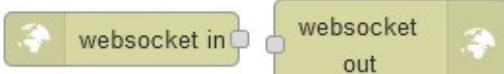


websocket



WebSocket ist ein bidirektionales Kommunikationsprotokoll, das Daten vom Client zum Server oder umgekehrt senden kann, indem der eingerichtete Verbindungskanal wiederverwendet wird. Die Verbindung bleibt bestehen, bis sie vom Client oder Server beendet wird.

Node-RED bietet zwei WebSocket-Knoten, die unterschiedlichen Zwecken dienen und in zwei Modi betrieben werden können.

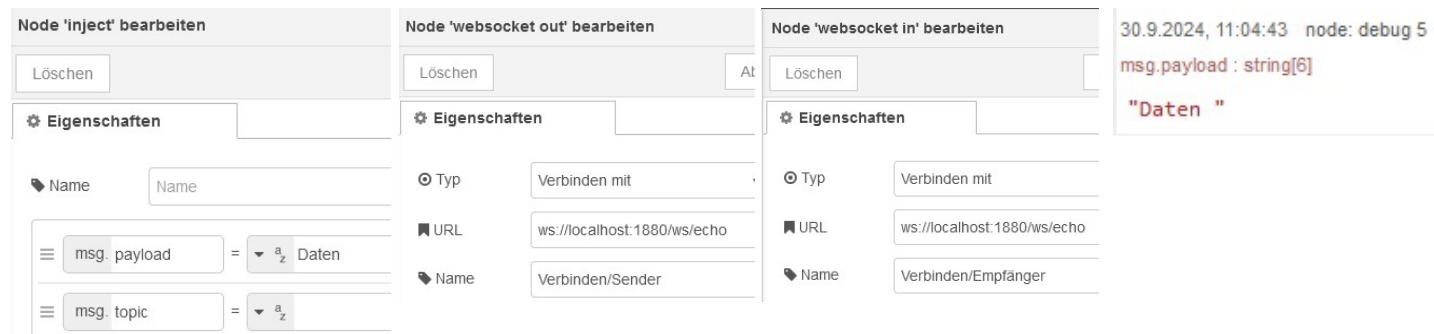
Lauschen: Hier fungiert Node-RED als WebSocket-Server und ermöglicht Remote-Clients, Verbindungen herzustellen.

- Der WebSocket-in-Knoten ist für den Empfang von Nachrichten verantwortlich, die von Remoteclients gesendet werden.
- Der WebSocket-out-Knoten erleichtert den Nachrichtenfluss, um entweder Nachrichten an einen bestimmten verbundenen Client zu senden oder Nachrichten an alle verbundenen Clients zu übertragen.

Verbinden mit: Hier fungiert Node-RED als Client und stellt Verbindungen mit Remote-WebSocket-Servern her.

- Der WebSocket-in-Knoten empfängt Nachrichten, die vom Remote-WebSocket-Server an Node-RED gesendet werden.
- Der WebSocket-out-Knoten ermöglicht dem Flow, Nachrichten an den Remote-Server zu senden.

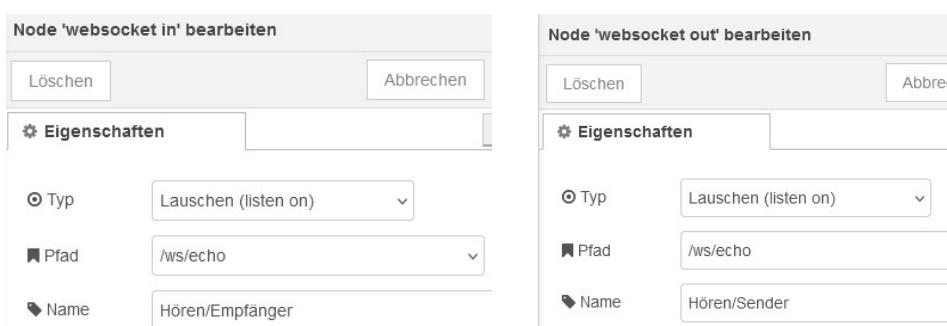
Beispiel Verbinden mit:



Beide Websockets haben den Typ *Verbinden mit*. Die URL muss mit ws:// oder wss:// beginnen und dann folgt ein WebSocket-Serververbindungslink. (im Beispiel localhost:1880/ws/echo) Für Testzwecke gibt es auch den öffentlichen WebSocket-Echoserver ws.ifelse.io

URL
wss://ws.ifelse.io

Beispiel Lauschen (basierend auf dem oberen Beispiel)



Beide Websockets haben jetzt den Typ *Lauschen*. Als Pfad reicht nun die eigentliche URL hinter der Serveradresse.

30.9.2024, 11:04:43 node: debug 5

msg.payload : string[6]

"Daten "

Da auf der einen Seite nur gelauscht wurde und das Ergebnis auf der anderen Seite wieder ausgegeben wird, empfängt die Websocket-in-Node (Verbinden mit) den Payload erneut und gibt ihn ein zweites Mal in der debug-Node aus.