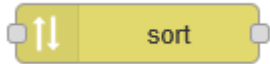


sort



Der Sortierknoten ordnet Daten aufsteigend oder absteigend an. Die Sortierung kann bei einem Array innerhalb einer Nachrichtennutzlast oder eine Nachrichtenfolge anhand der Eigenschaften sortiert werden.

Der Sortierknoten arbeitet in zwei verschiedenen Modi:

Array-Sortierung

Sortiert ein in einer Nachrichteneigenschaft gespeichertes Array. Das gesamte Array wird anhand der Elementwerte oder eines JSON-Data-Ausdrucks sortiert.



```
29.12.2025, 17:54:27 node: debug 71
msg.payload : array[5]
▶ [ 1, 2, 3, 4, 6 ]
```

Node 'inject' bearbeiten

Löschen

Eigenschaften

Name

msg. payload = {} [1,4,6,2,3]

Node 'sort' bearbeiten

Löschen

Abbrechen

Eigenschaften

Sortierung: msg. payload

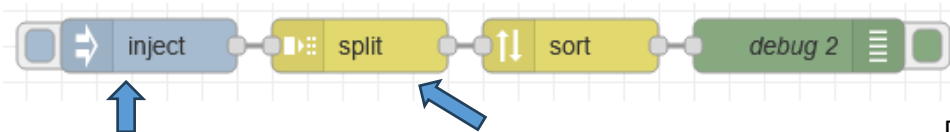
Schlüssel: Elementwert

Reihenfolge: aufsteigend

☒ als Zahlenwert

Nachrichtenreihenfolge-Sortierung

Sortiert eine Folge von Nachrichten, die eine bestimmte msg.partsEigenschaft besitzen.



Node 'inject' bearbeiten

Löschen

Abbrechen

Eigenschaften

Name

msg. payload = [{"id":2,"name":"Uwe"}, {"id":3,"name":"Paul"}, {"id":1,"name":"Axel"}]

Node 'split' bearbeiten

Löschen

Abbrechen

Eigenschaften

Aufteilung von msg.payload entsprechend dem Typ:

string / buffer

Aufteilung: a, z, \n

Der split-Knoten zerlegt das Array in einzelne Nachrichten, die jeweils nach ihrem Mengenwert sortiert werden, wobei Nachrichten mit höherer Menge zuerst angezeigt werden.

```
[{"id":2,"name":"Uwe"}, {"id":3,"name":"Paul"}, {"id":1,"name":"Axel"}]
```

Node 'sort' bearbeiten

Löschen

Abbrechen

Eigenschaften

Sortierung: Nachrichtensequenz

Schlüssel: msg. payload.id

Reihenfolge: aufsteigend

```
30.12.2025, 07:53:45 node: debug 2
msg.payload : Object
▶ { id: 1, name: "Axel" }
30.12.2025, 07:53:45 node: debug 2
msg.payload : Object
▶ { id: 2, name: "Uwe" }
30.12.2025, 07:53:45 node: debug 2
msg.payload : Object
▶ { id: 3, name: "Paul" }
```

Node 'sort' bearbeiten

Löschen

Abbrechen

Eigenschaften

Sortierung: Nachrichtensequenz

Schlüssel: msg. payload.name

```
30.12.2025, 07:59:03 node: debug 2
msg.payload : Object
▶ { id: 1, name: "Axel" }
30.12.2025, 07:59:03 node: debug 2
msg.payload : Object
▶ { id: 3, name: "Paul" }
30.12.2025, 07:59:03 node: debug 2
msg.payload : Object
▶ { id: 2, name: "Uwe" }
```