

Node-Red

Der Fluss der Dinge
oder

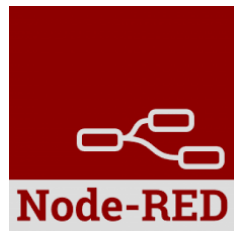
Klick dir dein Internet der Dinge zusammen

Uwe Berger
bergeruw@gmx.net

Uwe Berger



- Beruf: Softwareentwickler
- Freizeit: u.a. mit Hard- und Software rumspielen
- Linux seit ca. 1995
- BraLUG e.V.
- bergeruw@gmx.net

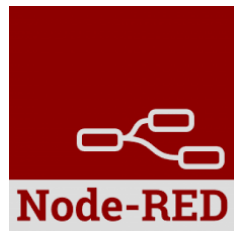


Das Problem...

...Internet of Things (IoT)...:

- Jedes neue „Ding“ bringt eine neue API mit...
- Jeder neue Dienst bringt eine neue API mit...
- Für eine IoT-Lösung müssen diese (vielen) unterschiedlichen Dinge und Dienste verknüpft werden...!

Es kostet Zeit, immer wieder darüber nachzudenken, wie man das Zusammenspiel vieler unterschiedlicher Schnittstellen implementiert!



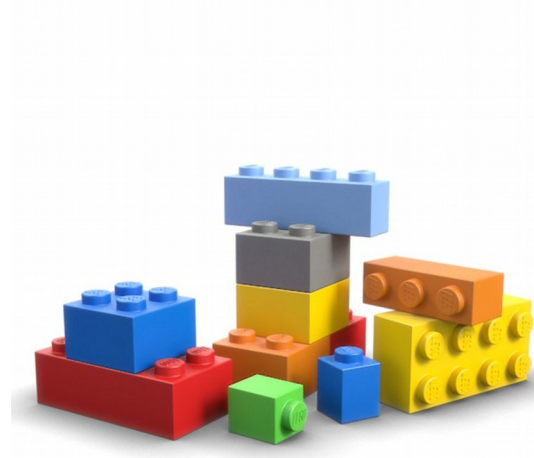
...eine Lösung

Ein Baukasten...:

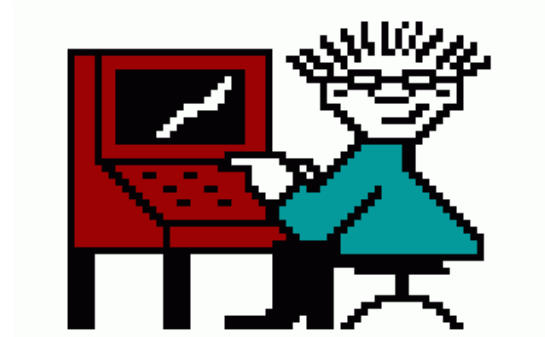
- ...in dem die APIs bereits implementiert und untereinander „normalisiert“ sind...
- ...und in dem man die „Dinge“ einfach nur noch zusammenstecken muss...

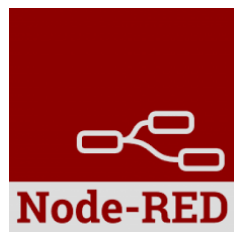
→ Node-Red könnte da eine Lösung sein!

Hääää,... Baukasten...?



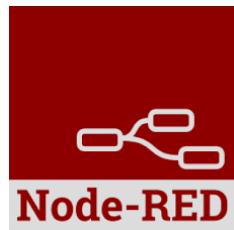
...ich bin doch ein Programmier-
Nerd...?#!.+?!....?





Inhalt

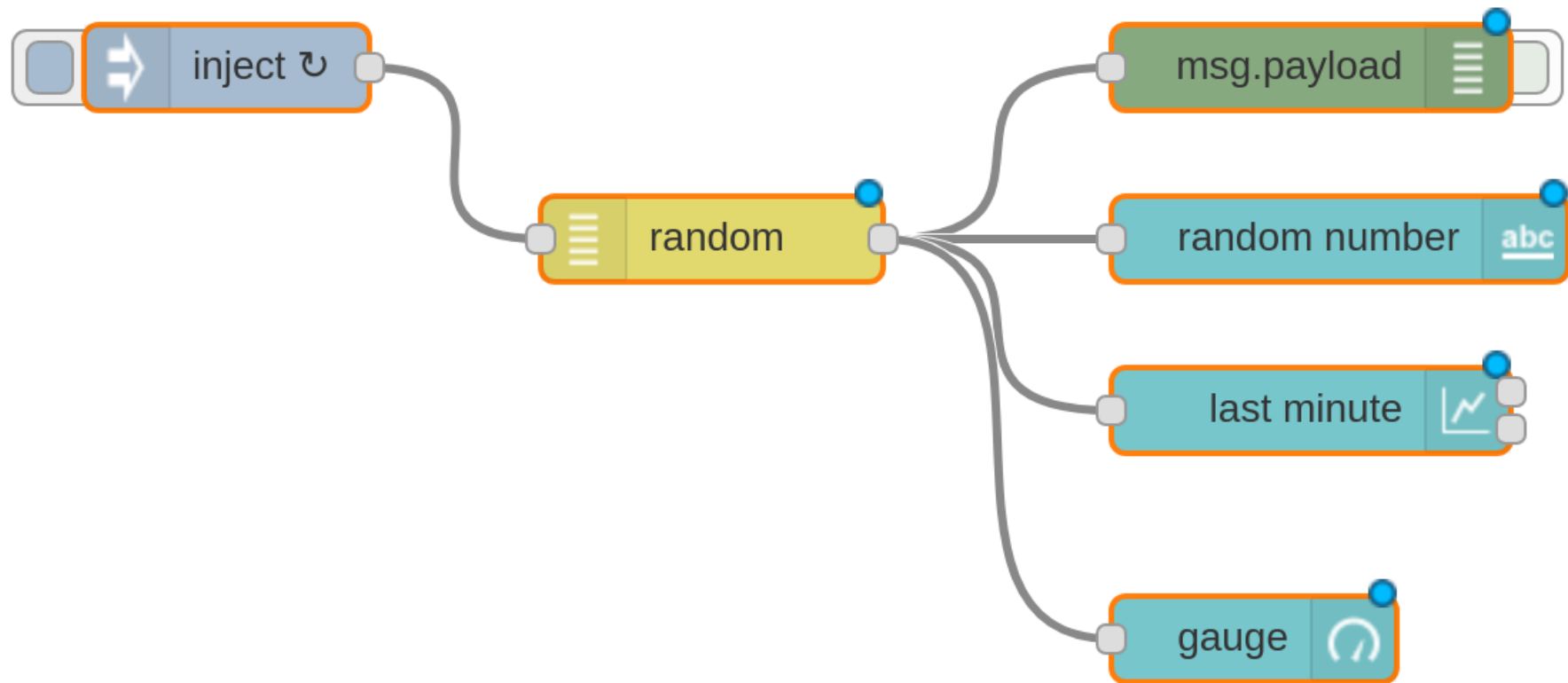
- Was ist Node-Red?
- Installation, Konfiguration von Node-Red
- Arbeiten mit Node-Red...

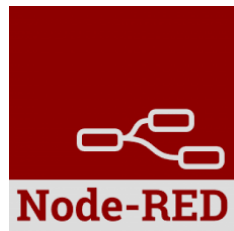


Node-Red („Kennzahlen“)

- Erstes Release: 2013
- IBM; Nick O‘Leary, Dave Conway-Jones
- mittlerweile Open Source (Apache License 2.0)
- Basiert auf Node.js (damit auch plattformübergreifend)
- Browser-basierte Entwicklungsumgebung
- Konzept der Datenflussprogrammierung...

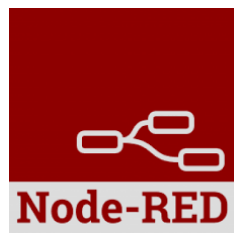
...Datenflussprogrammierung...





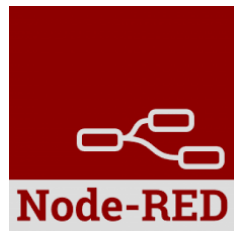
Node-Red ist...

- ein Baukasten zum Zusammensetzen von Anwendungen...
- ...via Drag and Drop
- Ideal für Prototyp-Entwicklung (proof of concept)
- Ideal zur Implementierung von einfachen Abläufen
- einfach zu erweitern
- mehr als ein Baukasten zum Zusammenstellen von Weboberflächen!



Node-Red ist...

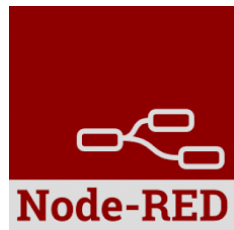
- ...**keine** Laufzeitumgebung für unternehmenskritische Anwendungen



Node-Red installieren...

- Node.js und npm (node package manager) via apt installieren
- Node-Red via npm installieren

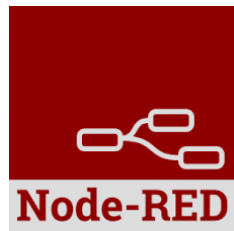
```
$ sudo apt-get install nodejs npm
...
$ sudo npm install -g --unsafe-perm node-red
...
```



Node-Red konfigurieren...

- Datei **settings.js** (Reihenfolge der Suche)
 - über Kommandozeilenparameter
 - in **/home/<user>/.node-red/**
 - im Installationsverzeichnis von Node-Red

- Was kann konfiguriert werden?
 - Ports, Verzeichnisse
 - Sicherheitseinstellungen(!)
 - ...etc. (originale settings.js ist gut kommentiert und/oder RTFM)



Node-Red starten...

- Kommandozeile:

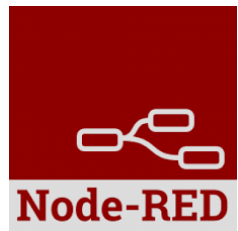
```
$ whereis node-red
node-red: /usr/local/bin/node-red

$ node-red
...
```

- ...oder als Dienst:

- RTFM...
- ...mit systemd z.B.:

→ <https://diyprojects.io/node-red-installation-configuration-ubuntu-16-04-lts/>



http://localhost:1880/

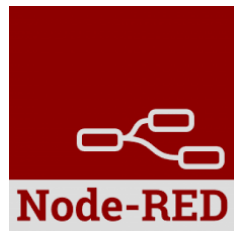
The screenshot shows the Node-RED web interface. The main workspace contains a flow named "forecast" with the following nodes and connections:

- Input:** "trigger http-request" (blue) connects to "http request" (yellow).
- Input:** "request yahoo..." (light blue) connects to "http request" (yellow).
- Input:** "forecast_json" (blue) connects to "json" (yellow).
- Switch:** "switch_first" (orange) receives input from "http request" and "json". It has five outputs connected to "css (text)" nodes.
- Output:** "css (text)" nodes connect to "location", "sun", "last update", "out conditions", and "wind" (teal).
- Output:** "format table datavar" (orange) receives input from "switch_first" and "msg.payload" (green).
- Output:** "format table datavar" connects to "css (table)" (orange), which connects to "generate table forecast" (teal).

The right sidebar shows the "info" panel with the following information:

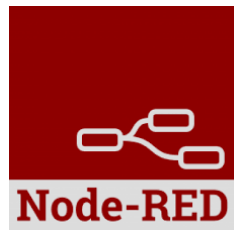
Information	
Flow	"e347139a_b567c"
Name	forecast
Status	Enabled

Below the information panel is the "Flow Description" section, which is currently empty (None).

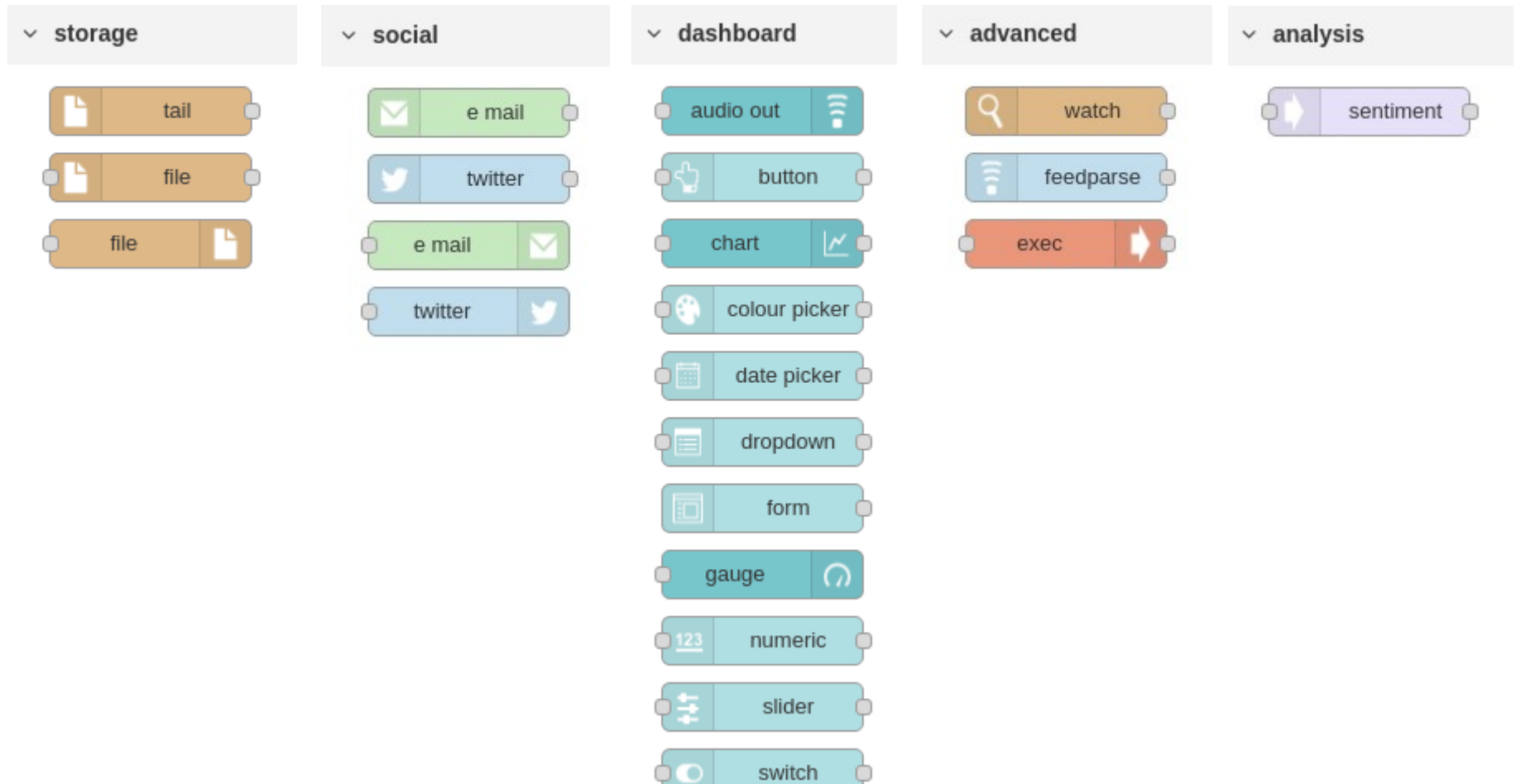


Nodes (Eingabe/Verarbeitung/Ausgabe)

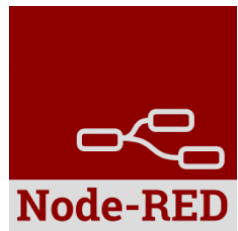
A screenshot of the Node-RED interface showing three categories of nodes: input, function, and output. Each category is represented by a grey header with a dropdown arrow and a list of nodes below it. The 'input' category includes inject, catch, status, link, mqtt, http, websocket, tcp, and udp. The 'function' category includes function, template, delay, trigger, comment, http request, tcp request, switch, change, range, split, and join. The 'output' category includes debug, link, mqtt, http response, websocket, tcp, and udp. Each node is represented by a colored rounded rectangle with an icon and a label.



...viele weitere Nodes



...für Raspberry Pi gibt es da auch was...!

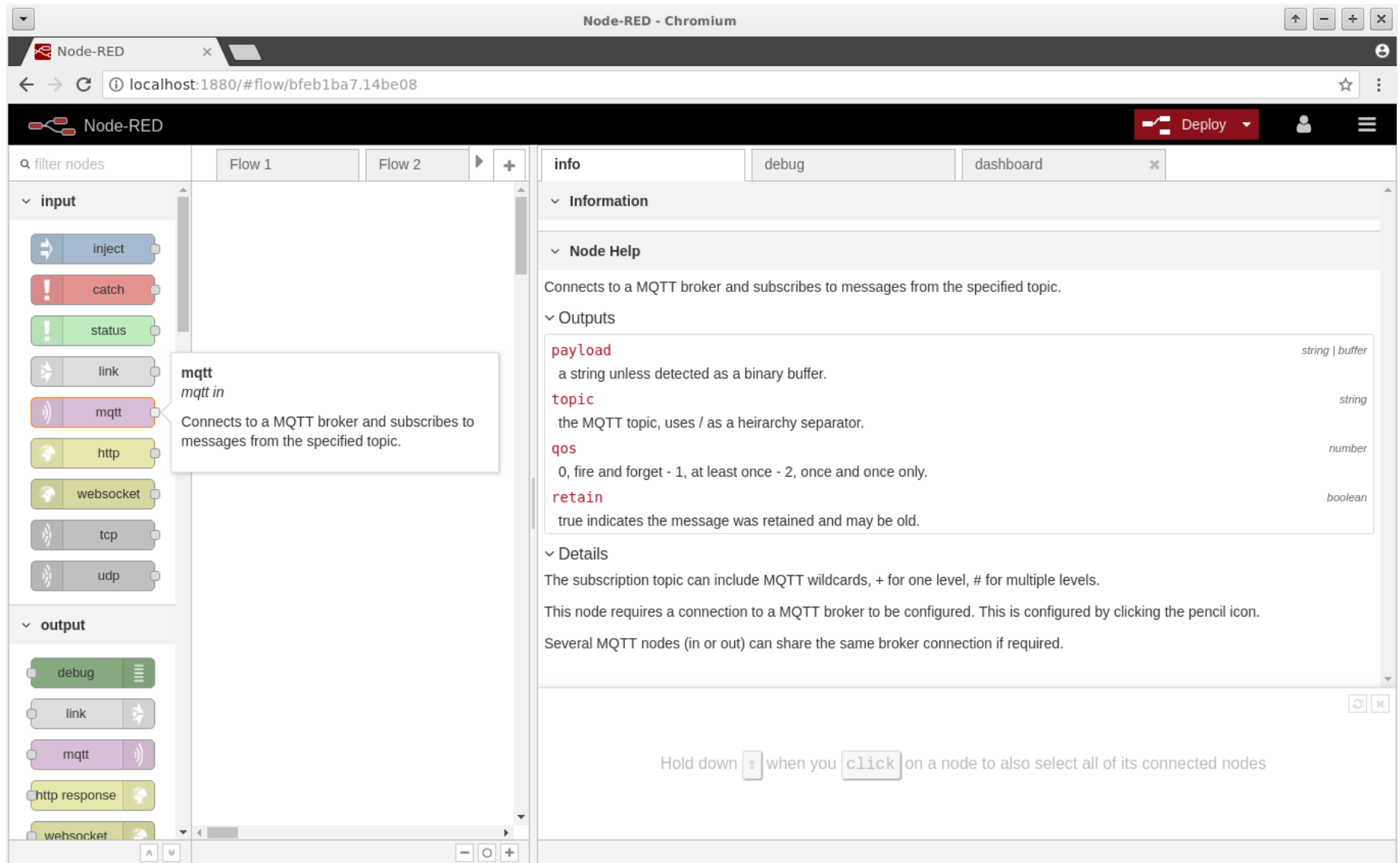


Nodes nachinstallieren...

The screenshot shows the Node-RED interface with the 'User Settings' dialog open. The 'Nodes' tab is selected, and the search bar contains 'Raspberry Pi'. The search results list several nodes with their descriptions, versions, and dates. The 'install' button is highlighted for the first node, 'node-red-contrib-gpio'. Red arrows point from the 'Install' button in the top right of the dialog to the search bar and the 'install' button in the results list.

Node Name	Description	Version	Time Ago	Action
node-red-contrib-gpio	A set of node-red nodes for using johnny-five and IO plugins	0.14.0	4 months ago	install
node-red-contrib-mcp23017	MCP23017 controller	0.0.1	2 years, 8 months ago	install
node-red-contrib-nfc	SL030 RFID/NFC reader node for Node RED on a Raspberry Pi	1.0.0	2 months ago	install
node-red-contrib-opi-gpio	Orange Pi GPIO. Digital input/output for most Pi computers	0.0.5	8 months ago	install
node-red-contrib-pitft-touch	This reads /dev/input/event0 (or other input device) on the Raspberry Pi, which is the Adafruit PITFT touch screen if you used their tutorial to install the shield. *May be* compatible with other touch screens and other Linux-es.	0.0.2	3 years, 11 months ago	install
node-red-contrib-smartmesh	Node-RED nodes for Analog Devices' SmartMesh IP™ Motes and Manager	1.0.1	1 month ago	install

Hilfe zu Nodes



The screenshot shows the Node-RED web interface in a Chromium browser window. The address bar displays `localhost:1880/#flow/bfeb1ba7.14be08`. The interface includes a top navigation bar with a "Deploy" button and a hamburger menu. Below this is a search bar for nodes and tabs for "Flow 1" and "Flow 2".

The left sidebar shows a list of nodes under "input" and "output" categories. The "mqtt" node is selected, and a tooltip is visible over it with the text: `mqtt`, `mqtt in`, and "Connects to a MQTT broker and subscribes to messages from the specified topic."

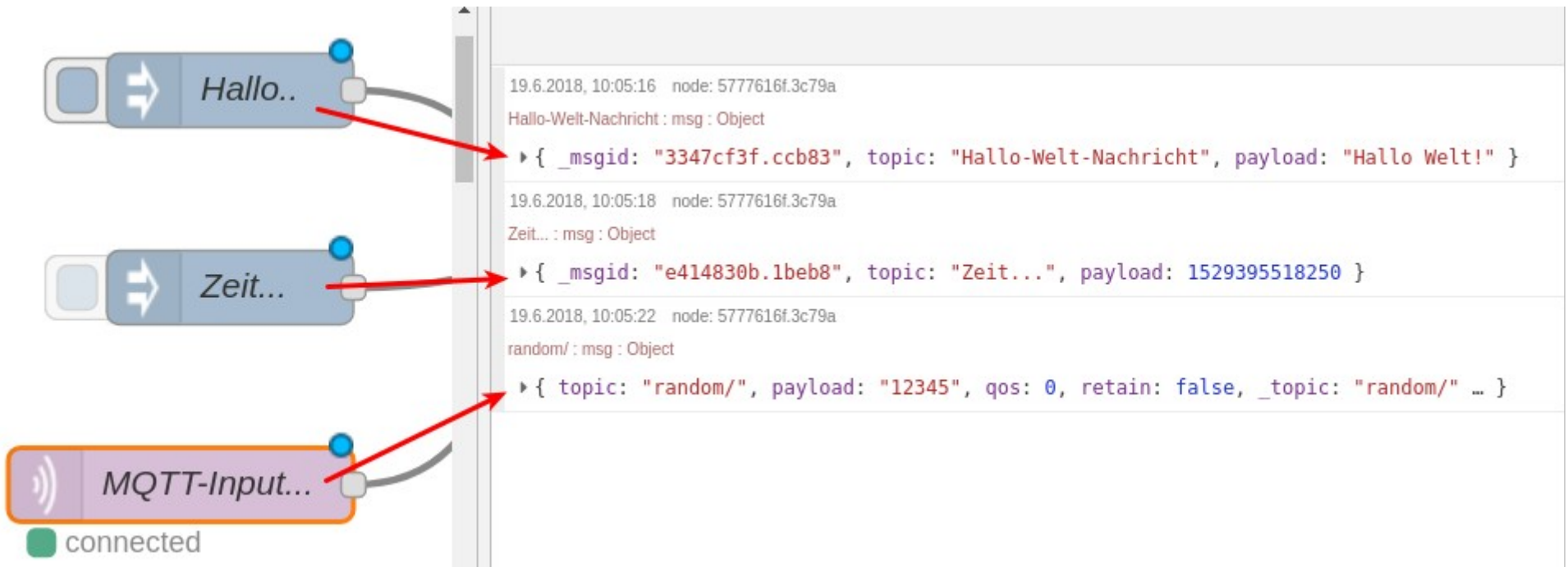
The right sidebar shows the "info" tab, which displays the help for the selected "mqtt" node. The help content is as follows:

- Information**
- Node Help**
 - Connects to a MQTT broker and subscribes to messages from the specified topic.
- Outputs**
 - payload** string | buffer
a string unless detected as a binary buffer.
 - topic** string
the MQTT topic, uses / as a heirarchy separator.
 - qos** number
0, fire and forget - 1, at least once - 2, once and once only.
 - retain** boolean
true indicates the message was retained and may be old.
- Details**
 - The subscription topic can include MQTT wildcards, + for one level, # for multiple levels.
 - This node requires a connection to a MQTT broker to be configured. This is configured by clicking the pencil icon.
 - Several MQTT nodes (in or out) can share the same broker connection if required.

At the bottom of the help panel, there is a tip: "Hold down `⌘` when you `click` on a node to also select all of its connected nodes".

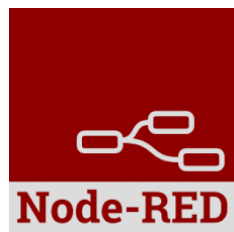
Message-Objekt...

- In der Regel immer:
 - msg.topic
 - msg.payload
- ...plus node-spezifische Eigenschaften

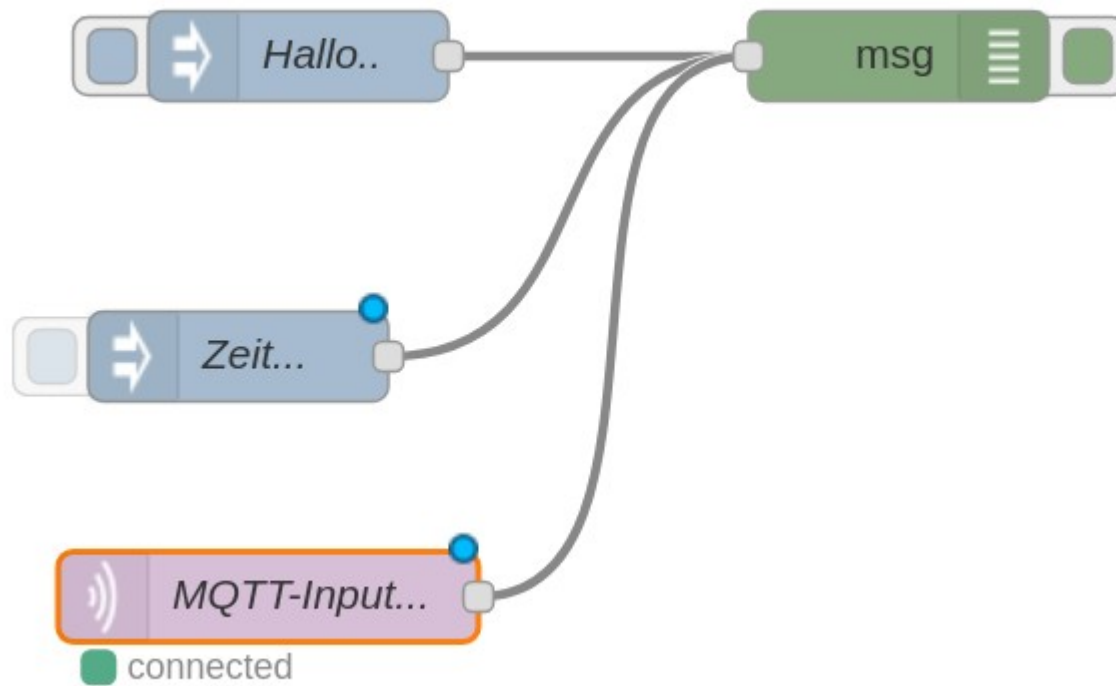


The screenshot shows the Node-RED interface with three nodes on the left and a console on the right. Red arrows point from each node to its corresponding message object in the console.

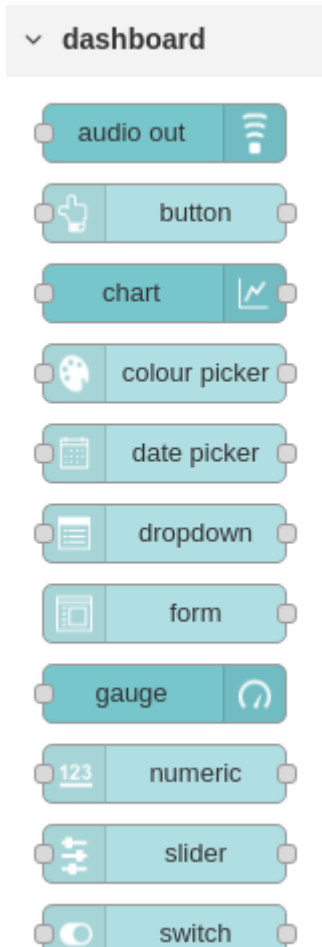
- Hallo..** node: Message object with `_msgid: "3347cf3f.ccb83"`, `topic: "Hallo-Welt-Nachricht"`, and `payload: "Hallo Welt!"`.
- Zeit...** node: Message object with `_msgid: "e414830b.1beb8"`, `topic: "Zeit..."`, and `payload: 1529395518250`.
- MQTT-Input...** node: Message object with `topic: "random/"`, `payload: "12345"`, `qos: 0`, `retain: false`, and `_topic: "random/" ...`.



Live... - ein „Hello World!“

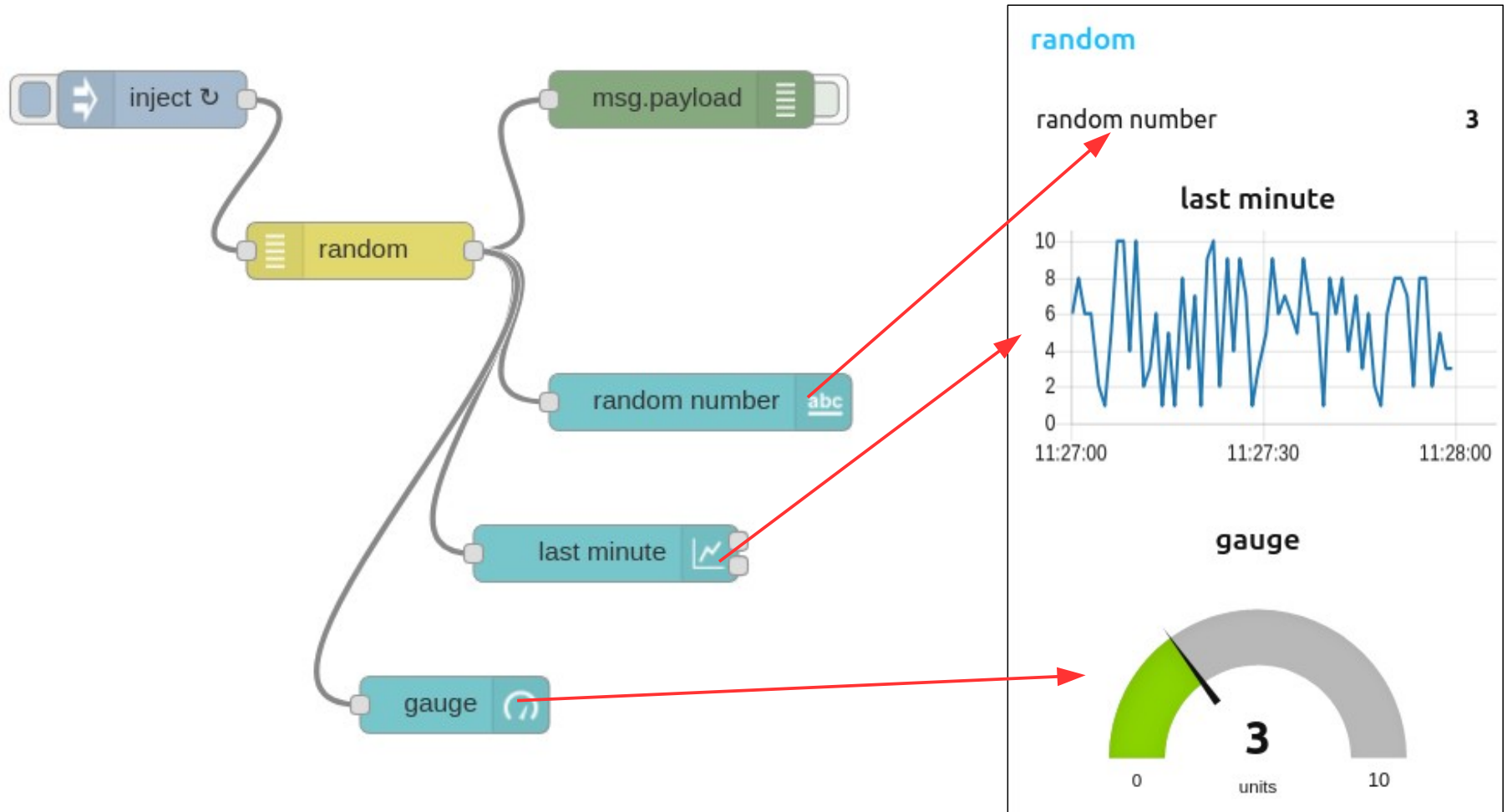


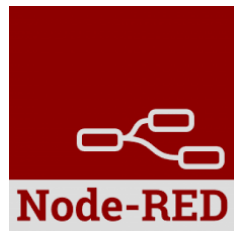
Dashboard



- ...nachinstallieren... (node-red-dashboard)
- u.a. diverse Ein-/Ausgabe-Elemente für Weboberflächen (mit Node-Red)
- <http://localhost:1880/ui/>

Live... ein kleines Dashboard





Stichworte für eigene Experimente...

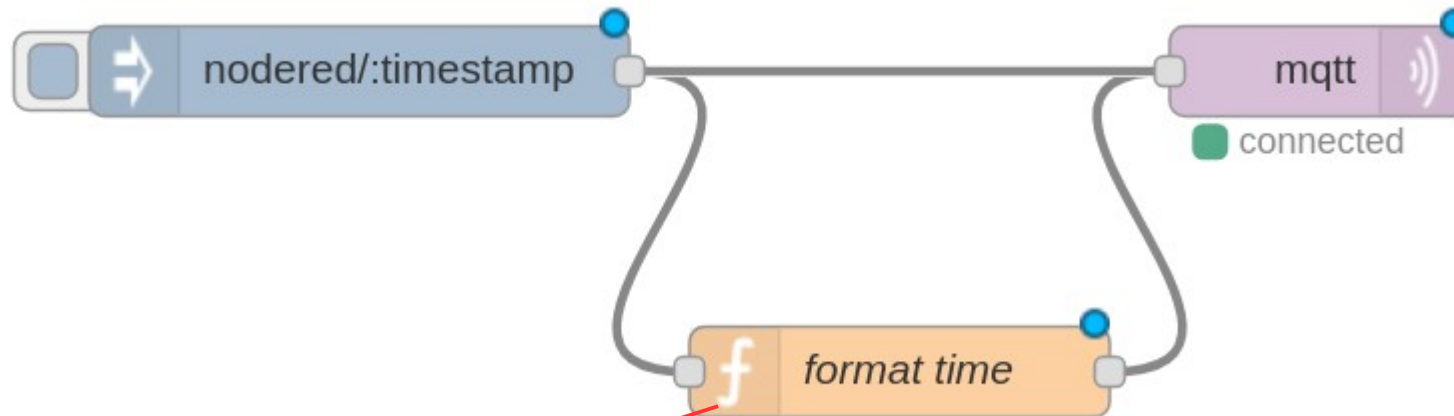
Arbeiten mit Flows

- Export/Import von Flows
- Subflows

Programmieren

- Function-Nodes (die mit eigenem JavaScript-Code „befüllt“ werden können)
- Variablenreichweite:
 - Node-weit: `context.set()`, `context.get()`, ...
 - Flow-weit: `flow.set()`, `flow.get()`, ...
 - Node-Red-weit: `global.set()`, `global.get()`, ...

Function-Node



🔧 Function

```
1 var date_time = new Date(msg.payload);  
2  
3 msg.payload="Zeitstempel: "+ date_time;  
4 msg.topic=msg.topic+"date_time/";  
5  
6 return msg;
```

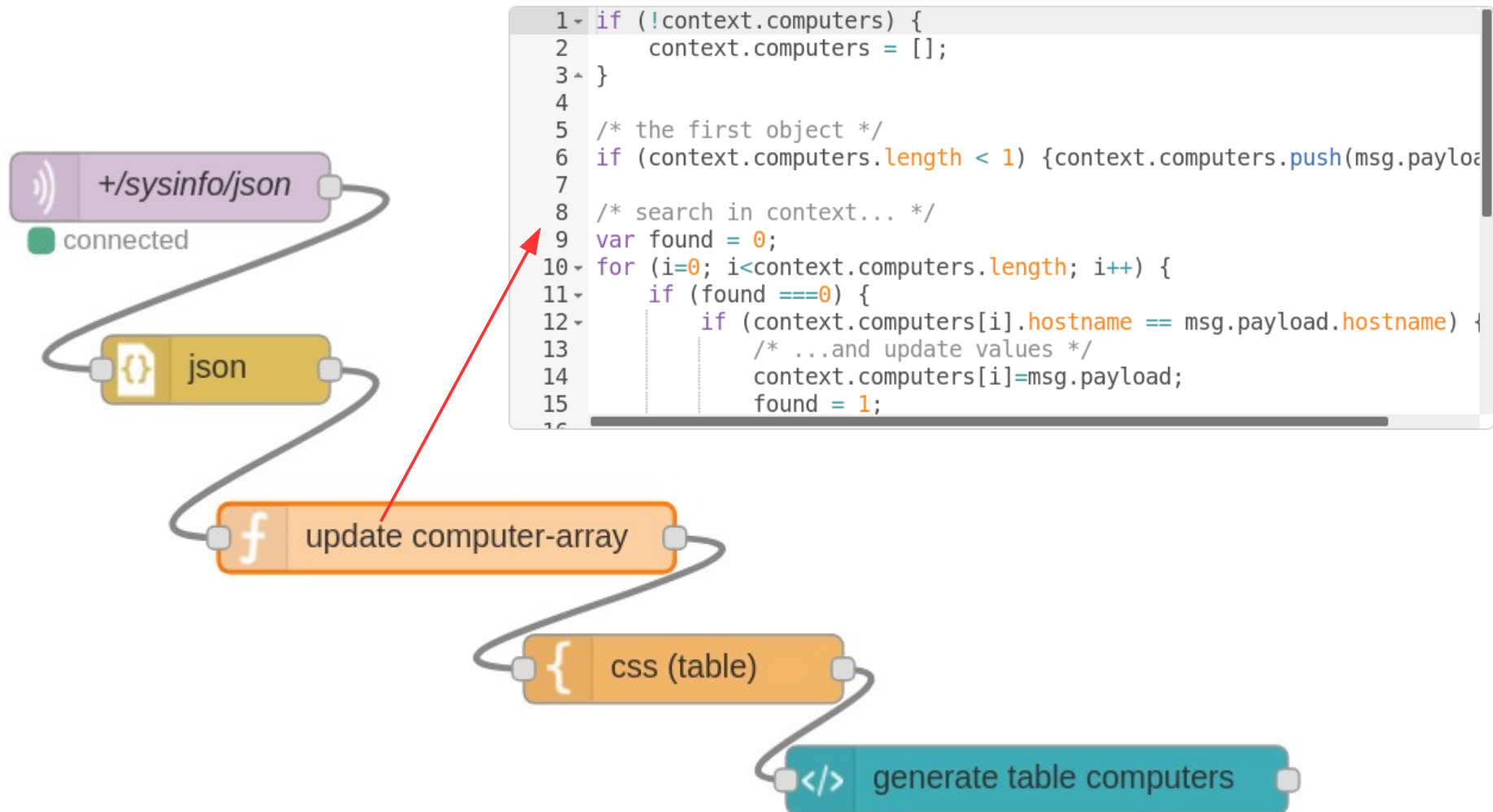

Beispiel: Computer

≡ computers

computers

hostname	last seen	uptime	load			free	ram [kByte]			swap [kByte]		processes
			1min	5min	15min		share	buffer	total	free	total	
samsung	2018/06/10 14:46:01	0 days, 1:01:35	0.13	0.12	0.27	203620	47956	167828	2051668	2983348	2988028	397
dockstar	2018/06/11 20:45:01	26 days, 14:15:26	0.09	0.07	0.05	267080	232	140944	996700	102396	102396	123
dellesus	2018/06/08 20:22:01	0 days, 0:51:00	0.25	0.15	0.27	975628	49352	587980	3980192	7807552	7807552	500
nodered	2018/06/11 20:45:02	26 days, 14:15:25	0.06	0.02	0.00	511808	48452	91192	961752	102396	102396	158
rasputin	2018/06/11 20:45:01	26 days, 14:15:27	0.09	0.17	0.15	20216	0	99636	447416	102396	102396	81
banane	2018/06/11 20:45:01	26 days, 14:15:24	0.00	0.01	0.05	589268	0	78316	993780	524284	524284	83
tuxedo	2018/06/11 20:45:01	0 days, 0:22:01	1.02	0.96	0.87	13682224	356468	77836	16301844	8388604	8388604	511
zerow	2018/06/11 20:45:01	9 days, 22:29:50	0.01	0.01	0.00	177572	24996	49180	492620	102396	102396	93

Beispiel: Computer (Flow)



Beispiel: Wettervorhersage

☰ yahoo weather


current weather

Brandenburg (Germany)

sun: 4:47 am - 9:32 pm

Mon, 11 Jun 2018 07:00 PM CEST

23°C












Cloudy

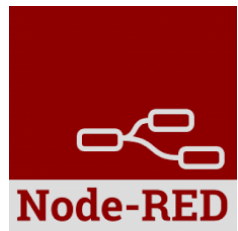
wind: 320° / 17.70km/h

REQUEST YAHOO...

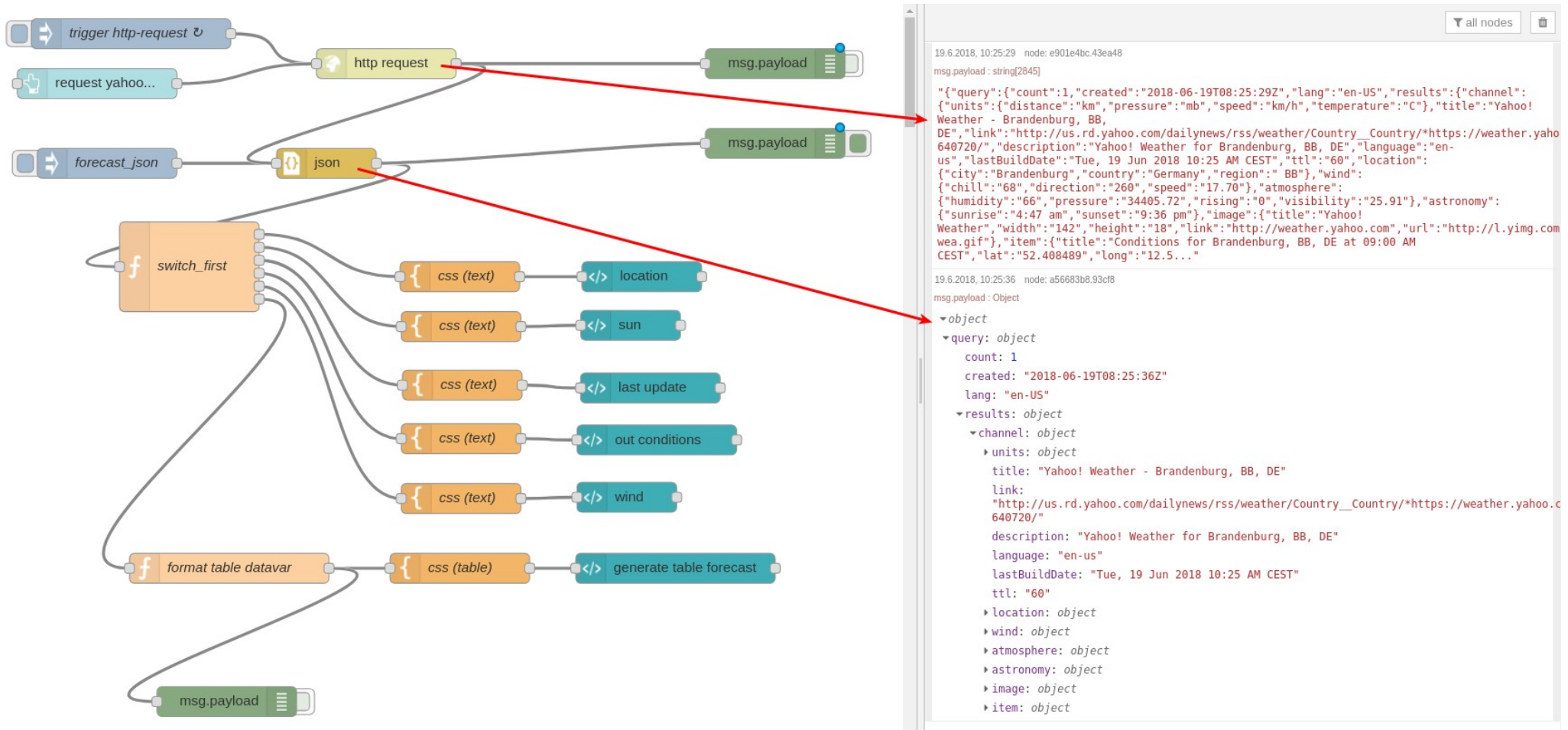
Mon, 11 Jun 2018 08:31 PM CEST

forecast weather

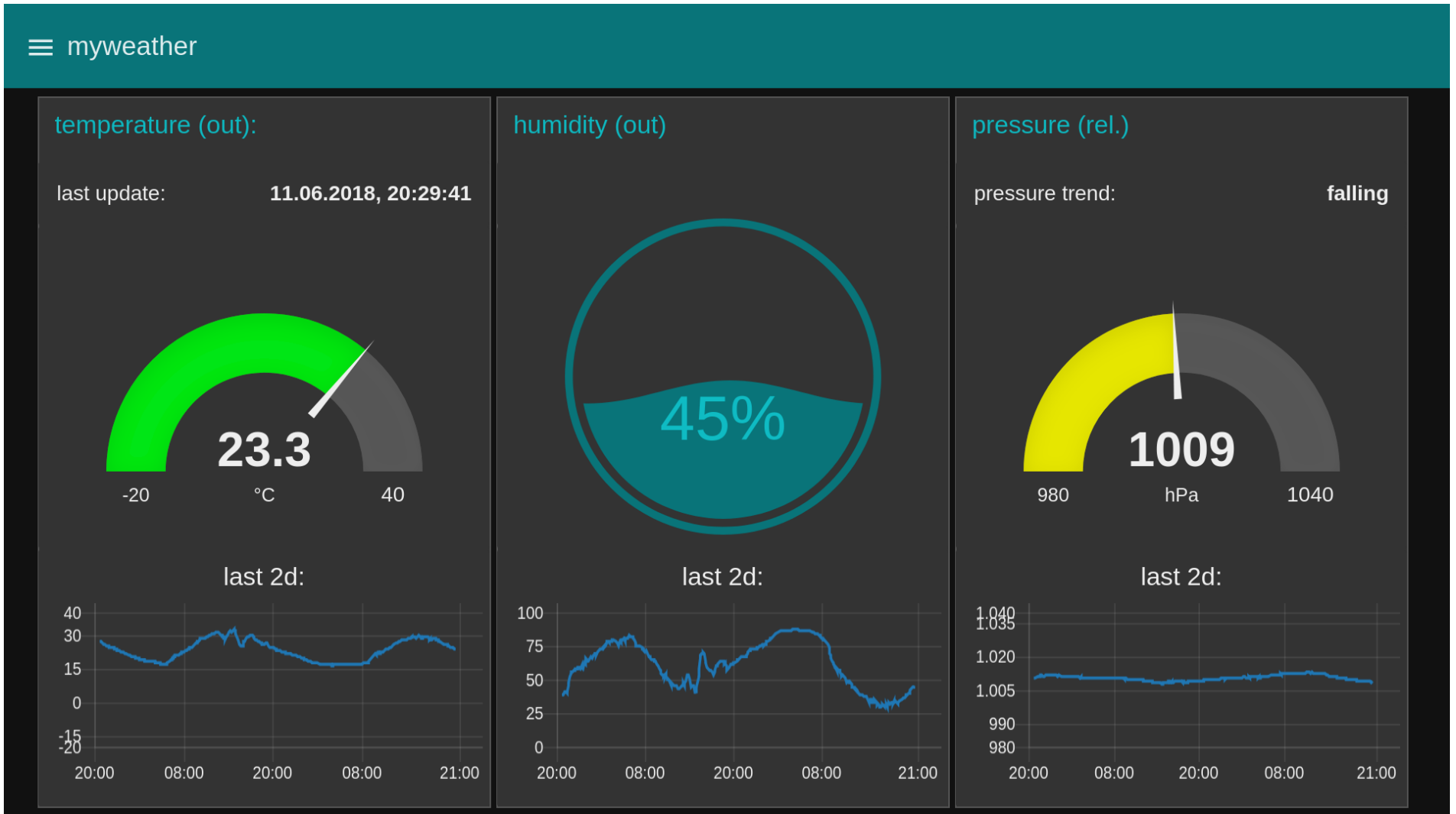
Mon, 11 Jun 2018 15°C/23°C  Mostly Cloudy	Tue, 12 Jun 2018 11°C/20°C  Mostly Cloudy	Wed, 13 Jun 2018 13°C/19°C  Mostly Cloudy
Thu, 14 Jun 2018 11°C/22°C  Partly Cloudy	Fri, 15 Jun 2018 13°C/22°C  Partly Cloudy	Sat, 16 Jun 2018 15°C/27°C  Partly Cloudy
Sun, 17 Jun 2018 16°C/23°C  Mostly Cloudy	Mon, 18 Jun 2018 15°C/25°C  Partly Cloudy	Tue, 19 Jun 2018 16°C/25°C  Partly Cloudy



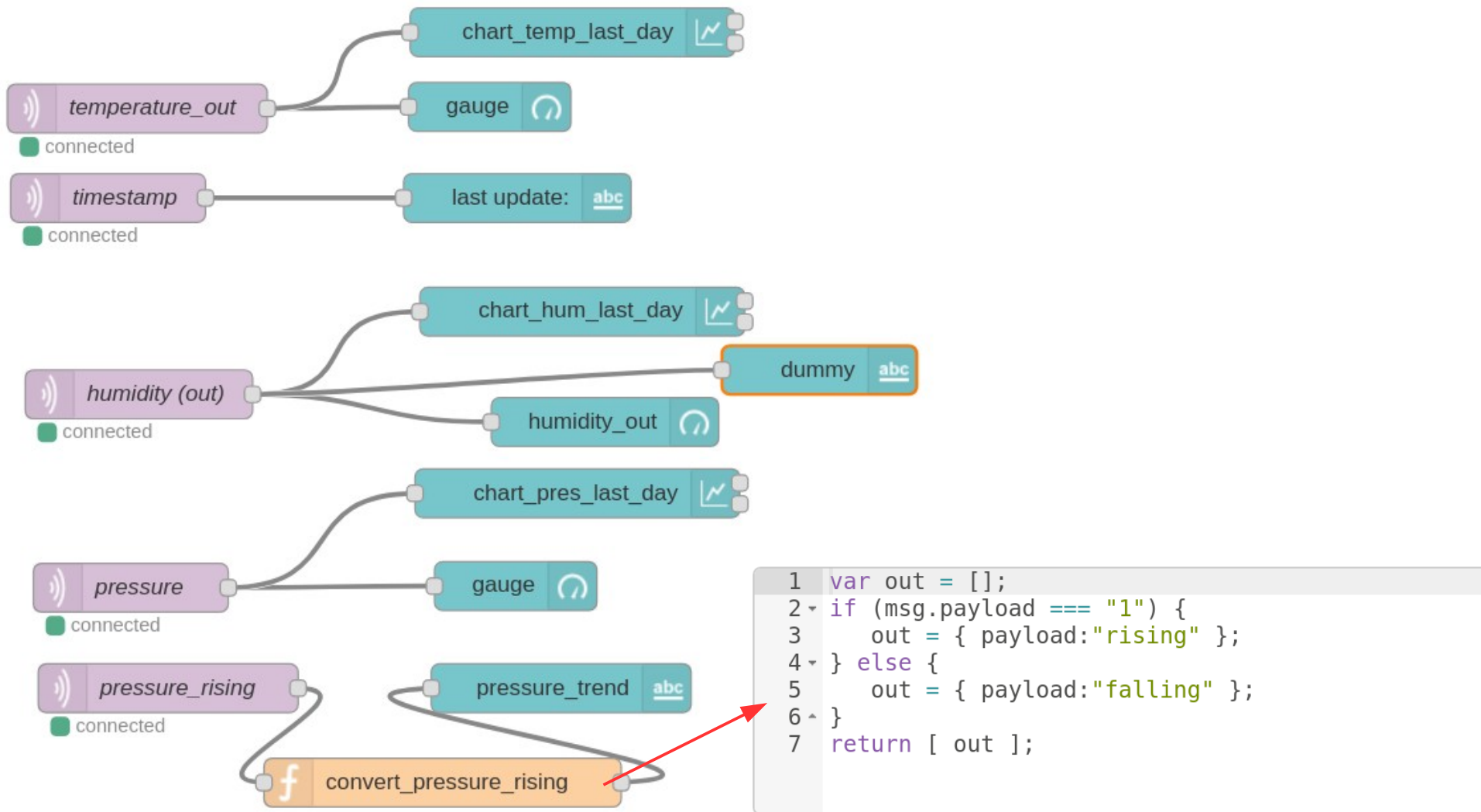
Beispiel: Wettervorhersage (Flow)



Beispiel: Wetter



Beispiel: Wetter (Flow)



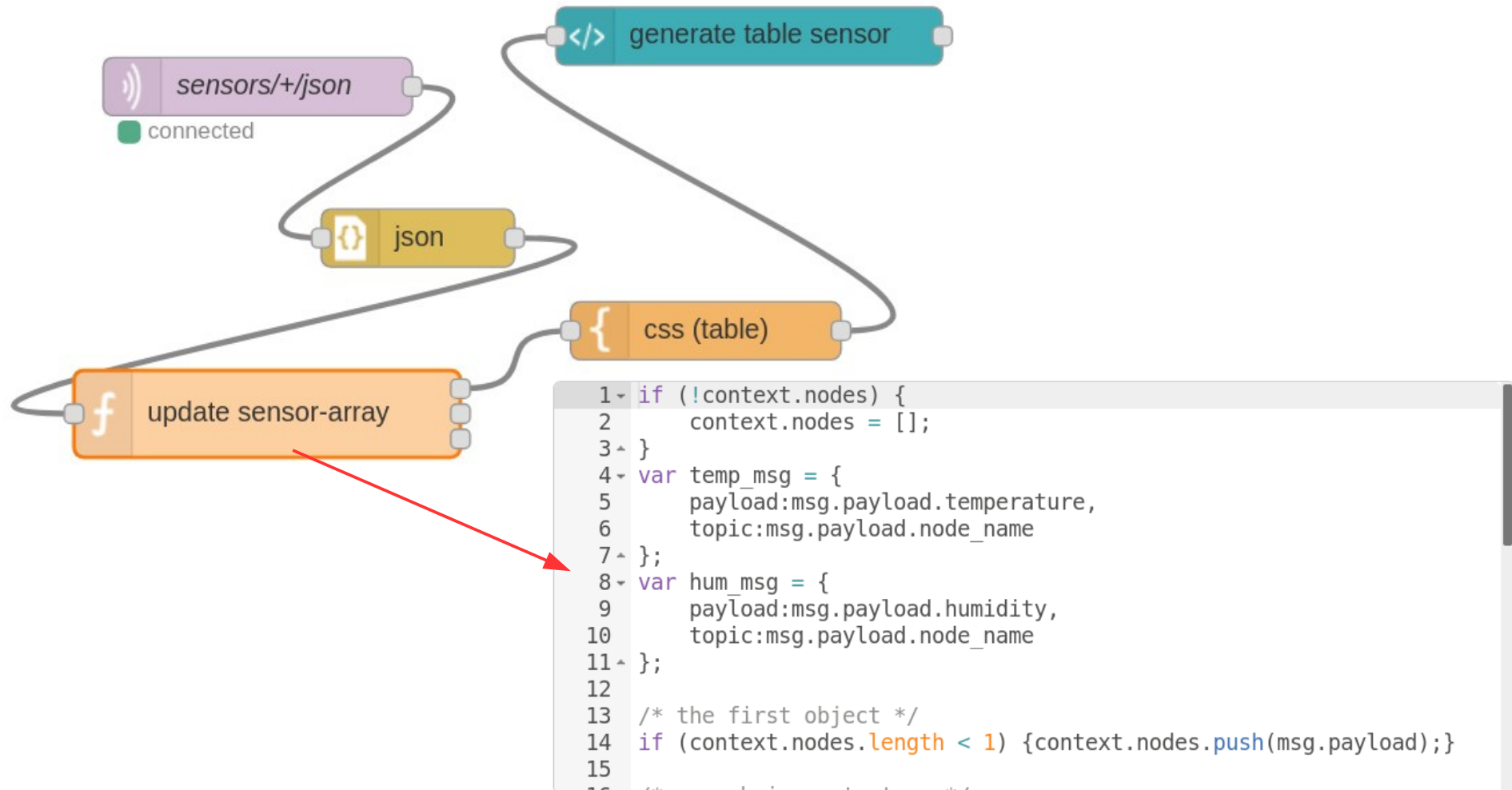
Beispiel: Sensoren

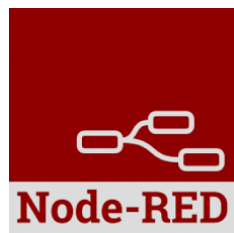
☰ sensors

sensors

name	alias	type	last seen	temp. [°C]	hum. [%]	press. [hPa]	dewp. [°C]
esp8266-9982412	Bad	dht22	2018/06/11 20:39:59	27.9	43.3		
esp8266-10441060	Gaestezimmer	dht22	2018/06/11 20:37:08	26.5	27.8		
esp8266-8671893	OLED-Device	dht22	2018/06/11 20:37:38	26.7	39.8		
esp8266-9981731	Garage	dht22	2018/06/11 20:40:08	27.2	51.4		
esp8266-12130765	Schlafzimmer	dht22	2018/06/11 20:40:18	26.9	41.4		
esp8266-12129880	testding	bme280	2018/06/11 20:37:42	25.8	46.2	1009.8	13.4

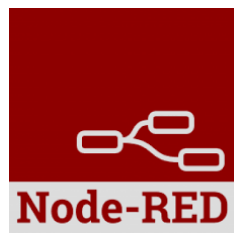
Beispiel: Sensoren (Flow)





Weiterführende Informationen

- <https://nodered.org/>
- <https://github.com/node-red/node-red>



Fragen, Feedback...?