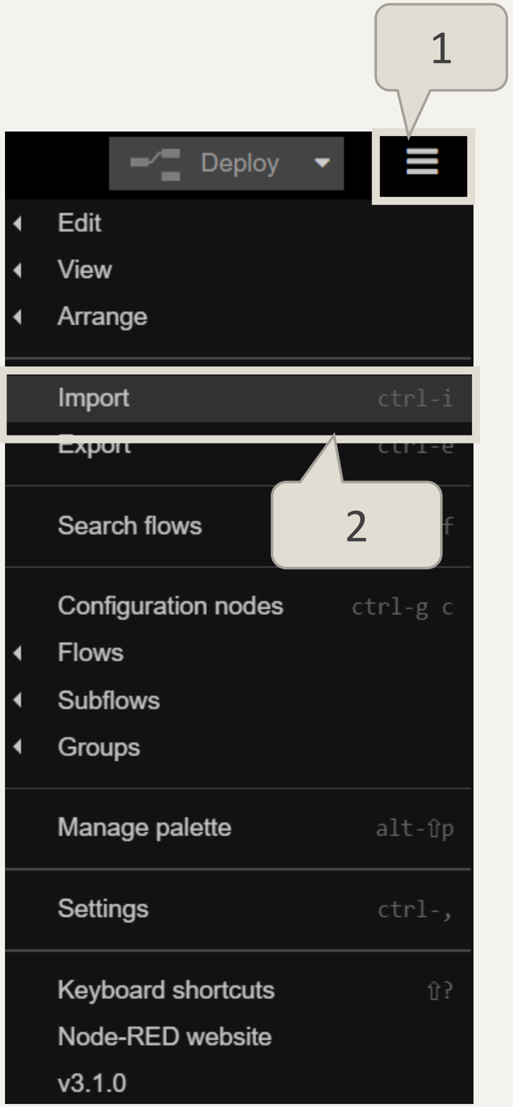
Export

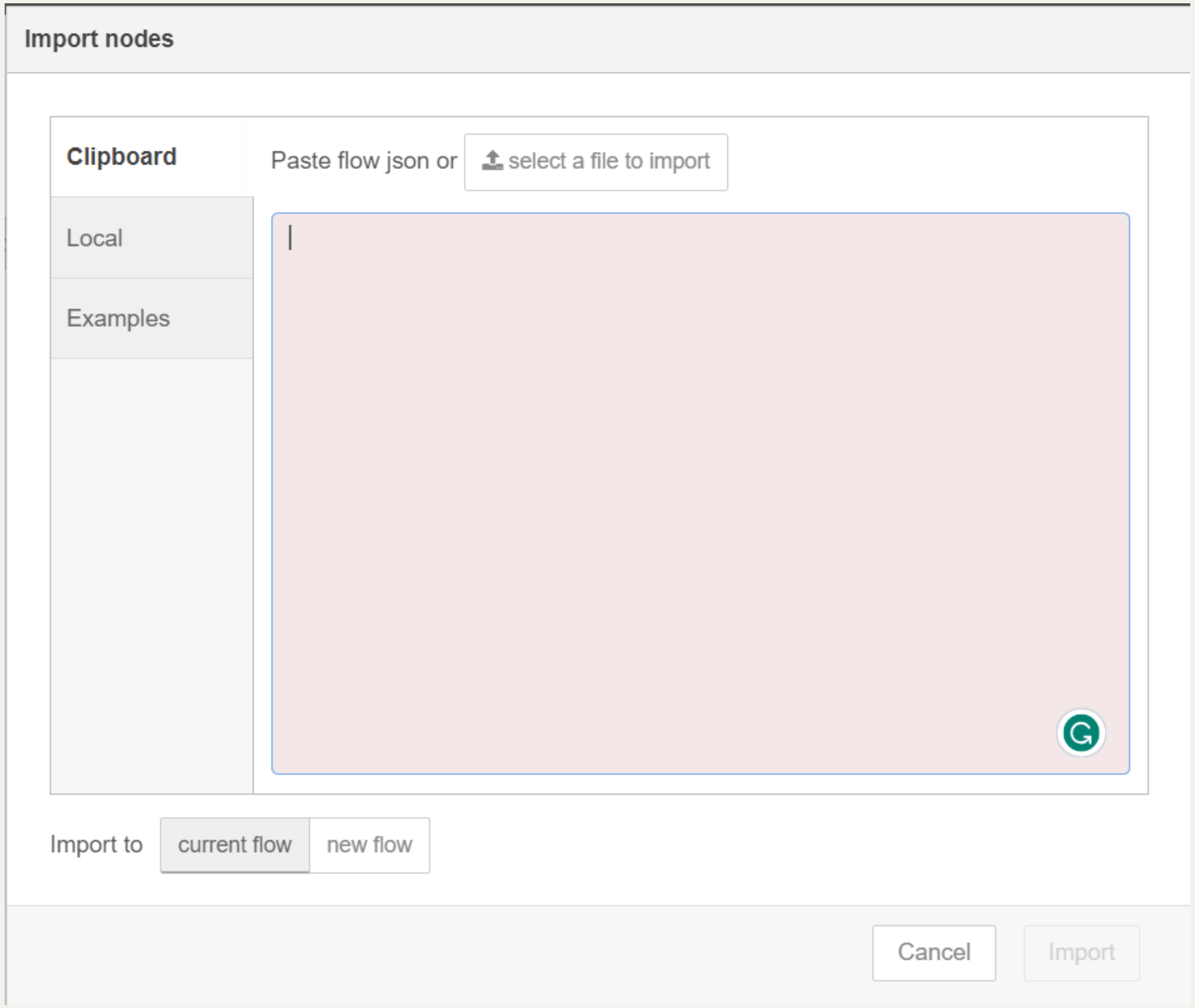

1
Deploy
Edit
View
Arrange
Import ctrl-i
Export ctrl-e
2
Search flows ctrl-f
Configuration nodes ctrl-g c
Flows
Subflows
Groups
Manage palette alt-⇧p
Settings ctrl-,
Keyboard shortcuts ⇧?
Node-RED website
v3.1.0

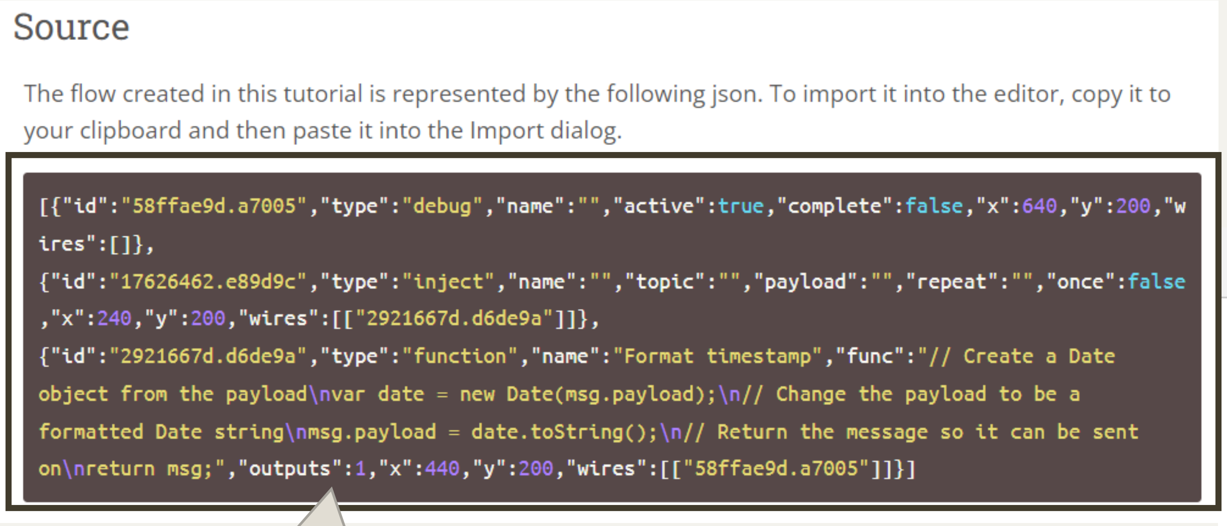

Export nodes
3
Export
selected nodes
current flow
all flows
Clipboard
Local
Export nodes
JSON
Flow 1
timestamp
debug 1
function 1
4
Cancel
Download
Copy to clipboard

Homework 2-1

Import flows.json to FlowFuse


1
Deploy
Edit
View
Arrange
Import ctrl-i
Export ctrl-e
2
Search flows
Configuration nodes ctrl-g c
Flows
Subflows
Groups
Manage palette alt-⇧p
Settings ctrl-,
Keyboard shortcuts ⇧?
Node-RED website
v3.1.0


Import nodes
Clipboard
Local
Examples
Paste flow json or
select a file to import
Import to
current flow
new flow
Cancel
Import

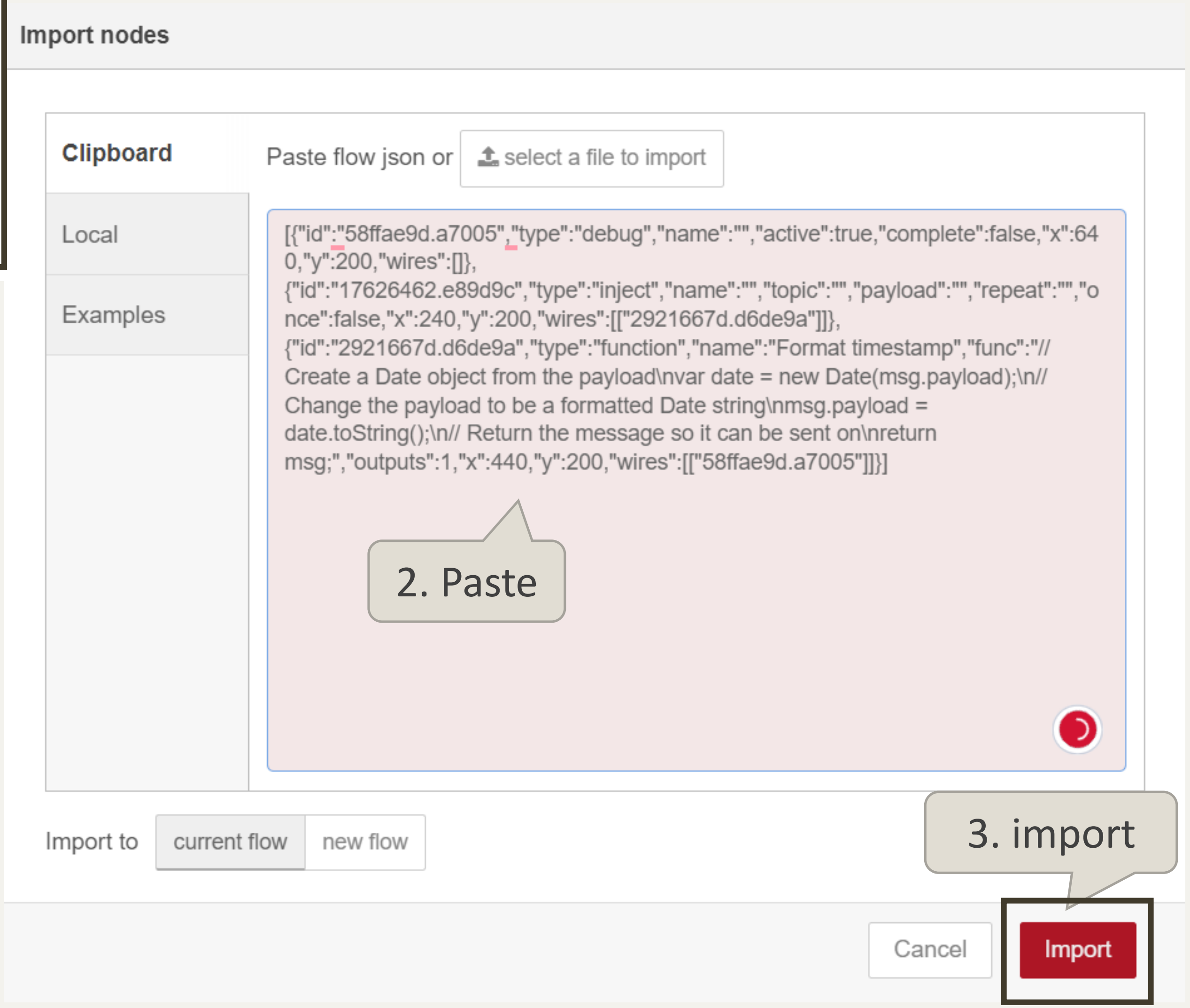


Source

The flow created in this tutorial is represented by the following json. To import it into the editor, copy it to your clipboard and then paste it into the Import dialog.

```
[{"id":"58ffae9d.a7005","type":"debug","name":"","active":true,"complete":false,"x":640,"y":200,"w
ires":[]},
{"id":"17626462.e89d9c","type":"inject","name":"","topic":"","payload":"","repeat":"","once":false
,"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"]]}]
```

1. copy



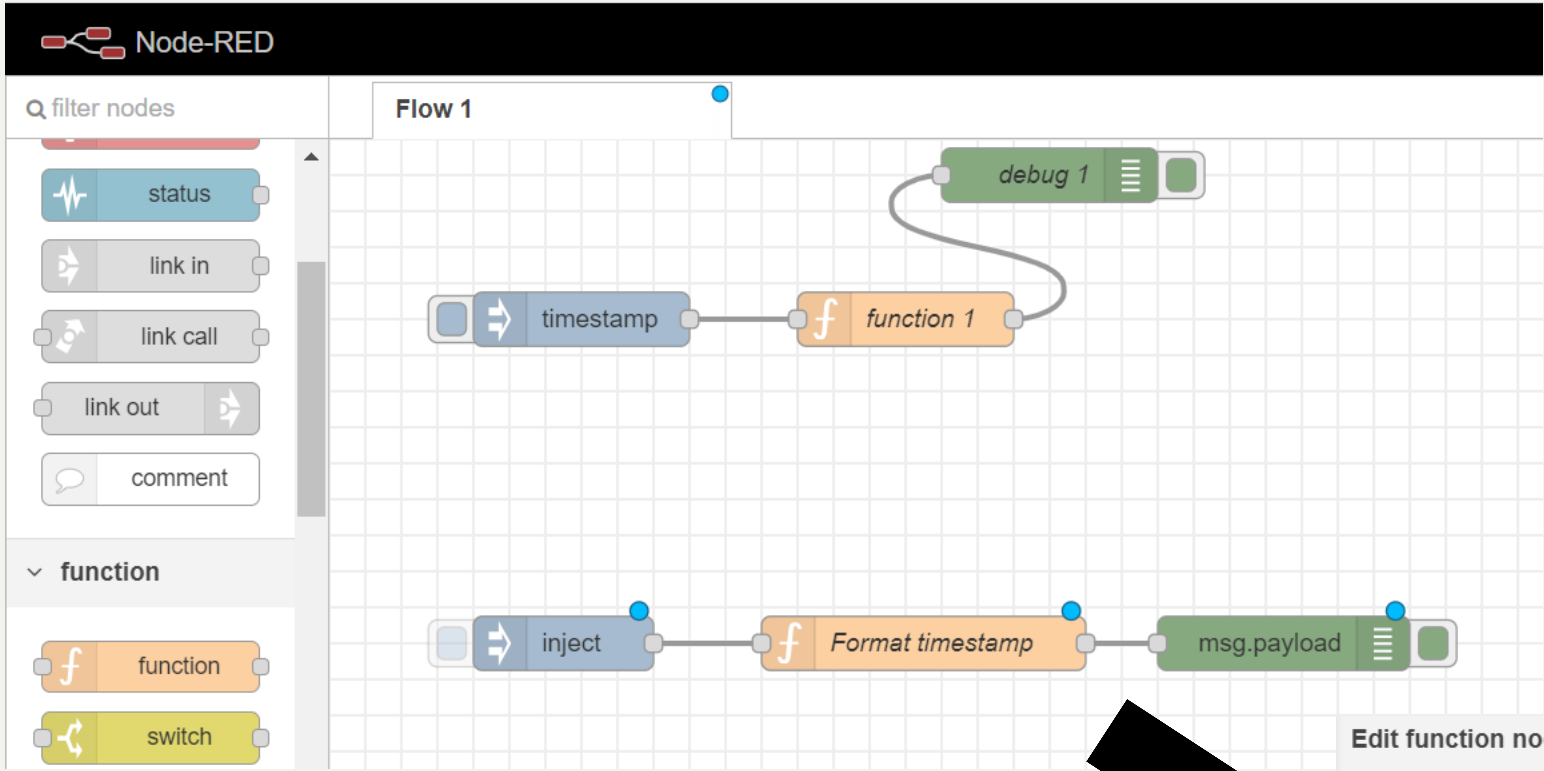
Import nodes

[{"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":false,"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"]]}]

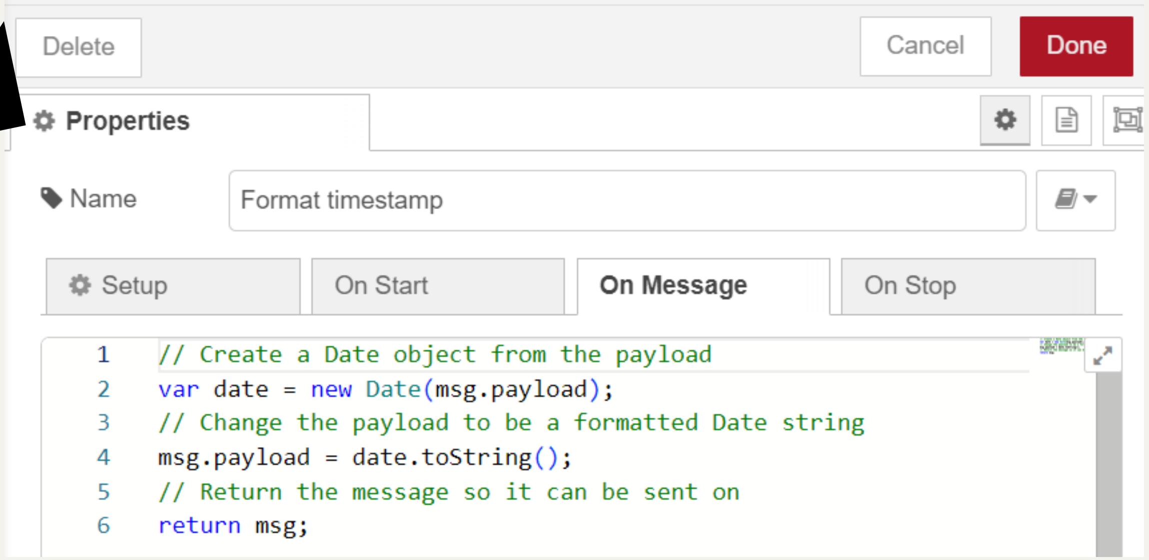
2. Paste

3. import

https://nodered.org/docs/tutorials/first-flow



Edit function node



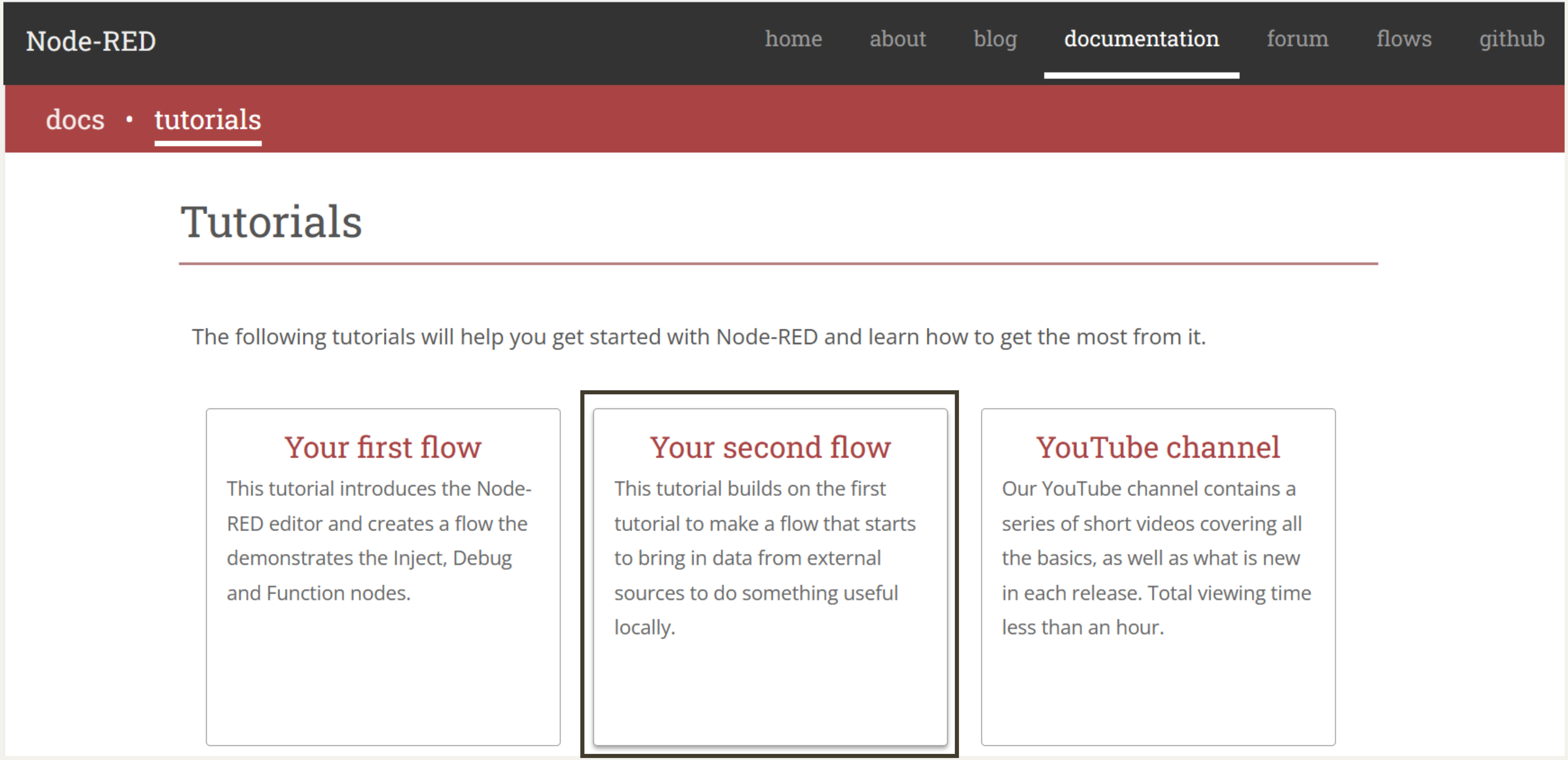
Name: Format timestamp

```
// Create a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent on
return msg;
```

可檢查與自己拉的方塊裡設的功能是否正確

Homework 2-2

Creating your second flow



Tutorials

The following tutorials will help you get started with Node-RED and learn how to get the most from it.

Your first flow

This tutorial introduces the Node-RED editor and creates a flow the demonstrates the Inject, Debug and Function nodes.

Your second flow

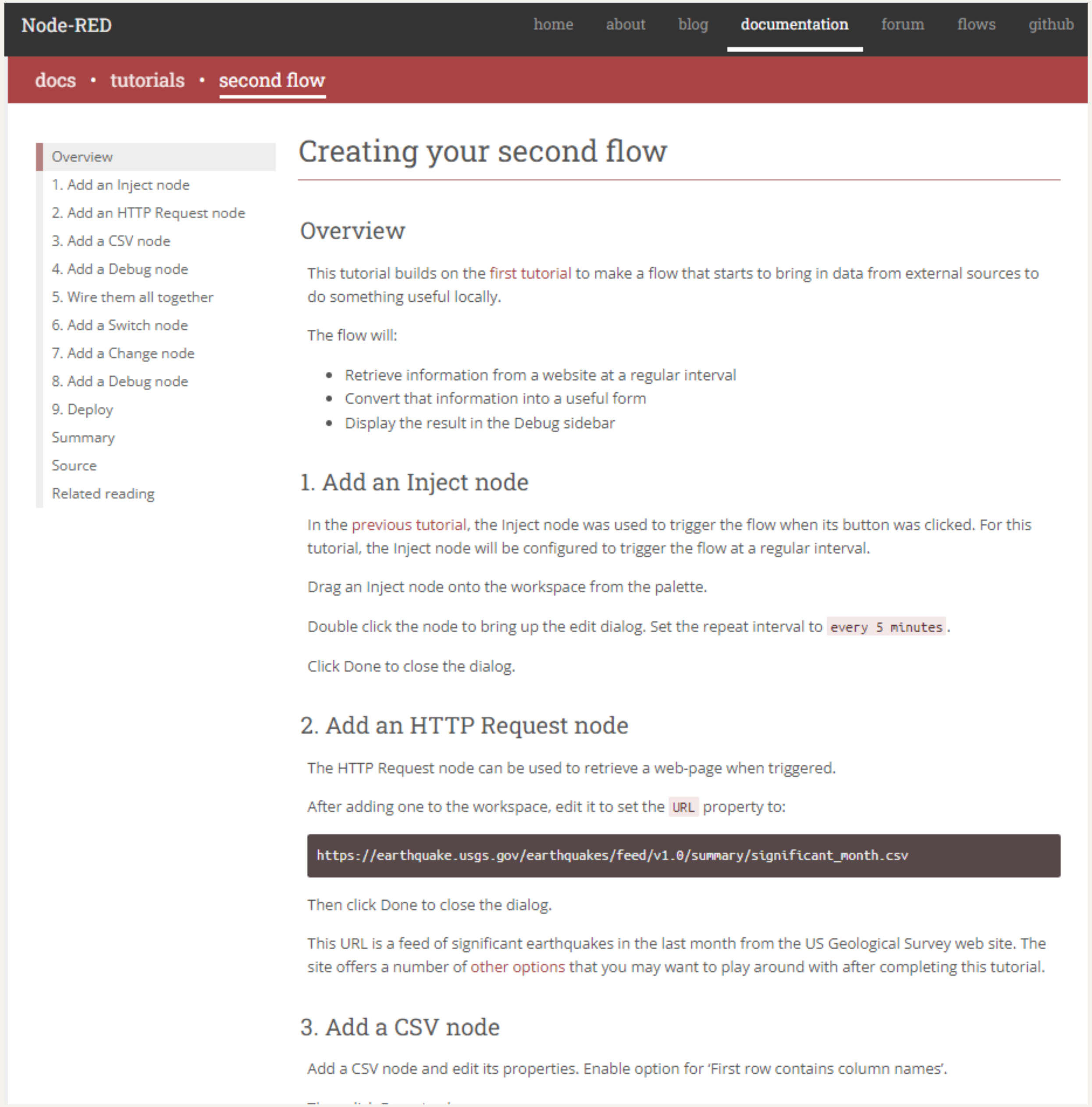
This tutorial builds on the first tutorial to make a flow that starts to bring in data from external sources to do something useful locally.

YouTube channel

Our YouTube channel contains a series of short videos covering all the basics, as well as what is new in each release. Total viewing time less than an hour.

Follow this page

https://nodered.org/docs/tutorials/second-flow



Creating your second flow

Overview

This tutorial builds on the first tutorial to make a flow that starts to bring in data from external sources to do something useful locally.

The flow will:

- Retrieve information from a website at a regular interval
- Convert that information into a useful form
- Display the result in the Debug sidebar

1. Add an Inject node

In the previous tutorial, the Inject node was used to trigger the flow when its button was clicked. For this tutorial, the Inject node will be configured to trigger the flow at a regular interval.

Drag an Inject node onto the workspace from the palette.

Double click the node to bring up the edit dialog. Set the repeat interval to `every 5 minutes`.

Click Done to close the dialog.

2. Add an HTTP Request node

The HTTP Request node can be used to retrieve a web-page when triggered.

After adding one to the workspace, edit it to set the `URL` property to:

```
https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_month.csv
```

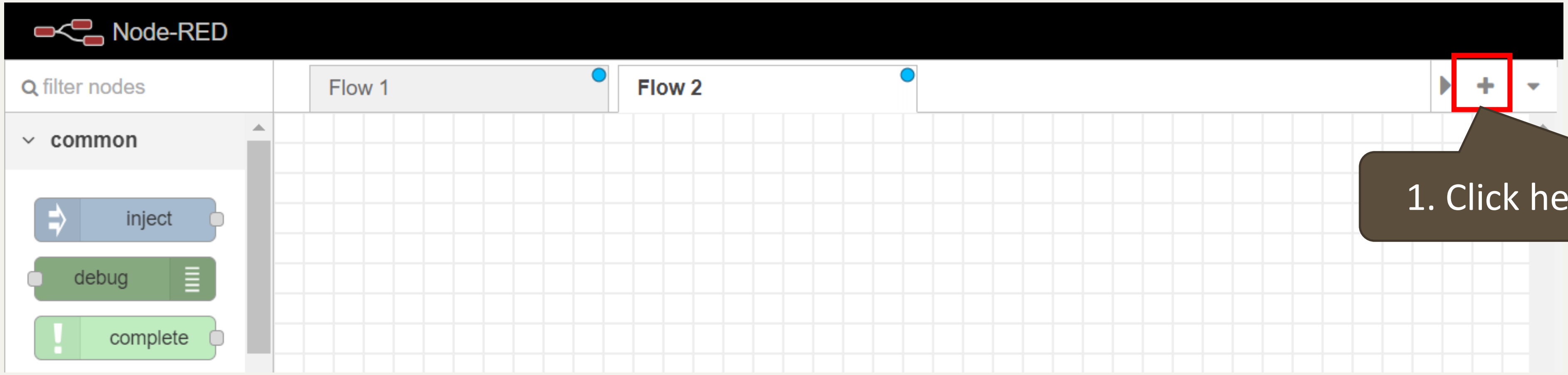
Then click Done to close the dialog.

This URL is a feed of significant earthquakes in the last month from the US Geological Survey web site. The site offers a number of other options that you may want to play around with after completing this tutorial.

3. Add a CSV node

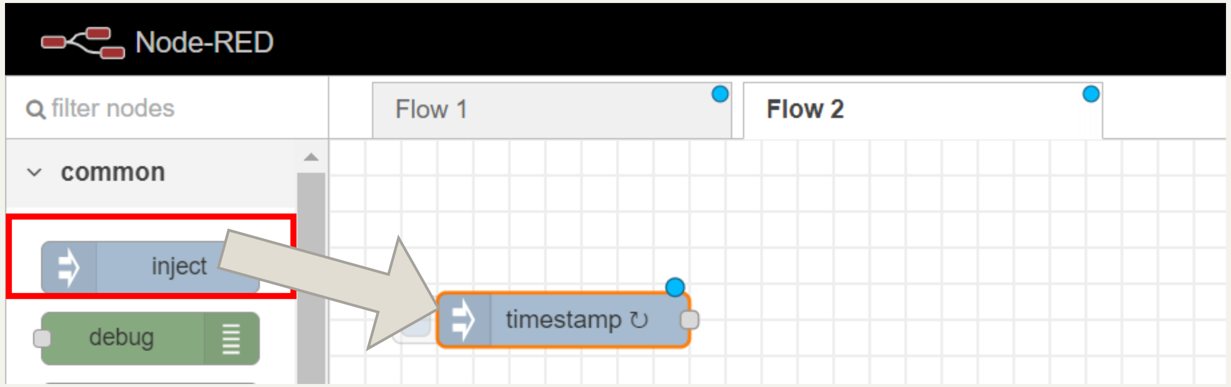
Add a CSV node and edit its properties. Enable option for 'First row contains column names'.

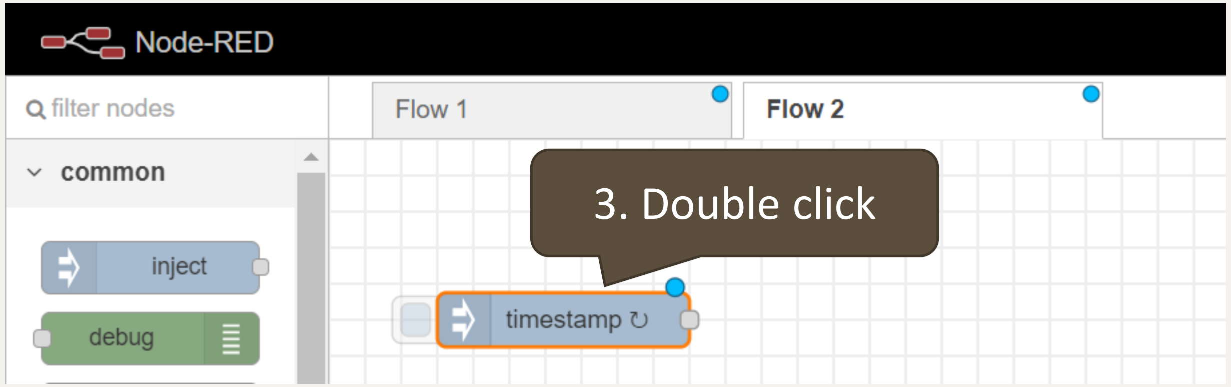
1. Add an Inject node



1. Click here to add a new Flow

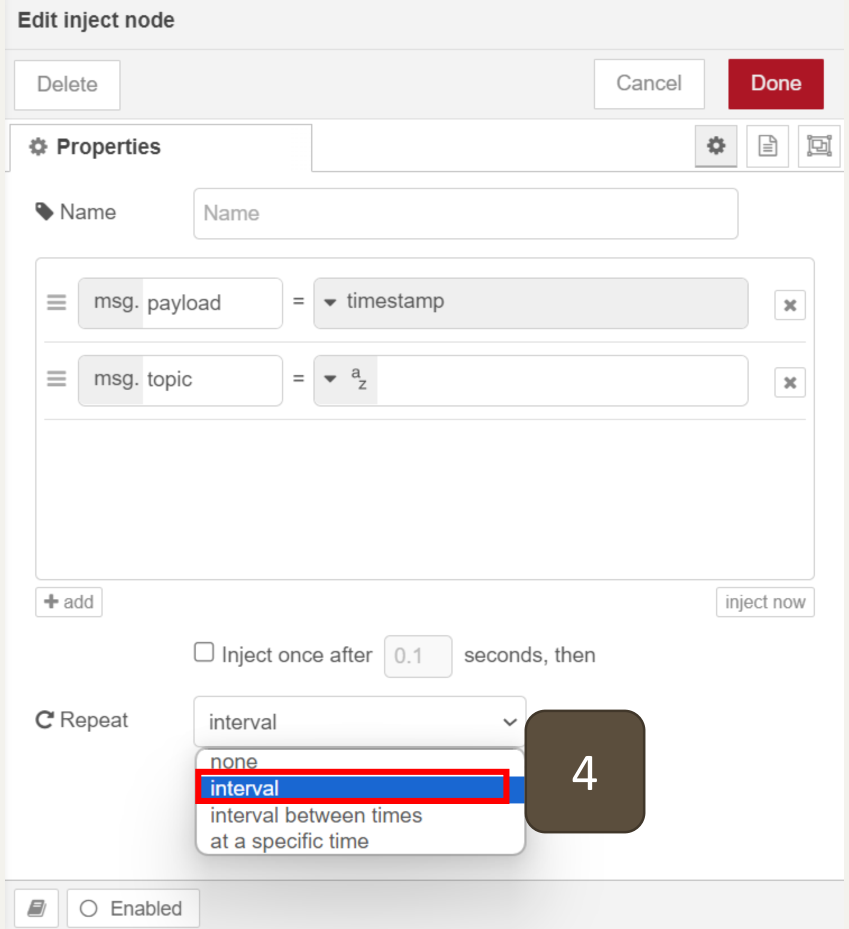
2. Add an Inject node





3. Double click

timestamp ↻



Edit inject node

Name: Name

msg. payload = timestamp

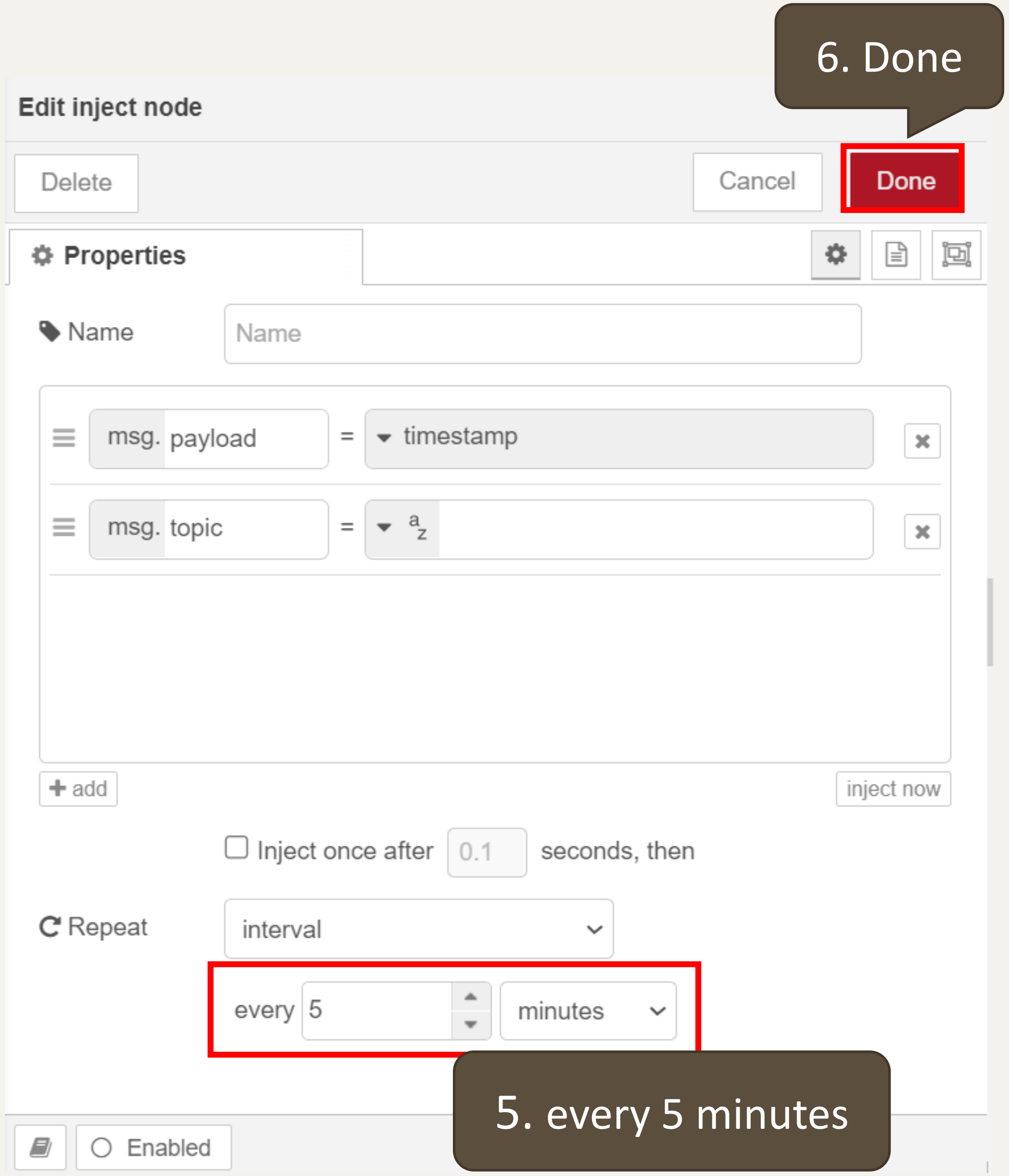
msg. topic = ^a_z

□ Inject once after 0.1 seconds, then

Repeat: interval

4

○ Enabled



6. Done

Edit inject node

Name: Name

msg. payload = timestamp

msg. topic = ^a_z

□ Inject once after 0.1 seconds, then

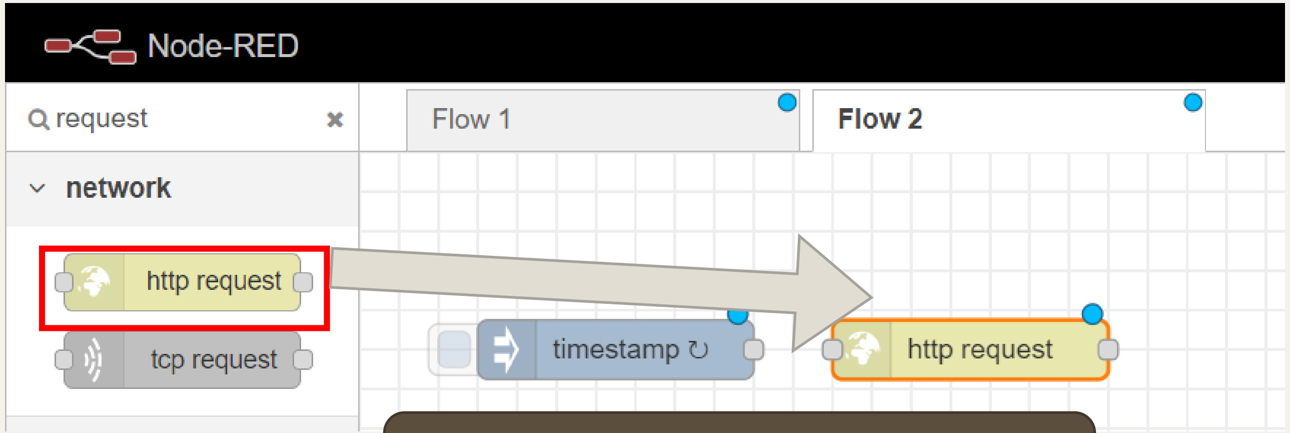
Repeat: interval

every 5 minutes

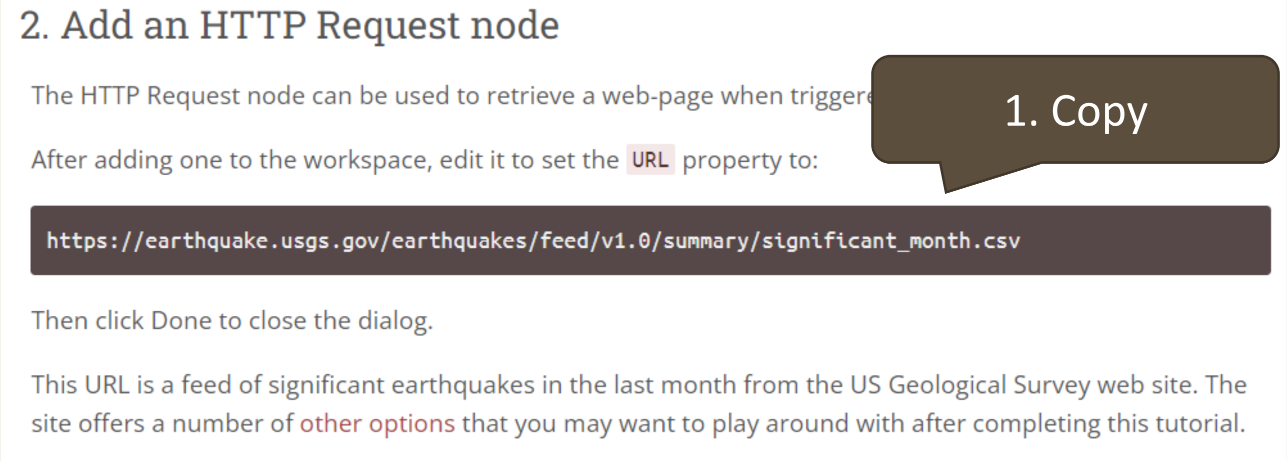
5. every 5 minutes

○ Enabled

2. Add an HTTP Request node



Add an HTTP Request node



2. Add an HTTP Request node

The HTTP Request node can be used to retrieve a web-page when triggere

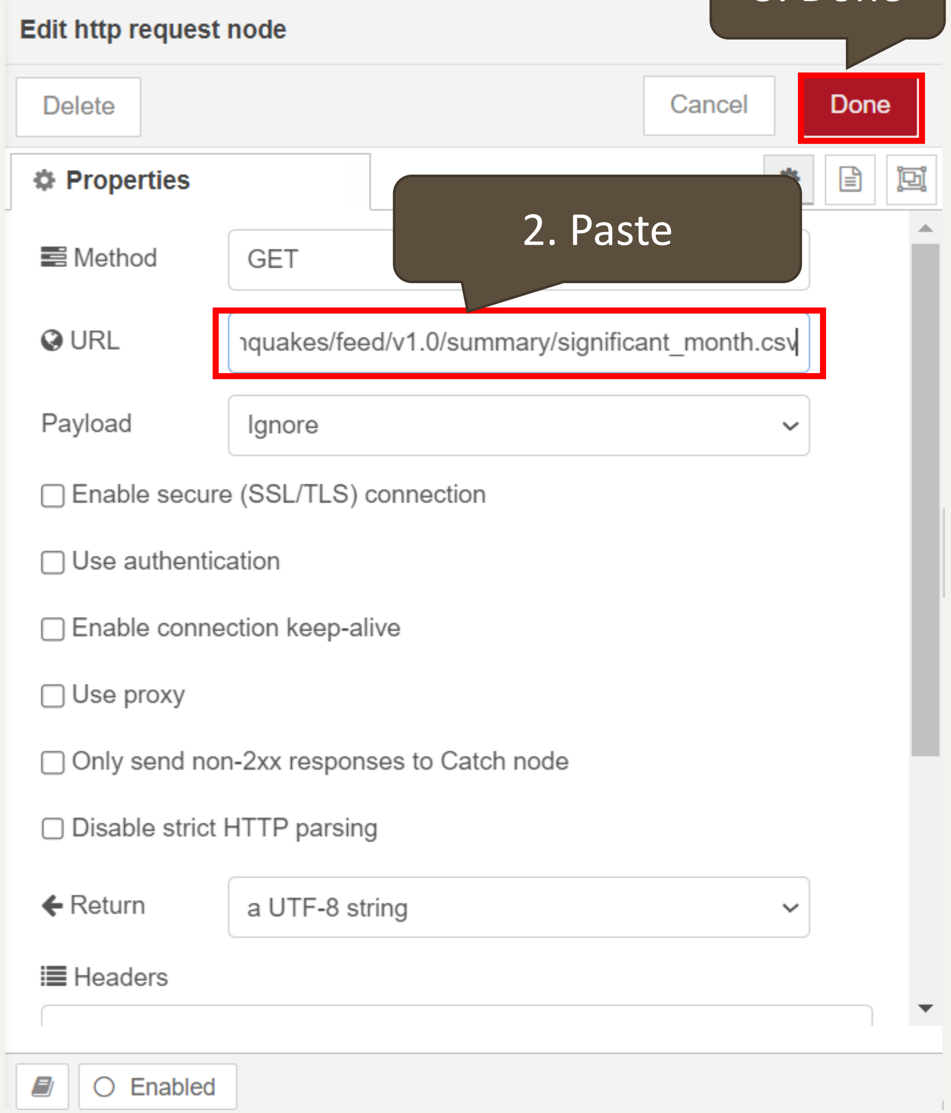
After adding one to the workspace, edit it to set the `URL` property to:

```
https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_month.csv
```

Then click Done to close the dialog.

This URL is a feed of significant earthquakes in the last month from the US Geological Survey web site. The site offers a number of other options that you may want to play around with after completing this tutorial.

1. Copy



Edit http request node

Method: GET

URL: ıquakes/feed/v1.0/summary/significant_month.csv

Payload: Ignore

□ Enable secure (SSL/TLS) connection

□ Use authentication

□ Enable connection keep-alive

□ Use proxy

□ Only send non-2xx responses to Catch node

□ Disable strict HTTP parsing

Return: a UTF-8 string

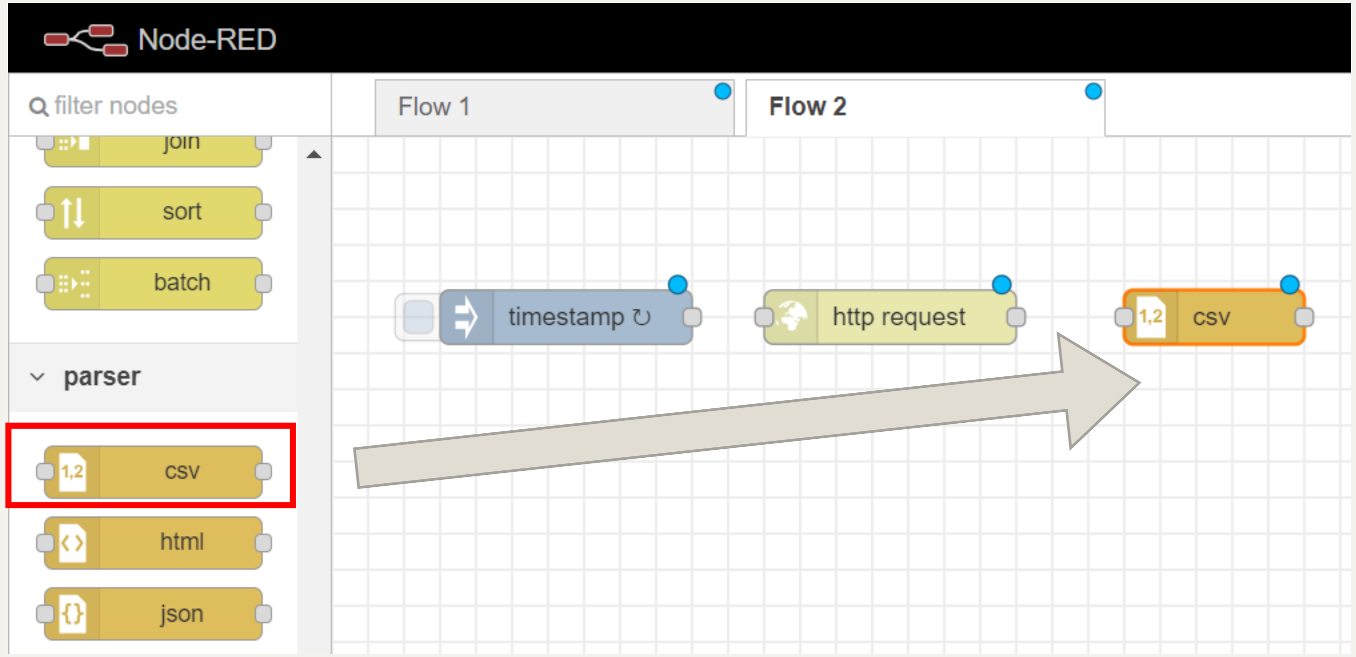
Headers

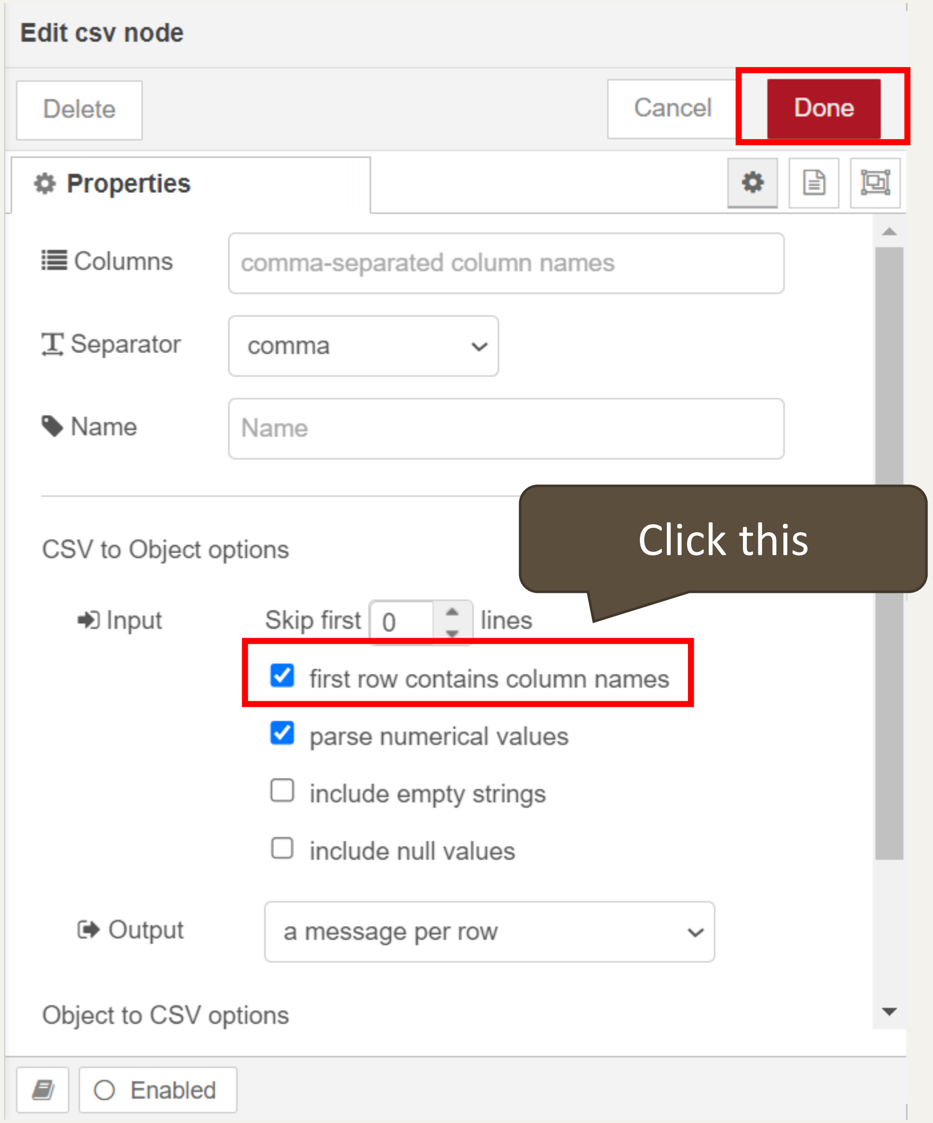
○ Enabled

2. Paste

3. Done

3. Add a CSV node





Edit csv node

Delete | Cancel | Done

Properties

Columns: comma-separated column names

Separator: comma

Name: Name

CSV to Object options

Input: Skip first 0 lines

☑ first row contains column names

☑ parse numerical values

□ include empty strings

□ include null values

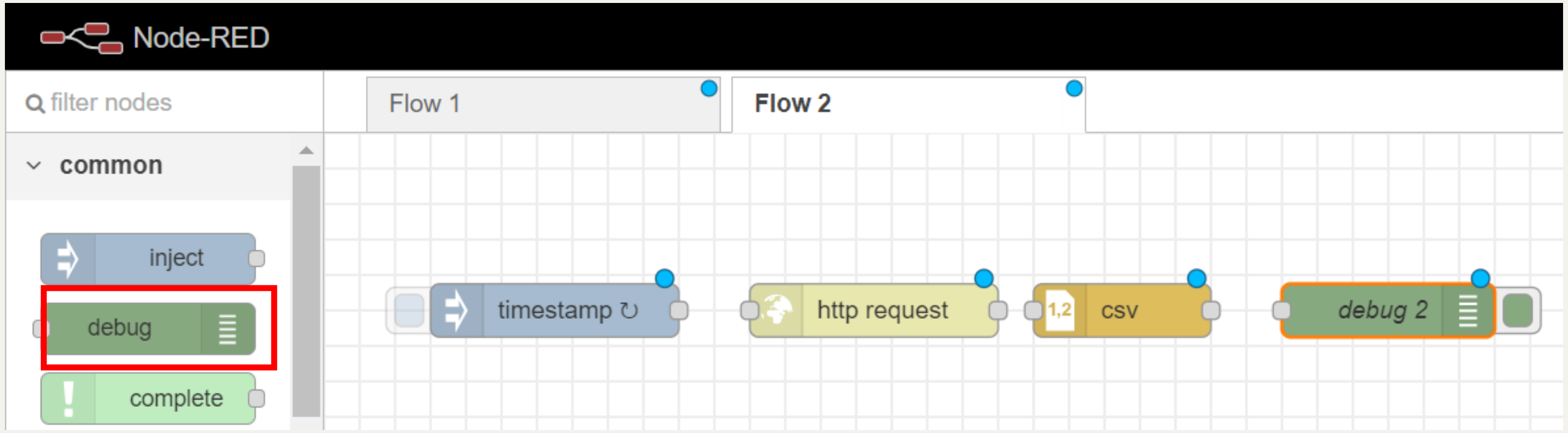
Output: a message per row

Object to CSV options

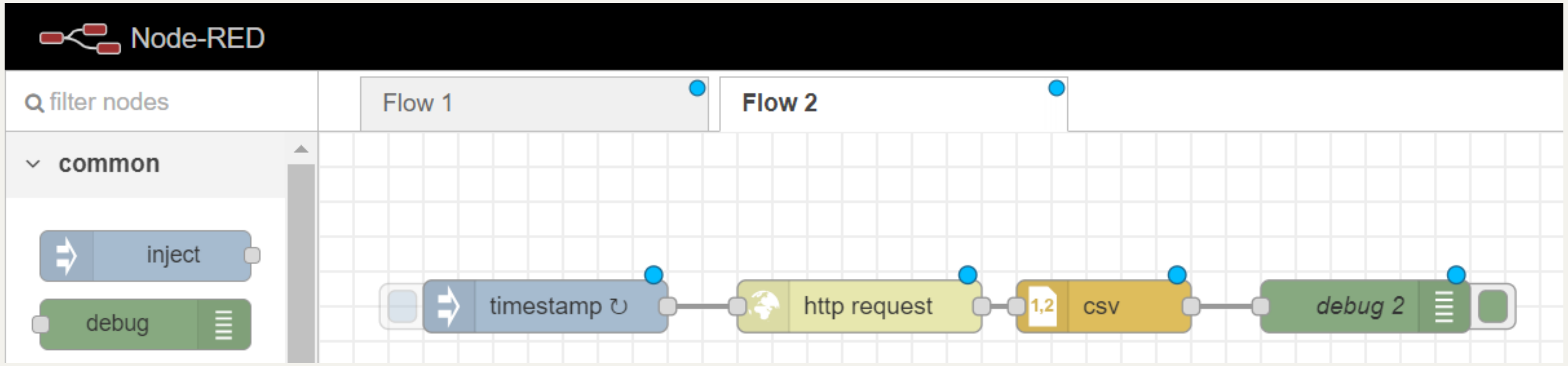
○ Enabled

Click this

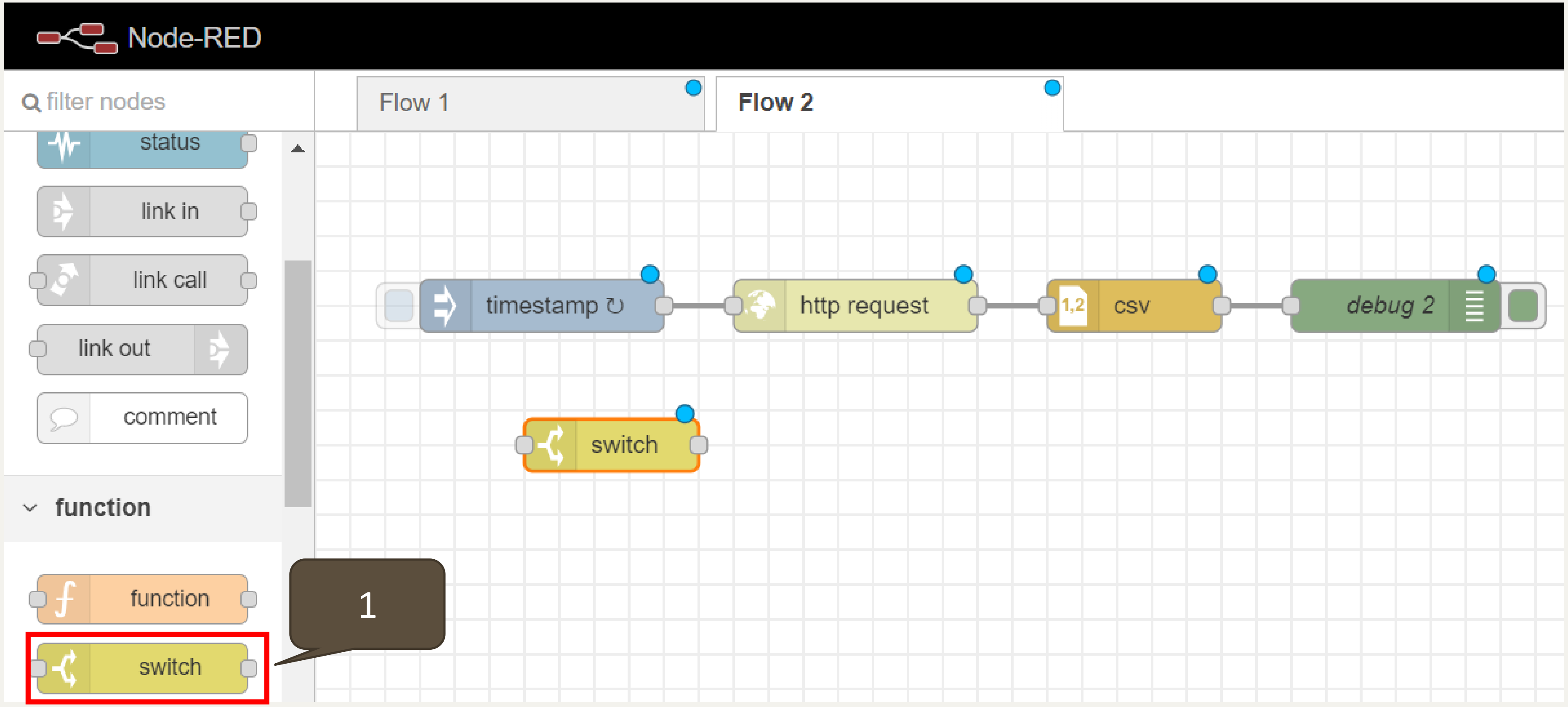
4. Add a Debug node

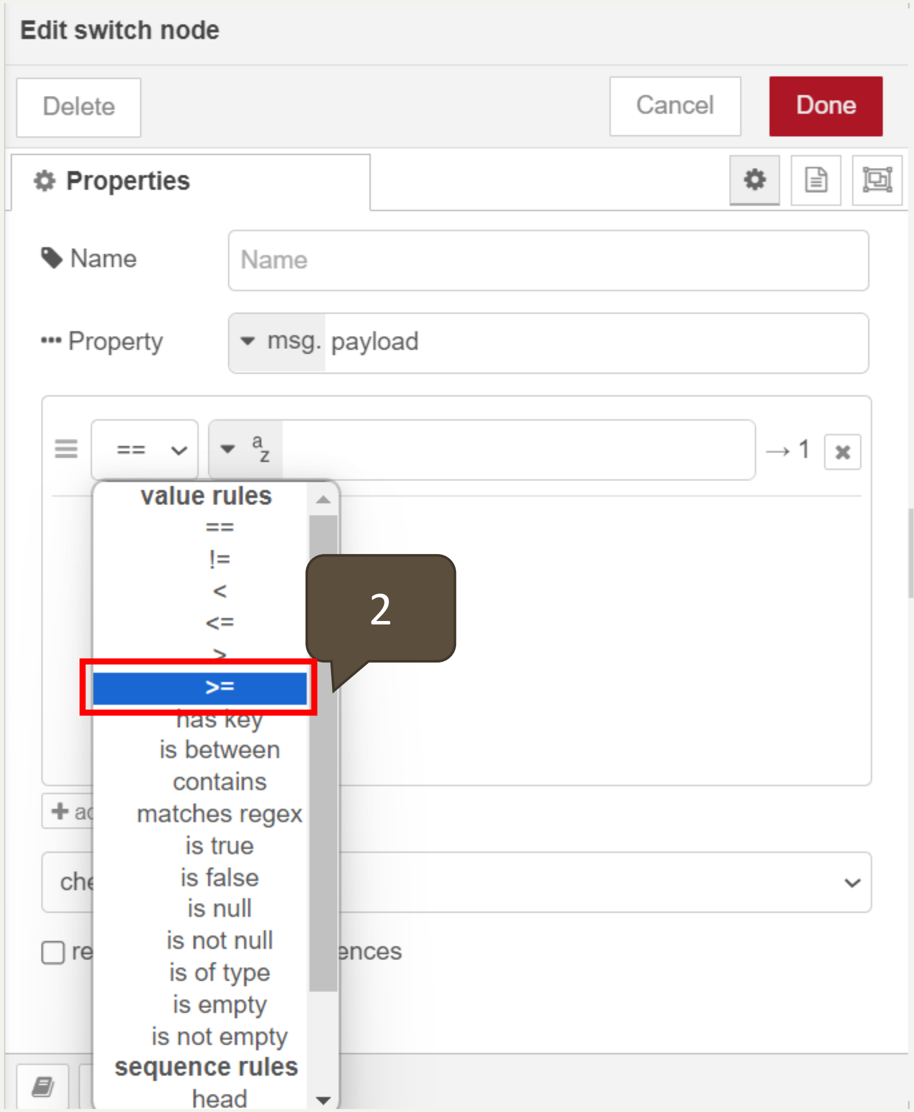


5. Wire them all together



6. Add a Switch node



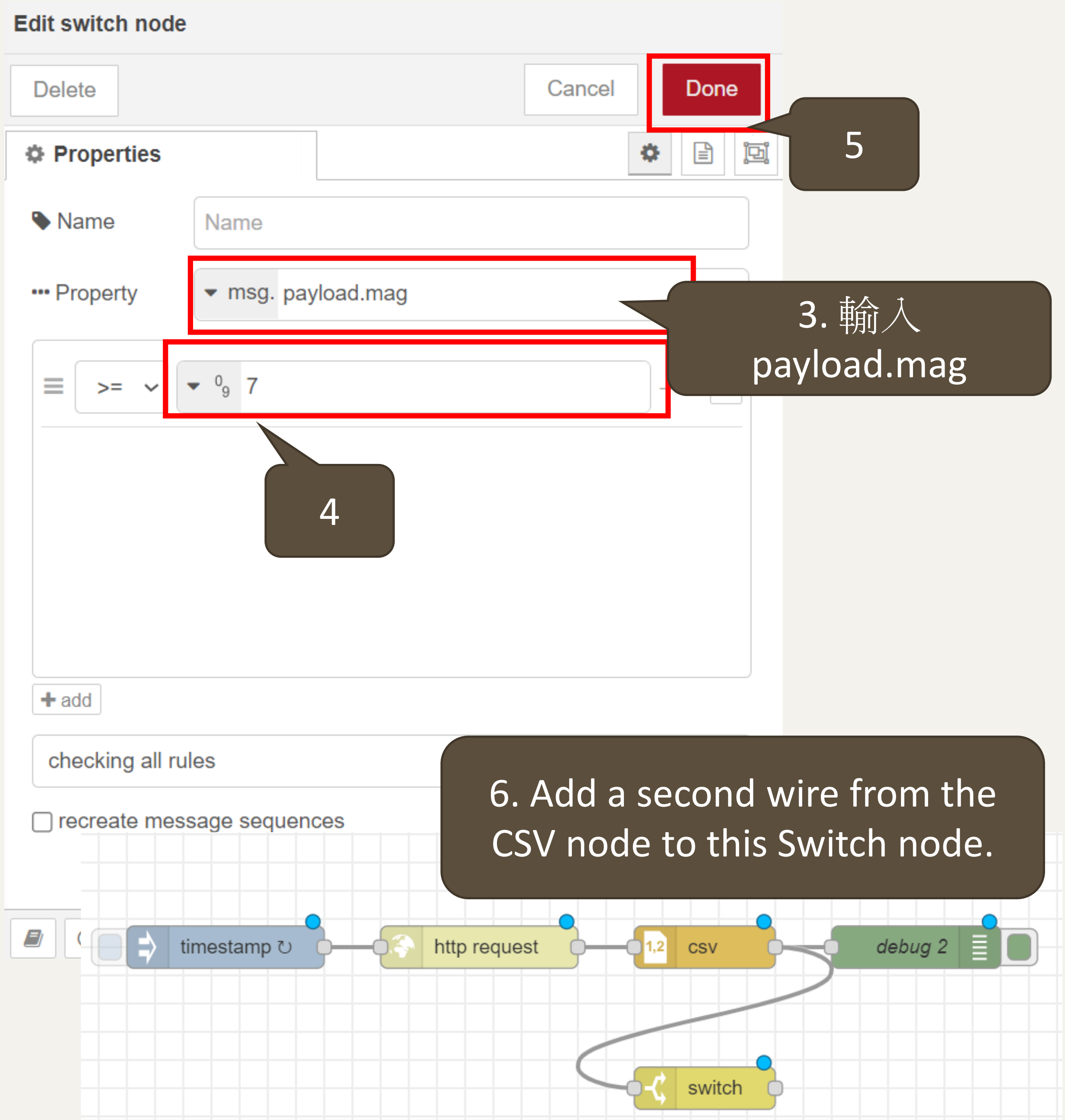


Name: Name

Property: msg. payload

== a-z → 1

2



Name: Name

Property: msg. payload.mag

>= 0-9 7

checking all rules

☐ recreate message sequences

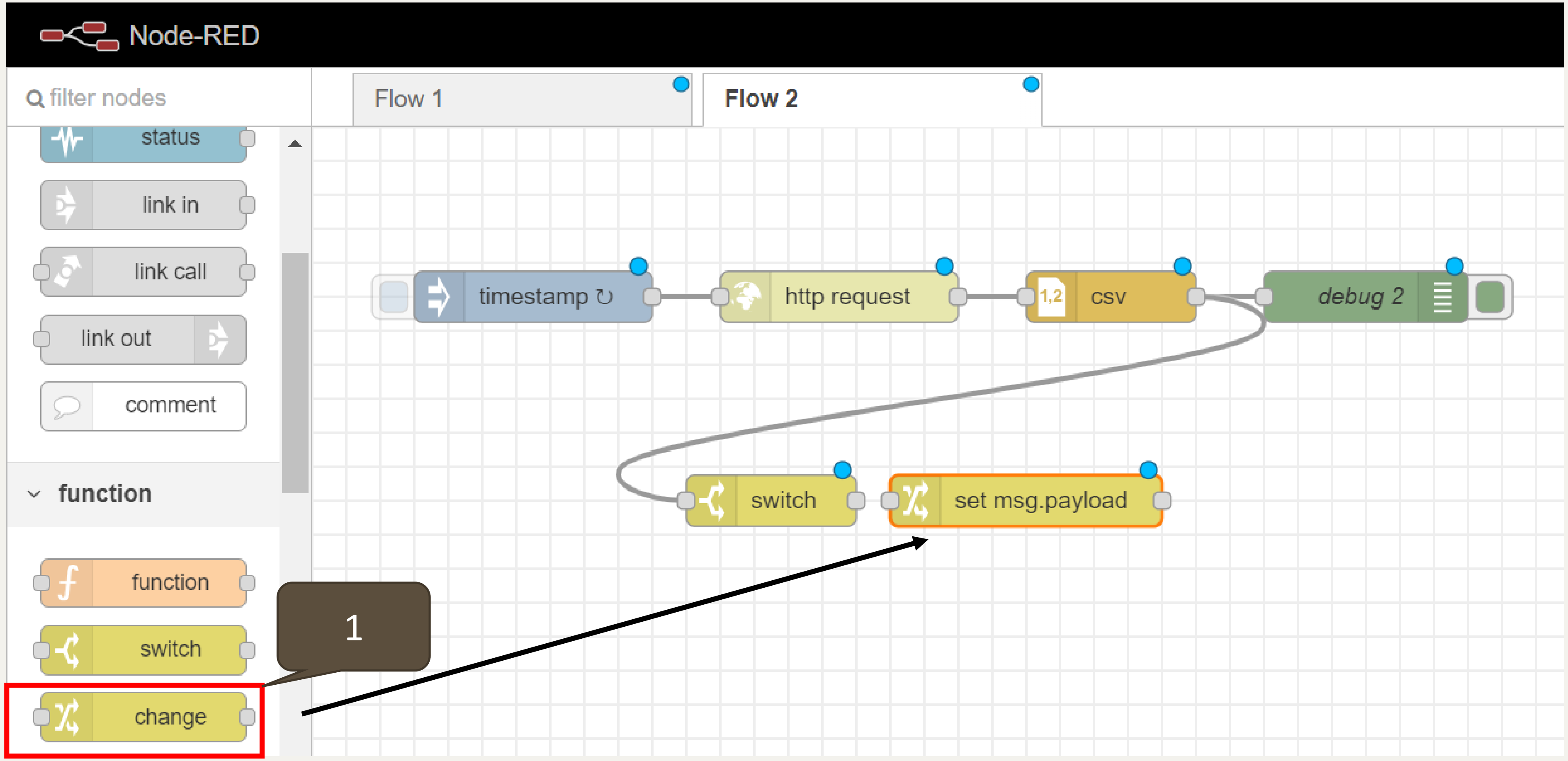
5

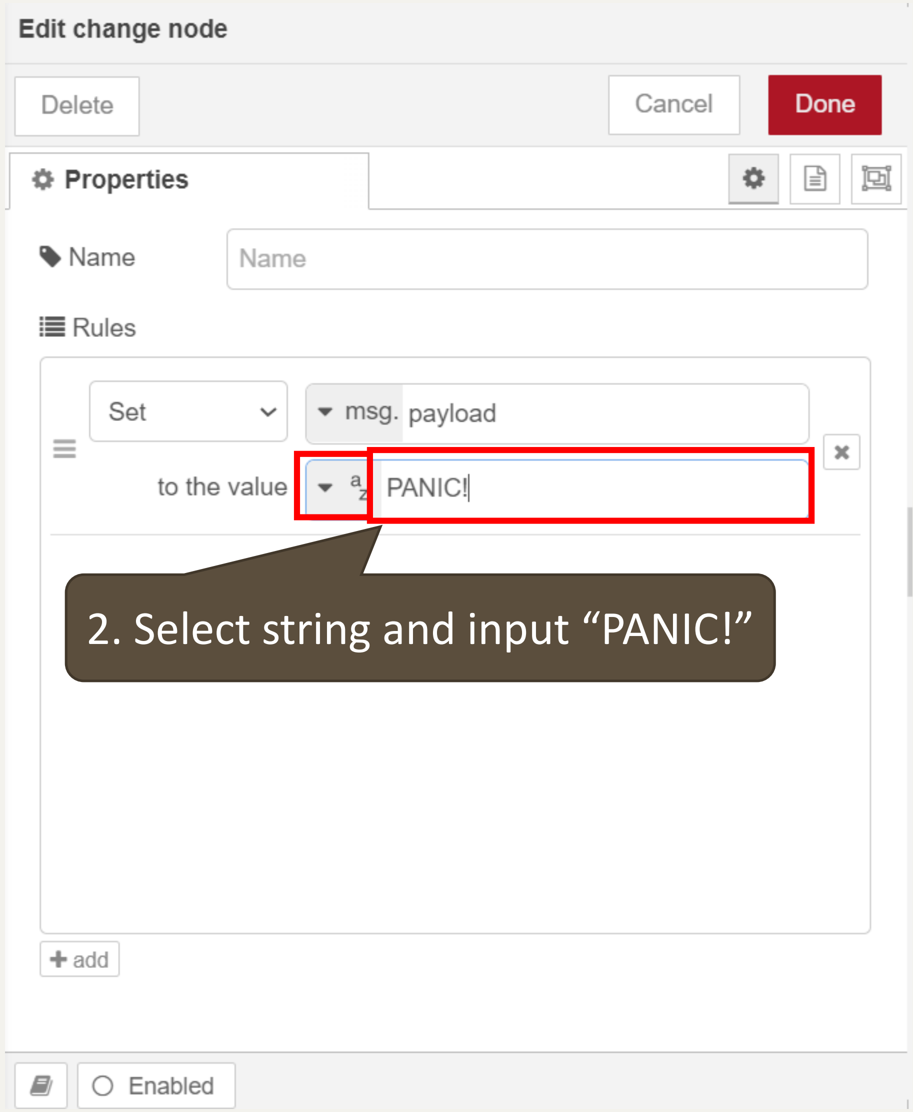
3. 輸入 payload.mag

4

6. Add a second wire from the CSV node to this Switch node.

7. Add a Change node





Edit change node

Delete | Cancel | Done

⚙ Properties

Name: Name

Rules

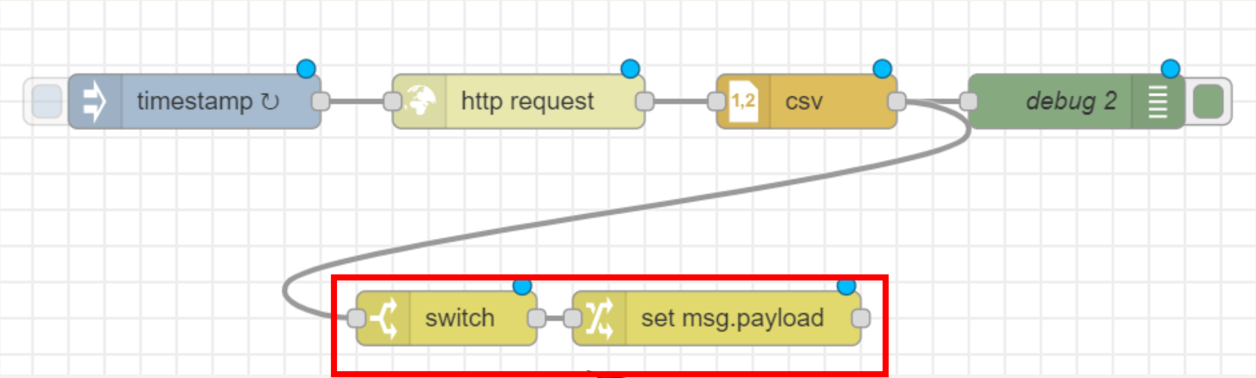
Set msg. payload

to the value PANIC!

+ add

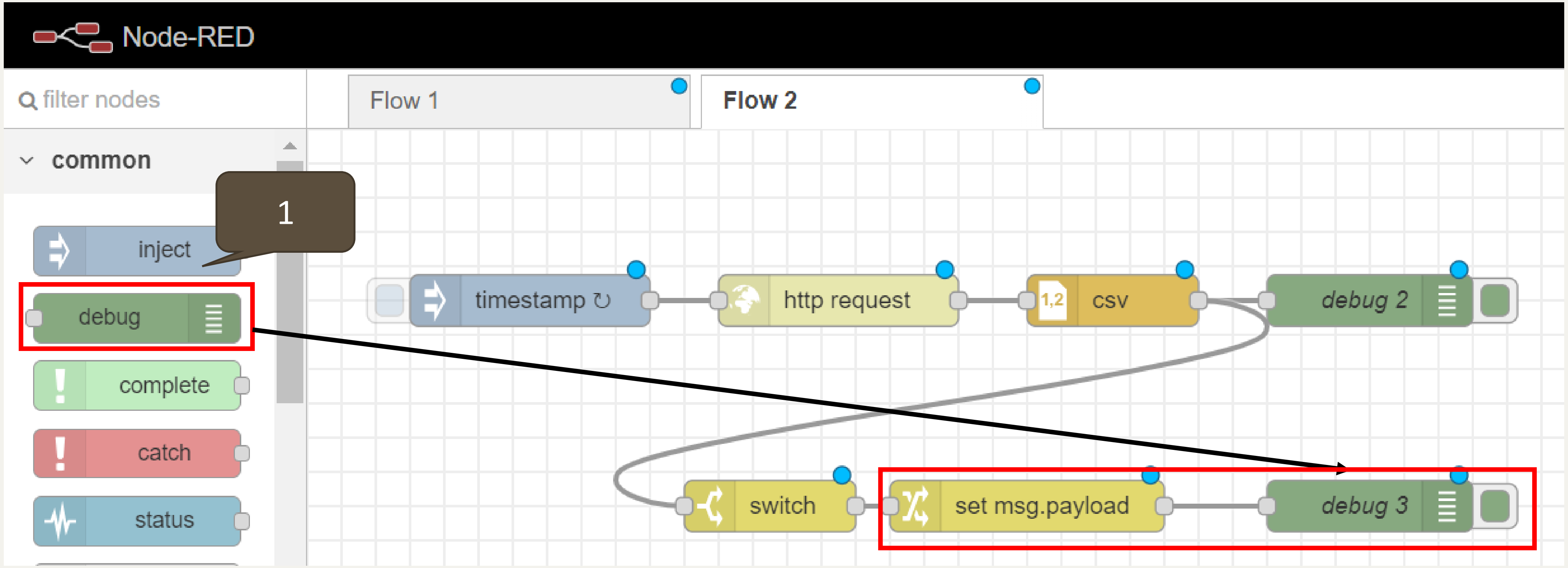
○ Enabled

2. Select string and input “PANIC!”



3. Wire them together

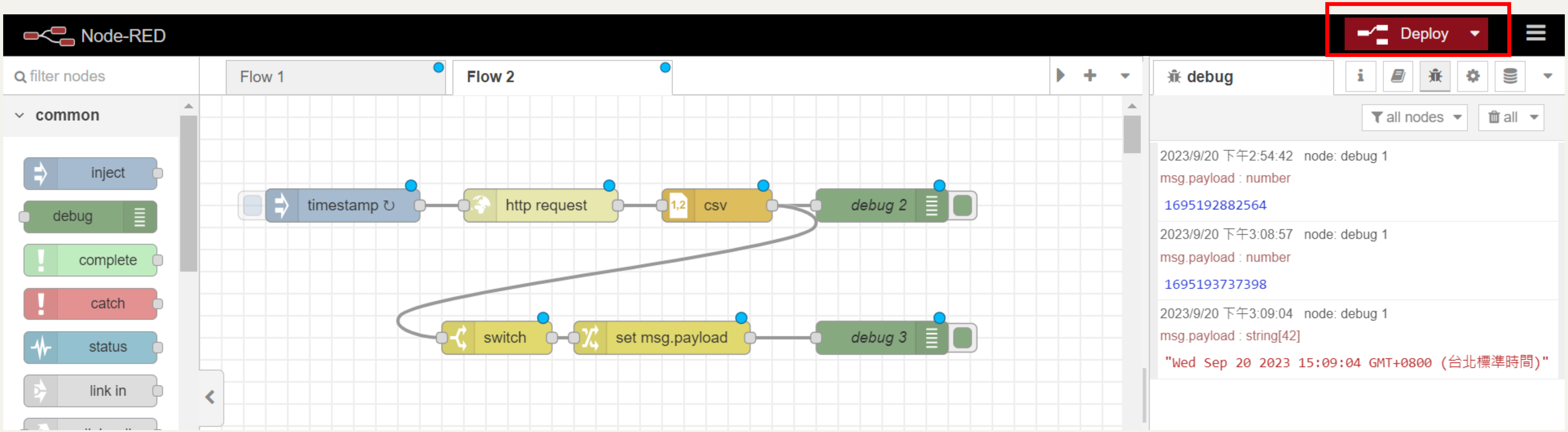
8. Add a Debug node



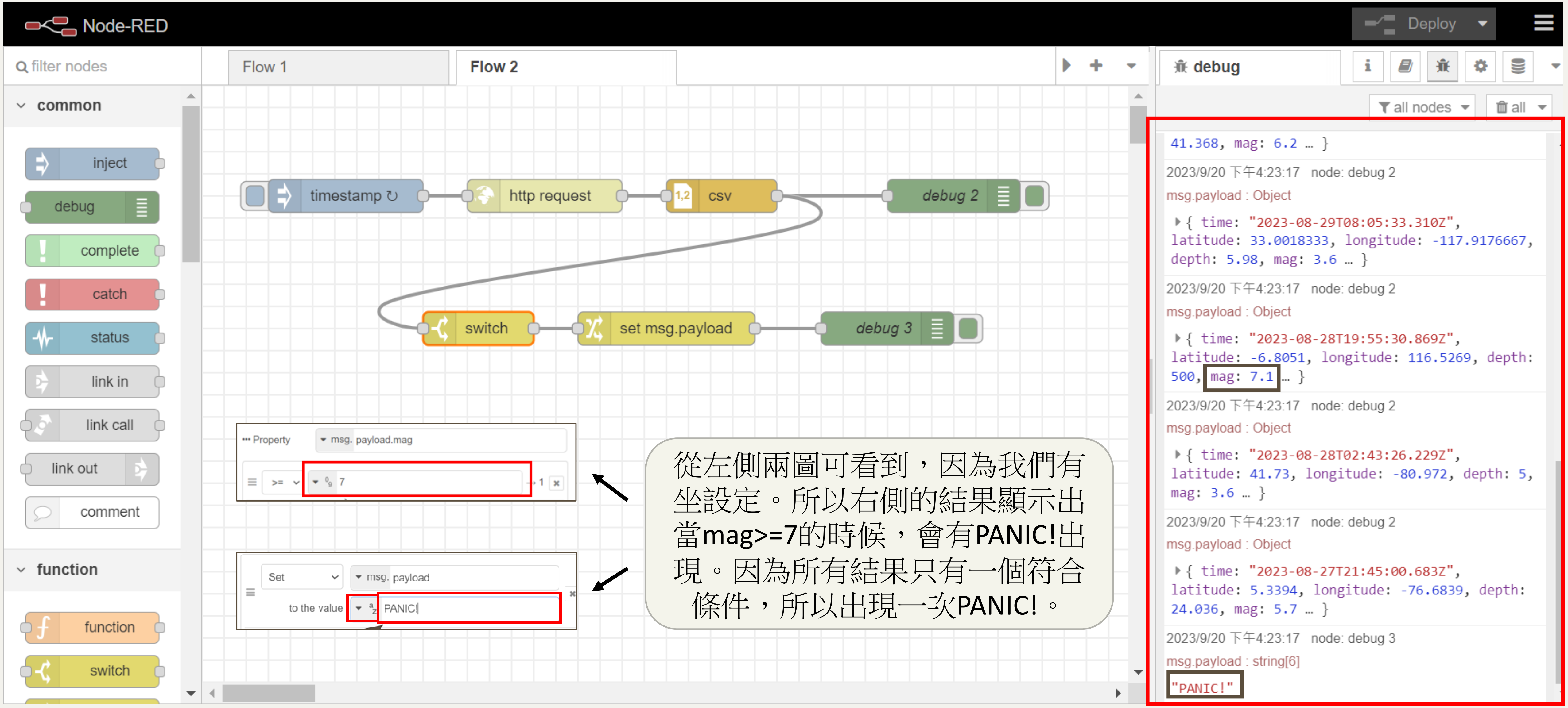
1

2. Wire them together

9. Deploy



```
2023/9/20 下午2:54:42   node: debug 1
msg.payload : number
 1695192882564
2023/9/20 下午3:08:57   node: debug 1
msg.payload : number
 1695193737398
2023/9/20 下午3:09:04   node: debug 1
msg.payload : string[42]
 "Wed Sep 20 2023 15:09:04 GMT+0800 (台北標準時間)"
```



Property msg. payload.mag

>= 7 → 1

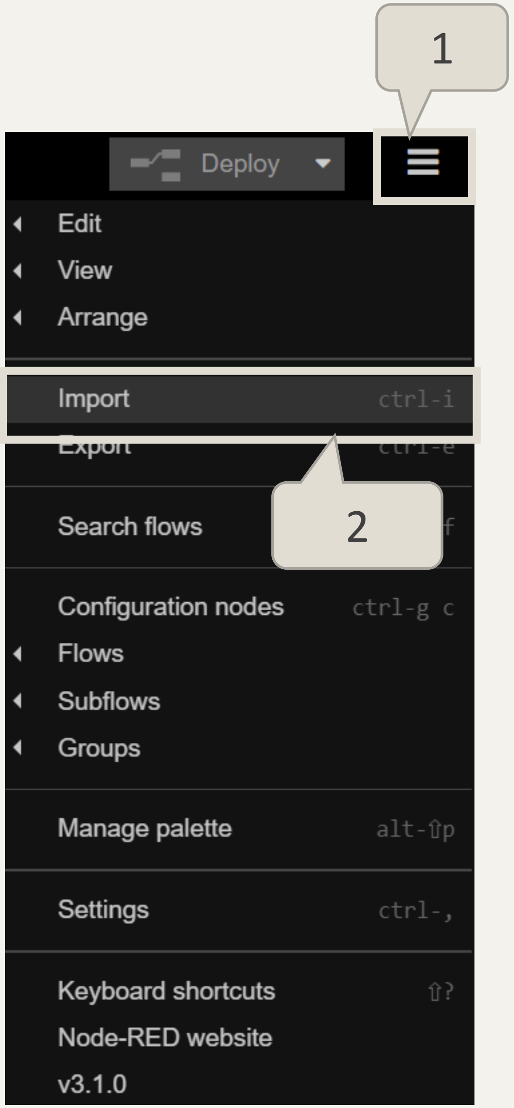
Set msg. payload

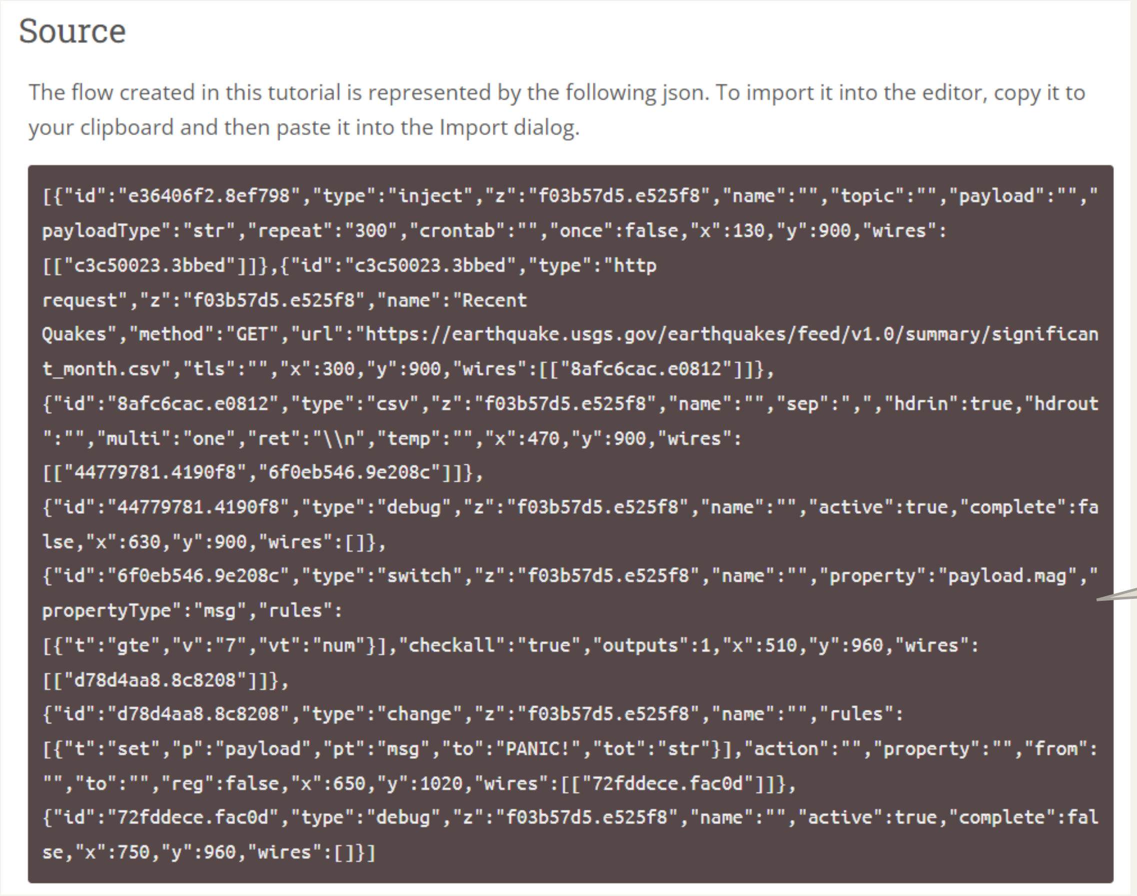
to the value PANIC!

```
41.368, mag: 6.2 … }
2023/9/20 下午4:23:17  node: debug 2
msg.payload : Object
▸{ time: "2023-08-29T08:05:33.310Z",
latitude: 33.0018333, longitude: -117.9176667,
depth: 5.98, mag: 3.6 … }
2023/9/20 下午4:23:17  node: debug 2
msg.payload : Object
▸{ time: "2023-08-28T19:55:30.869Z",
latitude: -6.8051, longitude: 116.5269, depth:
500, mag: 7.1 … }
2023/9/20 下午4:23:17  node: debug 2
msg.payload : Object
▸{ time: "2023-08-28T02:43:26.229Z",
latitude: 41.73, longitude: -80.972, depth: 5,
mag: 3.6 … }
2023/9/20 下午4:23:17  node: debug 2
msg.payload : Object
▸{ time: "2023-08-27T21:45:00.683Z",
latitude: 5.3394, longitude: -76.6839, depth:
24.036, mag: 5.7 … }
2023/9/20 下午4:23:17  node: debug 3
msg.payload : string[6]
"PANIC!"
```

從左側兩圖可看到，因為我們有坐設定。所以右側的結果顯示出當mag>=7的時候，會有PANIC!出現。因為所有結果只有一個符合條件，所以出現一次PANIC!。

透過import 來驗證我們拉的是否正確





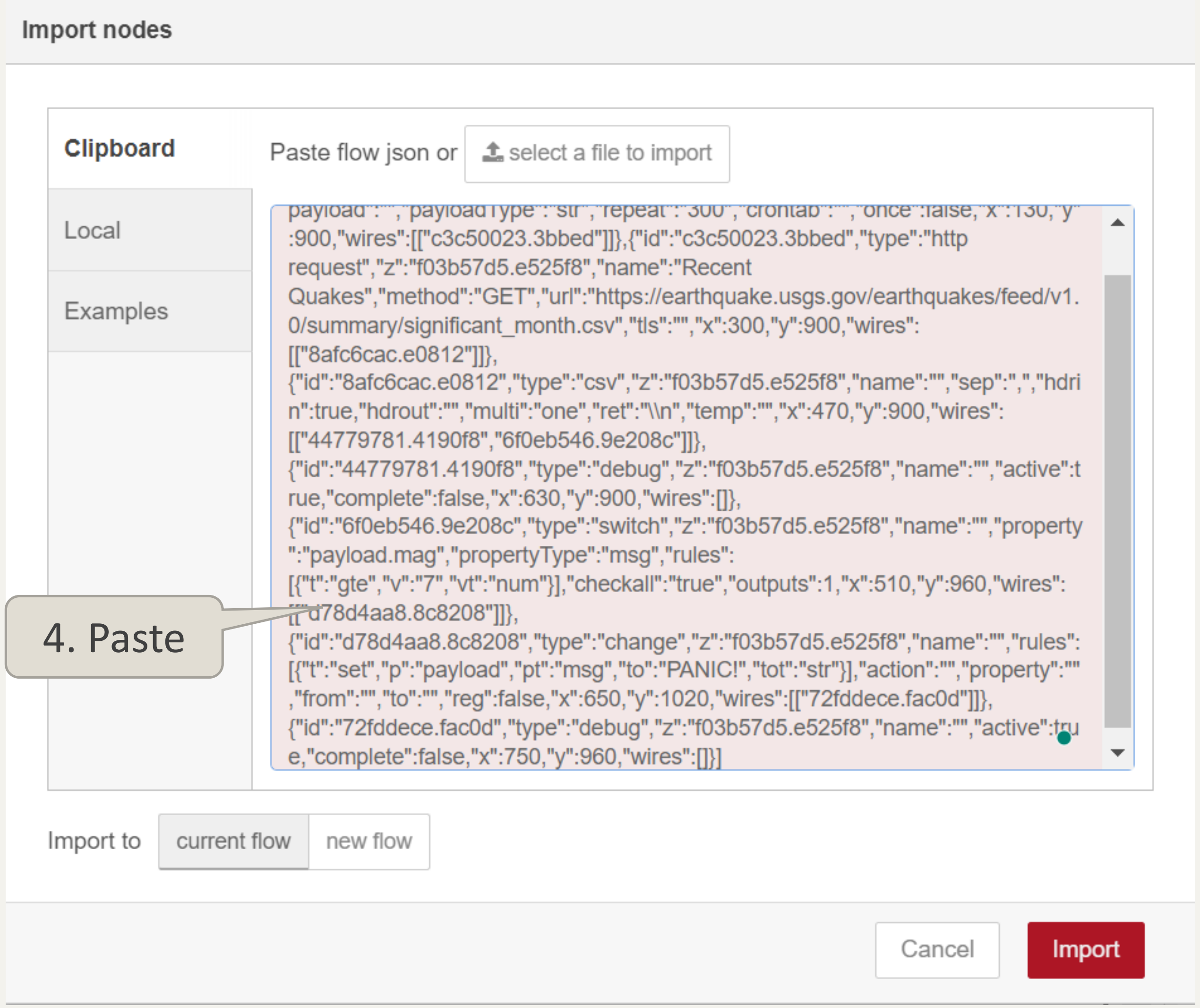
Source

The flow created in this tutorial is represented by the following json. To import it into the editor, copy it to your clipboard and then paste it into the Import dialog.

```
[{"id":"e36406f2.8ef798","type":"inject","z":"f03b57d5.e525f8","name":"","topic":"","payload":"","payloadType":"str","repeat":"300","crontab":"","once":false,"x":130,"y":900,"wires":[["c3c50023.3bbed"]]},{"id":"c3c50023.3bbed","type":"http request","z":"f03b57d5.e525f8","name":"Recent Quakes","method":"GET","url":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_month.csv","tls":"","x":300,"y":900,"wires":[["8afc6cac.e0812"]]},
{"id":"8afc6cac.e0812","type":"csv","z":"f03b57d5.e525f8","name":"","sep":",","hdrin":true,"hdrout":"","multi":"one","ret":"\\n","temp":"","x":470,"y":900,"wires":[["44779781.4190f8","6f0eb546.9e208c"]]},
{"id":"44779781.4190f8","type":"debug","z":"f03b57d5.e525f8","name":"","active":true,"complete":false,"x":630,"y":900,"wires":[]},
{"id":"6f0eb546.9e208c","type":"switch","z":"f03b57d5.e525f8","name":"","property":"payload.mag","propertyType":"msg","rules":[{"t":"gte","v":"7","vt":"num"}],"checkall":"true","outputs":1,"x":510,"y":960,"wires":[["d78d4aa8.8c8208"]]},
{"id":"d78d4aa8.8c8208","type":"change","z":"f03b57d5.e525f8","name":"","rules":[{"t":"set","p":"payload","pt":"msg","to":"PANIC!","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":650,"y":1020,"wires":[["72fddece.fac0d"]]},
{"id":"72fddece.fac0d","type":"debug","z":"f03b57d5.e525f8","name":"","active":true,"complete":false,"x":750,"y":960,"wires":[]}]
```

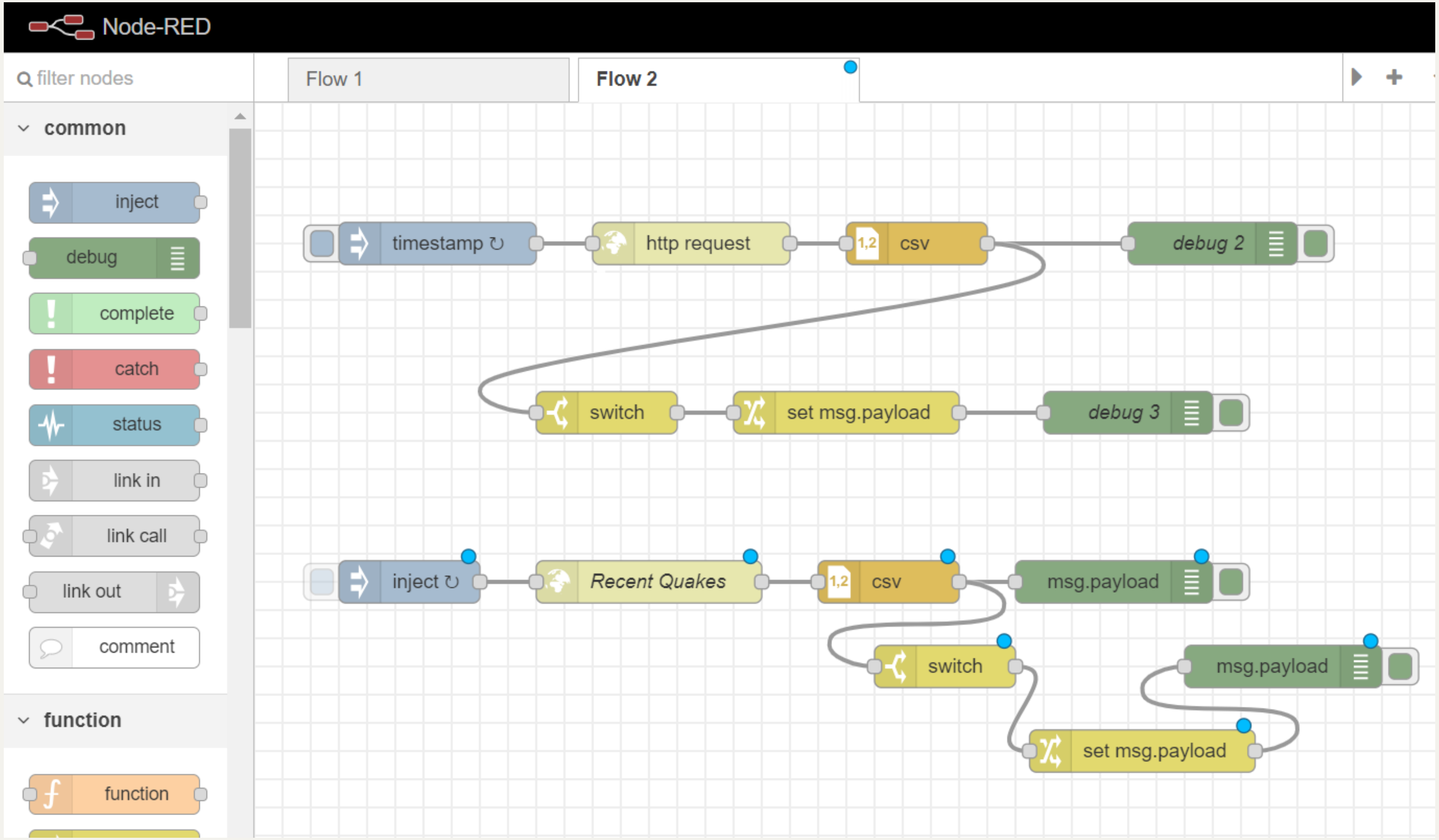
3.copy

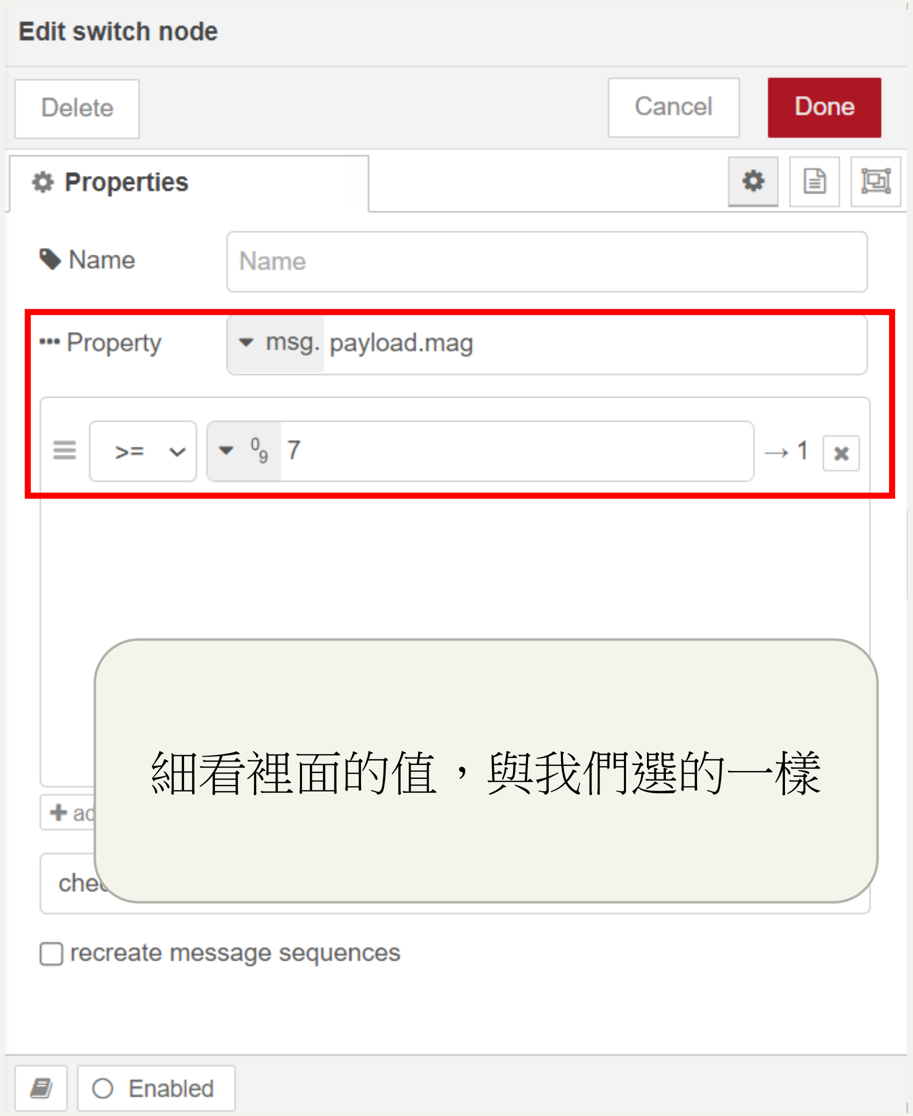
https://nodered.org/docs/tutorials/second-flow



```
payload":"","payloadType":"str","repeat":"300","crontab":"","once":false,"x":130,"y"
:900,"wires":[["c3c50023.3bbed"]]},{"id":"c3c50023.3bbed","type":"http
request","z":"f03b57d5.e525f8","name":"Recent
Quakes","method":"GET","url":"https://earthquake.usgs.gov/earthquakes/feed/v1.
0/summary/significant_month.csv","tls":"","x":300,"y":900,"wires":
[["8afc6cac.e0812"]]},
{"id":"8afc6cac.e0812","type":"csv","z":"f03b57d5.e525f8","name":"","sep":",","hdri
n":true,"hdrout":"","multi":"one","ret":"\\n","temp":"","x":470,"y":900,"wires":
[["44779781.4190f8","6f0eb546.9e208c"]]},
{"id":"44779781.4190f8","type":"debug","z":"f03b57d5.e525f8","name":"","active":t
rue,"complete":false,"x":630,"y":900,"wires":[]},
{"id":"6f0eb546.9e208c","type":"switch","z":"f03b57d5.e525f8","name":"","property
":"payload.mag","propertyType":"msg","rules":
[{"t":"gte","v":"7","vt":"num"}],"checkall":"true","outputs":1,"x":510,"y":960,"wires":
[["d78d4aa8.8c8208"]]},
{"id":"d78d4aa8.8c8208","type":"change","z":"f03b57d5.e525f8","name":"","rules":
[{"t":"set","p":"payload","pt":"msg","to":"PANIC!","tot":"str"}],"action":"","property":""
,"from":"","to":"","reg":false,"x":650,"y":1020,"wires":[["72fddece.fac0d"]]},
{"id":"72fddece.fac0d","type":"debug","z":"f03b57d5.e525f8","name":"","active":tru
e,"complete":false,"x":750,"y":960,"wires":[]}]
```

4. Paste


Node-RED
filter nodes
Flow 1
Flow 2
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
timestamp ↻
http request
csv
debug 2
switch
set msg.payload
debug 3
inject ↻
Recent Quakes
csv
msg.payload
switch
msg.payload
set msg.payload



Edit switch node

Name: Name

Property: msg. payload.mag

>= 0_9 7 → 1

□ recreate message sequences

細看裡面的值，與我們選的一樣