

## join



Hiermit werden verschiedene Datenquellen zu einer Nachricht zusammengefügt.

**Node 'join' bearbeiten**

Löschen Abbrechen Fertig

Eigenschaften

Modus: Manuell

Verbinde jede msg. payload

und erstelle ein Schlüssel/Wert-Objekt

mit dem Wert von msg. topic als Schlüssel

Senden der Nachricht:

- Nach einer Anzahl von Nachrichtenteilen 2
- und bei jeder nachfolgenden Nachricht
- Bei Zeitablauf nach erster Nachricht von Sekunder
- Nach Nachricht mit msg. complete-Eigenschaft

Name: Wetter

### Modus umschalten auf Manuell

Verbinde jede legt fest, was kombiniert werden soll

Hierbei ist nur payload der Werte-Nodes interessant

und erstelle ein Schlüsselwertobjekt (muss festgelegt werden)

mit dem Wert von den topics der beiden Werte-Nodes als Schlüssel

Nach einer Anzahl von Nachrichtenteilen hier dann 2

und bei jeder nachfolgenden Nachricht hier den Haken entfernen

Name Bezeichnung der Join-Node

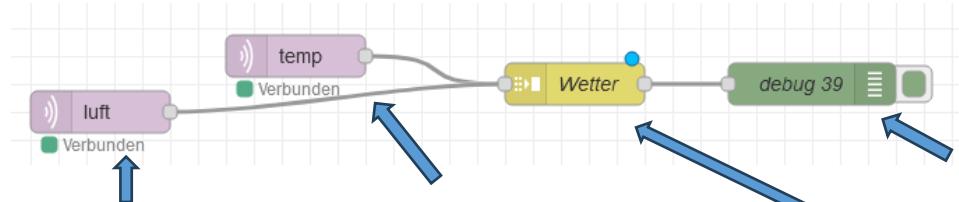
Für das folgende Beispiel sollen die Temperatur und die Luftfeuchtigkeit des Umweltsensors von ft zusammengefügt werden. Dazu muss der Sensor angeschlossen und bekannt gemacht werden. Das folgende Programm in ROBO Pro Coding wird dann gestartet und liefert jede Minute beide Werte an Node-RED

**Programmstart**

```

setze Mq auf MQTT-Client erstellen: Websockets
MQTT-Client Mq verbinden: Host "localhost" Port 1883 Benutzername " " Passwort " "
setze temp auf hole Umweltsensor TXT_M_I2C_1 Temperatur
setze luft auf hole Umweltsensor TXT_M_I2C_1 Luftfeuchtigkeit
dauerhaft wiederholen
mache MQTT-Client Mq Publish: Topic "temp" Payload temp Qos 0 Retain wahr
mache MQTT-Client Mq Publish: Topic "luft" Payload luft Qos 0 Retain wahr
warte s 60

```



16.9.2024, 08:00:21 node: debug 39  
luft : msg.payload : Object  
{} { temp: 23.62, luft: 53.639 }

**Node 'mqtt in' bearbeiten**

Löschen Abbrechen

Eigenschaften

Server: localhost:1883

Action: Subscribe to single topic

Topic: luft

**Node 'mqtt in' bearbeiten**

Löschen Abbrechen

Eigenschaften

Server: localhost:1883

Action: Subscribe to single topic

Topic: temp

**Node 'join' bearbeiten**

Löschen Abbrechen Fertig

Eigenschaften

Modus: Manuell

Verbinde jede msg. payload

und erstelle ein Schlüssel/Wert-Objekt

mit dem Wert von msg. topic als Schlüssel

Senden der Nachricht:

- Nach einer Anzahl von Nachrichtenteilen 2
- und bei jeder nachfolgenden Nachricht
- Bei Zeitablauf nach erster Nachricht von Sekunder
- Nach Nachricht mit msg. complete-Eigenschaft

Name: Wetter

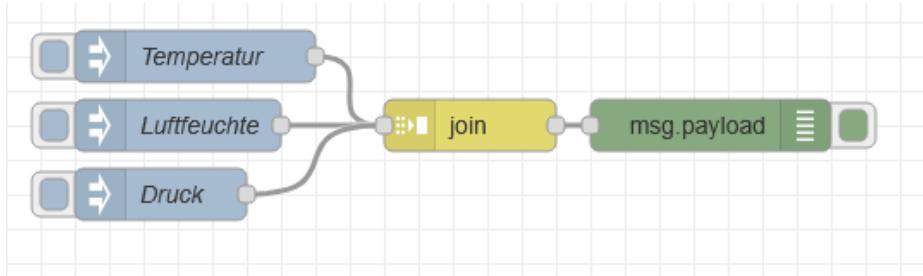
**Hinweis:** Die Ausgabe im debug-Node erfolgt unter dem Namen luft. Der topic-Name kann nicht vergeben werden. Mit einer change-Node kann der Topic-Name umbenannt werden.

Setze msg. topic to the value ft-Umwelt

16.9.2024, 08:28:44 node: debug 39  
ft-Umwelt : msg.payload : Object  
{} { temp: 23.62, luft: 53.639 }

Wetter

## Beispiel 2 ohne Fischertechnik



29.12.2025, 10:42:01 node: f9afb265.b11b7  
Druck : msg.payload : Object  
▶ { Temperatur: 10, Luftfeuchte: 56,  
Druck: 999 }

**Node 'inject' bearbeiten**

Löschen

**Eigenschaften**

Name: Temperatur

msg. payload = 10

msg. topic = Temperatur

**Luftfeuchte**

msg. payload = 56

msg. topic = Luftfeuchte

**Druck**

msg. payload = 999

msg. topic = Druck

**Node 'join' bearbeiten**

Löschen Abbrechen Fer

**Eigenschaften**

Modus: Manuell

Verbinde jede msg. payload

und erstelle ein Schlüssel/Wert-Objekt

mit dem Wert von msg. topic als Schlüssel

Senden der Nachricht:

- Nach einer Anzahl von Nachrichtenteilen 3
 und bei jeder nachfolgenden Nachricht
- Bei Zeitablauf nach erster Nachricht von Sekunden
- Nach Nachricht mit msg.complete-Eigenschaft

Name: Name

Erst nach der Eingabe des dritten Wertes erfolgt die Ausgabe aller Werte.