

# Node-Red

- ' Call Service Node' has no entity ID
- Check external IP address and send a message with Telegram on changes

# ' Call Service Node' has no entity ID

In that case, if it does not work with any service / entity, it may be something to do with your Home Assistant server connection.

The HA server configuration node sets up the WebSocket connection, and this node has settings to capture the HA state and to save it in global context. Also there are settings for the 'Cache Auto Complete results'.

As a first step, check your NR Global Context for a variable 'homeassistant' and look in this for your HA object, and for the 'services'. This is where the node gets the service values from, and it then looks in HA for the entities.

Sometimes the Autocomplete Cache needs to be toggled (if on - off, if off - on) just to try and see if that fixes the issue.

Otherwise, the usual - have you updated HA, Node-RED addon, WebSockets, Companion to the latest versions - applies.

This is where the home assistant configuration node can be found (I just happen to have two) Your one will probably be called Homeassistant.

The 'Enable heart beat' sometimes helps with keeping a dodgy connection alive.

The 'Enable global context store' is required to keep an HA state copy in global context (see below)

The 'Cache autocomplete results' helps with speed of getting the drop down lists populated, but caching can cause issues...

where to find the HA server configuration nodes and the settings

[where to find the HA server configuration nodes and the settings](#)869×951 42.7 KB

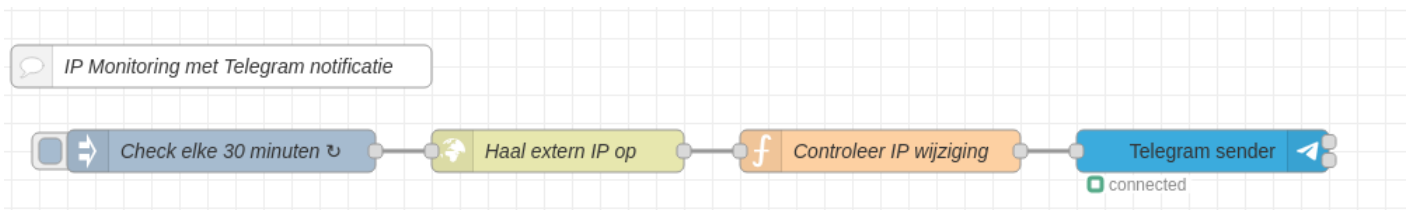
This is where to find the global context - there should be a nice homeassistant variable in there with all the states and the services - this is what is used for the service auto complete!

where to find the global context and homeassistant variable with states and services

As a workaround you can put the `entity_id` in the data field.

```
{  
  "entity_id": "number.volume"  
}
```

# Check external IP address and send a message with Telegram on changes



## What This Flow Does

This Node-RED flow periodically checks your **external IP address** and sends a **Telegram message** if it changes.

Every 30 minutes:

1. It checks your current external IP address.
2. If the IP is different from the last recorded one, it:
  - Saves the new IP.
  - Sends a Telegram message to notify you.

This is useful if your internet provider gives you a **dynamic IP** and you want to track when it changes—ideal for home servers or remote access setups. Here's how it works, step by step:

---

### 1. Inject Node: "Check every 30 minutes"

**Function:** Triggers the flow every 30 minutes.

- Also runs once at startup after a 5-second delay.
- Sends the current timestamp as `msg.payload`.

Edit inject node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖼️

📌 Name

Check elke 30 minuten

☰

msg. payload

=

⌵ ⌚

milliseconds since epoch

⌵

✕

+ add

inject now

☒ Inject once after

5

seconds, then

🔄 Repeat

interval

⌵

every

30

⬆️ ⬆️

minutes

⌵

📄

☐ Enabled

---

## 2. HTTP Request: "Get external IP"

**Function:** Makes a GET request to `https://api.ipify.org`.

- This service returns your current **public (external) IP address** in plain text.
- The response becomes the new `msg.payload`.

Edit http request node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

☰ Method

GET

▼

🌐 URL

https://api.ipify.org

Payload

Ignore

▼

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

⬅ Return

a UTF-8 string

▼

☰ Headers

+ add

📌 Name

Haal extern IP op

📄

☐ Enabled

### 3. Function Node: "Check IP address change"

**Function:** Compares the new IP address to the previously stored one.

- Retrieves the **last known IP** from Node-RED's flow memory ( `flow.get('previousIp')` ).
- If the new IP is **different**, it:
  - Stores the new IP ( `flow.set('previousIp', currentIp)` ).
  - Sets a message like `Nieuw extern IP-adres gedetecteerd: <new IP>` .
  - Passes this message along.
- If the IP hasn't changed, it stops (returns `null` ).

**Edit function node**

Delete

Cancel

Done

Properties

Name

Controleer IP wijziging

Setup

On Start

**On Message**

On Stop

```
1 // Haal het huidige IP-adres op uit de payload
2 const currentIp = msg.payload;
3
4 // Haal het vorige IP-adres op uit de flow context
5 const previousIp = flow.get('previousIp') || '';
6
7 // Vergelijk het huidige IP met het vorige IP
8 if (currentIp !== previousIp) {
9     // Bewaar het nieuwe IP in de flow context
10    flow.set('previousIp', currentIp);
11
12    // Stuur een bericht met het nieuwe IP
13    msg.payload = 'Nieuw extern IP-adres gedetecteerd: ' + currentIp;
14    return msg;
15 } else {
16     // IP is niet veranderd, stop de flow
17     return null;
18 }
19
```

Enabled

```
// Haal het huidige IP-adres op uit de payload
const currentIp = msg.payload;
```

```
// Haal het vorige IP-adres op uit de flow context
const previousIp = flow.get('previousIp') || '';

// Vergelijk het huidige IP met het vorige IP
if (currentIp !== previousIp) {
  // Bewaar het nieuwe IP in de flow context
  flow.set('previousIp', currentIp);
  // Stuur een bericht met het nieuwe IP
  msg.payload = 'Nieuw extern IP-adres gedetecteerd: ' + currentIp;
  return msg;
} else {
  // IP is niet veranderd, stop de flow
  return null;
}
```

---

## 4. Telegram Sender

**Function:** Sends the notification via your configured Telegram bot if there's a new IP address.



Edit sender node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

📡 Bot

██████████

▼

✎

+

🏷️ Name

Name

🔔 Send errors to second output

☐

📄

☐ Enabled

---

## 5. Telegram Bot Configuration

- Connected to a bot named `mybot`.
- Sends messages to a specified user or chat ID.

---