

# Node-RED

```
$ npm install -g node-red
```

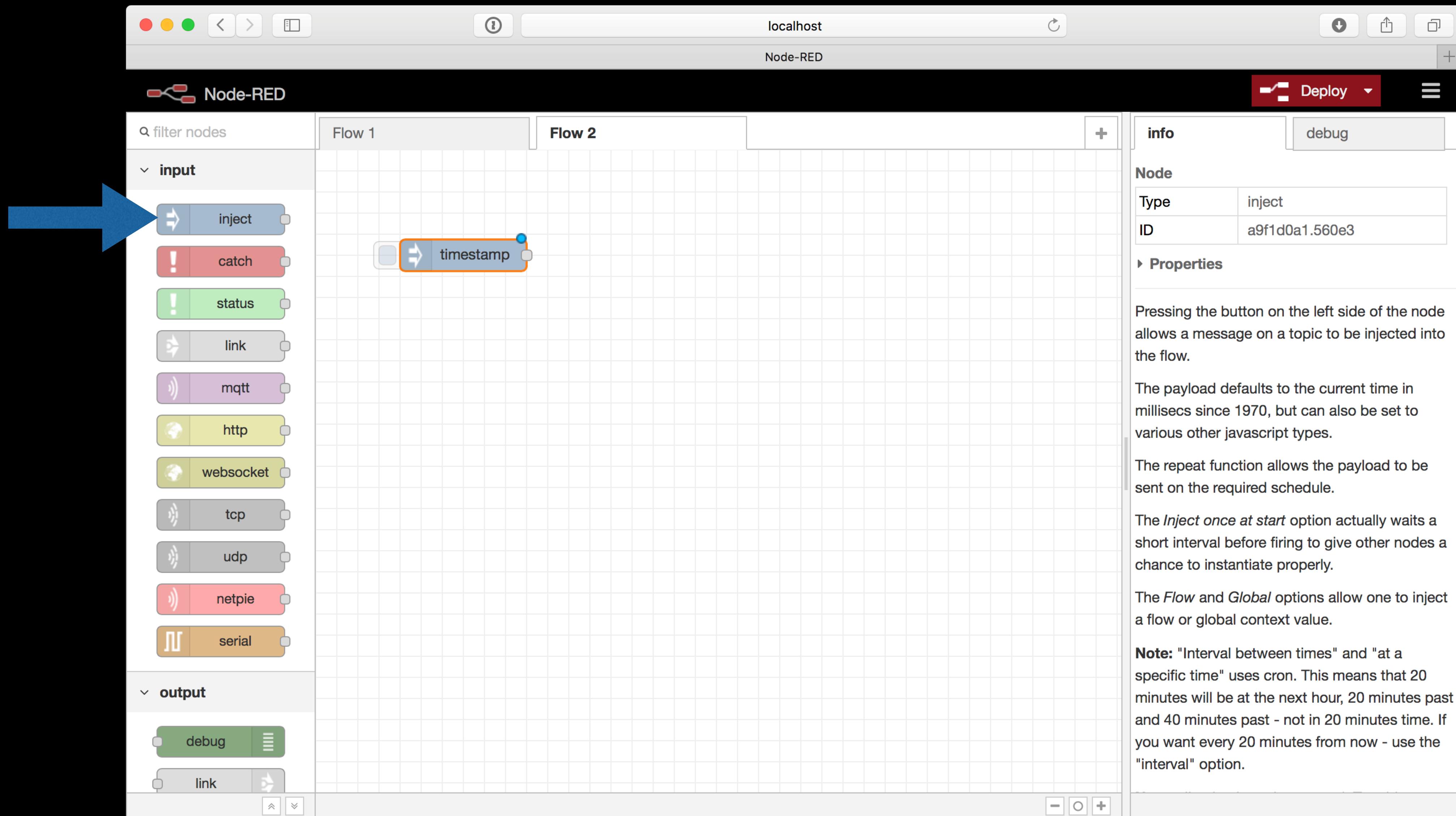
```
$ npm install -g node-red-node-netpie
```

```
$ node-red
```

```
1: v8::Template::Set(v8::Local<v8::Name>, v8::Local<v8::Data>, v8::PropertyAttribute)
2: SerialportPoller::Init(v8::Local<v8::Object>)
3: init
4: node::DLOpen(v8::FunctionCallbackInfo<v8::Value> const&)
5: v8::internal::FunctionCallbackArguments::Call(void (*) (v8::FunctionCallbackInfo<v8::Value> const&))
6: v8::internal::MaybeHandle<v8::internal::Object> v8::internal::(anonymous namespace)::HandleApiCallHelper<false>(v8::internal::Isolate*, v8::internal::(anonymous namespace)::BuiltinArguments<(v8::internal::BuiltinExtraArguments)1>)
7: v8::internal::Builtin_HandleApiCall(int, v8::internal::Object**, v8::internal::Isolate*)
8: 0x252e600961b
9: 0x252e61b4be4
```

```
19 Aug 00:56:43 - [warn] -----
19 Aug 00:56:43 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
19 Aug 00:56:43 - [warn] -----
19 Aug 00:56:43 - [info] Settings file : /Users/Nat/.node-red/settings.js
19 Aug 00:56:43 - [info] User directory : /Users/Nat/.node-red
19 Aug 00:56:43 - [info] Flows file : /Users/Nat/.node-red/flows_Nats-MacBook-Pro.local.json
19 Aug 00:56:43 - [info] Server now running at http://127.0.0.1:1880/
19 Aug 00:56:43 - [info] Starting flows
```

```
netpie out node created
19 Aug 00:56:43 - [info] Started flows
18c54ac5.e73ab5 is connected...
505b4692.afa4b8 is connected...
edd1cbbf.122e38 is connected...
2ee2dad1.d11d26 is connected...
fb5a25ec.04a5d8 is connected...
```



localhost

Node-RED

Flow 1

Flow

Node-RED

filter nodes

udp

netpie

serial

output

debug

link

mqtt

http response

websocket

tcp

udp

netpie

serial

function

ON

OFF

netpie in

Edit inject node

Cancel Done

Payload

Topic

Repeat

Name

a<sub>z</sub>

flow.

global.

a<sub>z</sub> string

0<sub>9</sub> number

boolean

{ } JSON

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

start?

info debug

Node

Type inject

ID a9f1d0a1.560e3

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.

Open # on this page in a new tab

localhost

Node-RED

Deploy

Flow 1

Flow

Node-RED

filter nodes

input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp
- netpie
- serial

output

- debug
- link

Edit inject node

Payload: ON

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.

info

Type: inject

ID: a9f1d0a1.560e3

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.

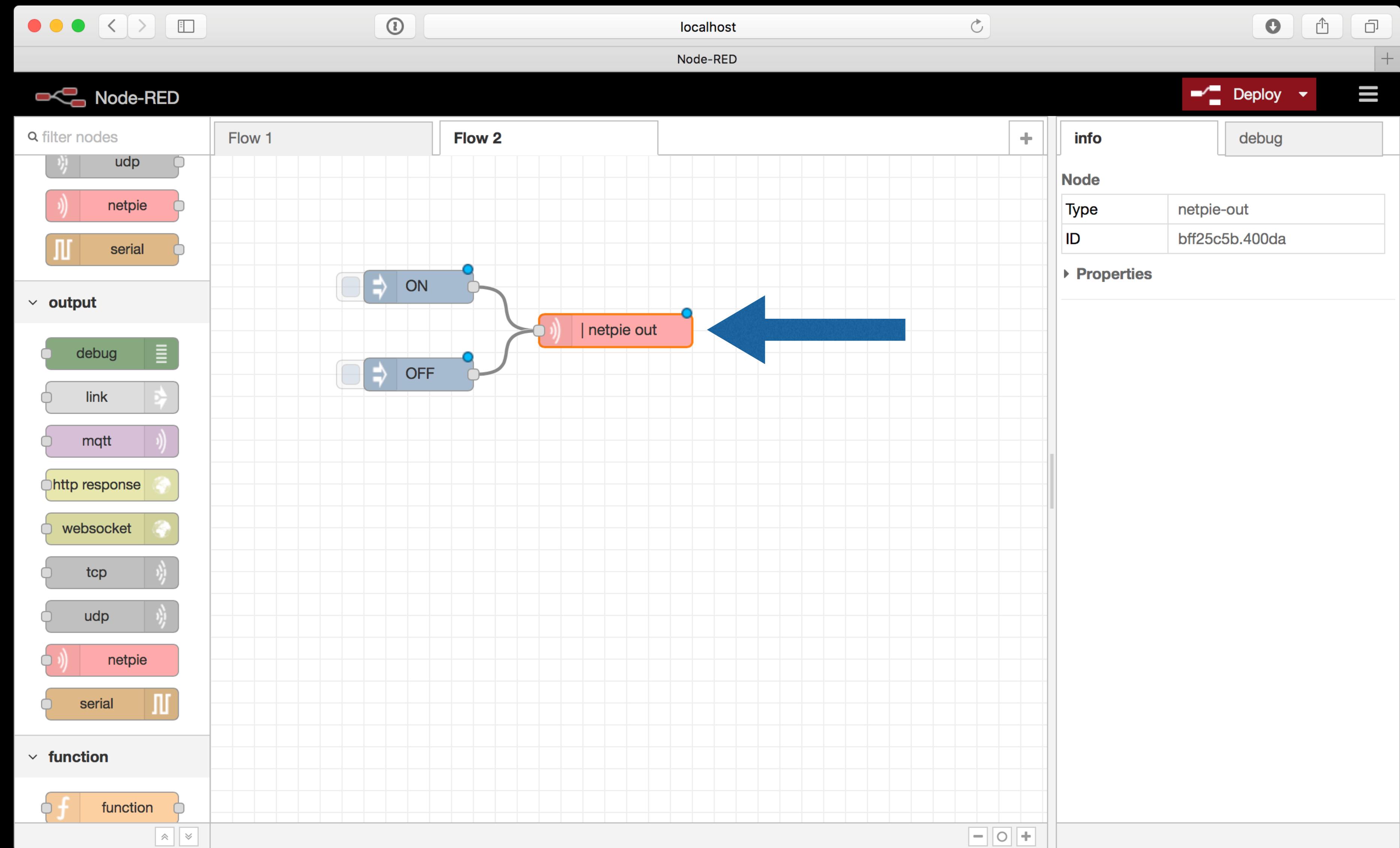
The repeat function allows the payload to be sent on the required schedule.

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.

```
graph LR; inject[inject] --> timestamp[timestamp];
```



localhost

Node-RED

Deploy

Flow 1

ON

OFF

Edit netpie-out node

Cancel Done

AppId: node-red:common.label.appId

AppKey: node-red:common.label.appKey

appSecret: node-red:common.label.appSecret

gearName: node-red:common.label.targetGearName

Tip: Leave target blank if you want to set them via msg properties.

info debug

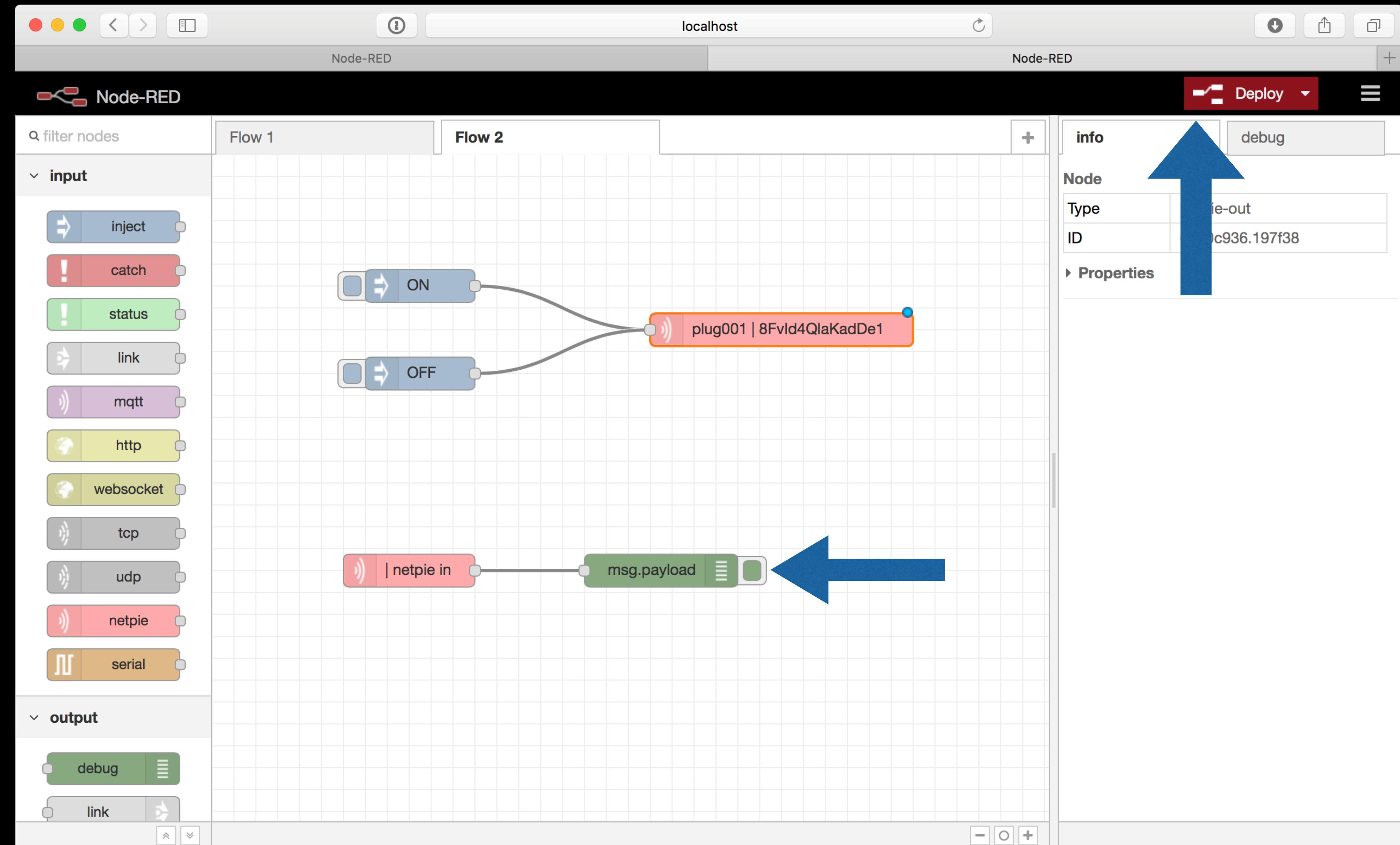
Node

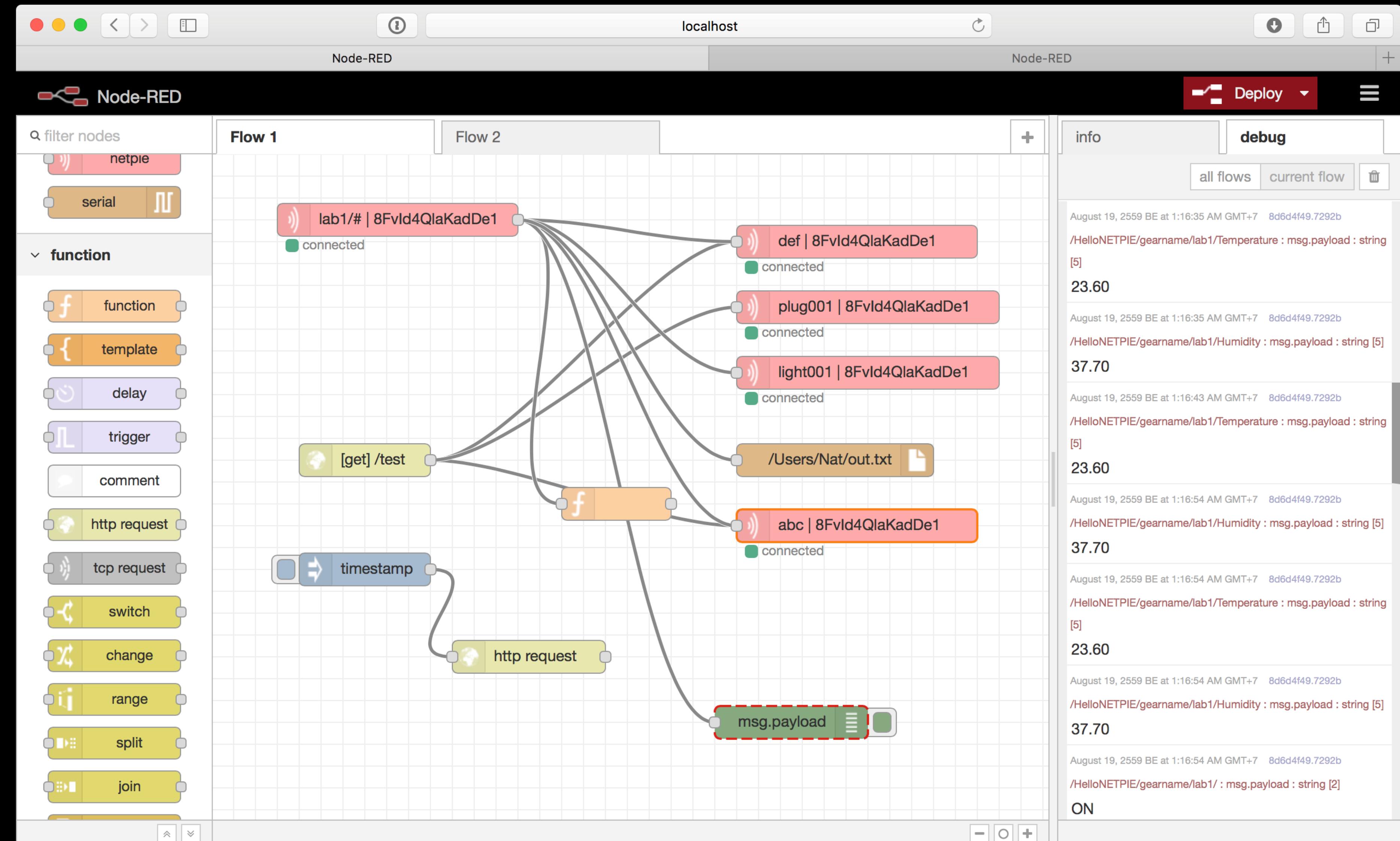
Type: netpie-out

ID: bff25c5b.400da

Properties

The screenshot shows the Node-RED interface running in a web browser. On the left, the palette contains nodes categorized by type: input (e.g., udp, netpie, serial), output (e.g., debug, link, mqtt, http response, websocket, tcp, udp, netpie, serial), and function. A flow titled "Flow 1" is visible on the canvas, starting with an "ON" button node followed by an "OFF" button node. In the center, a modal dialog titled "Edit netpie-out node" is open, showing configuration fields for AppId, AppKey, appSecret, and gearName, each pre-filled with a placeholder value. A tip message at the bottom of the dialog suggests leaving the target field blank for msg properties. To the right of the dialog, a sidebar displays the "info" tab of the node's properties, showing the Type as "netpie-out" and the ID as "bff25c5b.400da". The "Properties" tab is also visible.





<http://flows.nodered.org>

# Node-RED Library

Add a flow

Find new nodes, share your flows and see what other people have done with Node-RED.

mysql

flows     nodes

3 of 887 things

## node-red-node-mysql

A Node-RED node to read and write to a MySQL database



node

## node-red-contrib-odbc

Node for node-red to unixODBC and its supported drivers. (MS SQL, PostgreSQL, ...)

node

## Data in -> MySql and Google Spreadsheet out

by nuriyan

flow

```
$ npm install -g
```

# Node-RED Library

Add a flow

Find new nodes, share your flows and see what other people have done with Node-RED.

Search library

flows     nodes

887 things

**iot-camp-001-full-example**

by NAzT

flow

**node-red-contrib-thingrest**

A Node-RED node to REST write property values to ThingWorx

node

**node-red-node-wemo**

Input and Output nodes for Belkin WeMo devices



node

**node-red-contrib-chatbot**

REDBot a Chat bot for a full featured chat bot for Telegram\*\*, Facebook Messenger and Slack. Almost no coding skills required

node

**node-red-contrib-web-worldmap**

A Node-RED node to provide a web page of a world map for plotting things on.

node

**node-red-contrib-hdl**

HDL for Node-RED, utilizing pure JavaScript HDL Buspro driver

node

**node-red-contrib-fft**

transform an array via fft (fast fourier transformation), for vibration analysis...

node

**norelite**

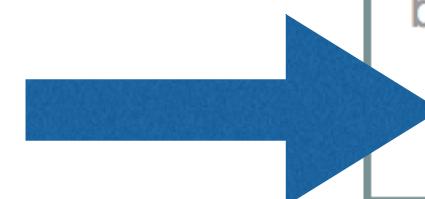
A set of Node-Red nodes to ease the implementation of your home automation requirements

node

**JeeLink to Debug Sample**

by IT-Berater

flow



# iot-camp-001-full-example

## iot-camp-001-full-example

```
[{"id": "fb5a25ec.04a5d8", "type": "netpie-out", "z": "e43f6dbf.1bc09", "appId": "HelloNETPIE", "appKey": "8FvId4QlaKadDe1", "appSecret": "JXcGkxZjcFV3Y1GlGQsKJ2KTs", "name": "", "gearName": "cccc", "targetGearName": "door001", "x": 379, "y": 770, "wires": []}, {"id": "8e3659db.71c9a8", "type": "netpie-out", "z": "e43f6dbf.1bc09", "appId": "HelloNETPIE", "appKey": "8FvId4QlaKadDe1", "appSecret": "JXcGkxZjcFV3Y1GlGQsKJ2KTs", "name": "", "gearName": "cccc", "targetGearName": "def", "x": 590, "y": 80, "wires": []}, {"id": "505b4692.afa4b8", "type": "netpie-out", "z": "e43f6dbf.1bc09", "appId": "HelloNETPIE", "appKey": "8FvId4QlaKadDe1", "appSecret": "JXcGkxZjcFV3Y1GlGQsKJ2KTs", "name": "", "gearName": "cccc", "targetGearName": "abc", "x": 590, "y": 20, "wires": []}, {"id": "8d6d4f49.7292b", "type": "debug", "z": "e43f6dbf.1bc09", "name": "", "active": true, "console": false, "complete": "payload", "x": 530, "y": 520, "wires": []}, {"id": "2391dba4.dc6e24", "type": "http-in", "z": "e43f6dbf.1bc09", "name": "", "url": "/test", "method": "get", "swaggerDoc": "", "x": 140, "y": 280, "wires": [[{"id": "505b4692.afa4b8", "x": 379, "y": 770}, {"id": "8e3659db.71c9a8", "x": 590, "y": 80}, {"id": "18c54ac5.e73ab5", "x": 600, "y": 140}]}], [{"id": "18c54ac5.e73ab5", "type": "netpie-out", "z": "e43f6dbf.1bc09", "appId": "HelloNETPIE", "appKey": "8FvId4QlaKadDe1", "appSecret": "JXcGkxZjcFV3Y1GlGQsKJ2KTs", "name": "", "gearName": "cccc", "targetGearName": "plug001", "x": 600, "y": 140, "wires": []}, {"id": "2ee2dad1.d11d26", "type": "netpie-in", "z": "e43f6dbf.1bc09", "appId": "HelloNETPIE", "appKey": "8FvId4QlaKadDe1", "appSecret": "JXcGkxZjcFV3Y1GlGQsKJ2KTs", "name": "", "gearName": "cccc", "targetGearName": "abc", "x": 590, "y": 20, "wires": []}], [{"id": "2ee2dad1.d11d26", "x": 530, "y": 520}], [{"id": "2ee2dad1.d11d26", "x": 530, "y": 520}]]
```



NAzT

## Flow Info

created 1 hour ago

[view on github](#)

## refresh

**remove**

## Node Types

Core

- debug (x1)
  - file (x1)
  - function (x1)
  - http in (x1)
  - http request (x2)
  - inject (x1)

Other

- netpie-in (x1)
  - netpie-out (x5)

## Tags

[edit](#)

