

Home Assistant

Guide

Sådan kommer du i gang med



Version 1.0
Af Anders Lukic

Introduktion

I denne guide vil jeg forsøge at beskrive, hvad Node-RED (NR) er og forklare, hvordan det kan bruges sammen med Home Assistant (HA) til at lave automatiseringer af dine smart enheder.

Guiden er først og fremmest henvendt til dem, der overvejer, om de skal lære NR, eller som lige er gået i gang med at lave de første automatiseringer i NR.

Der er forhåbentligt også inspiration at hente for dem, der allerede bruger NR sammen med HA, da jeg viser eksempler på automatiseringer, som jeg selv anvender i NR.

Guiden beskriver ikke, hvordan NR installeres eller al dens funktionalitet, men fokuserer på de elementer, som man oftest vil have behov for at anvende til at begynde med.

Se følgende links for vejledning i installationen af NR via en addon i HA:

- [Installationsvejledning til Node-RED addon i Home Assistant](#)

Hvad er Node-RED?

NR er et såkaldt flow-baseret programmeringsværktøj, som er udviklet af IBM til brug for programmering af primært Internet of Things (IoT) systemer.

I modsætning til traditionelle programmeringsværktøjer er NR kendetegnet ved, at de handlinger, som f.eks. en automatisering skal udføre, laves visuelt ved at knytte forskellige elementer (**noder**) sammen i en samlet proces (**flow**). Dette er vist her i eksemplet:

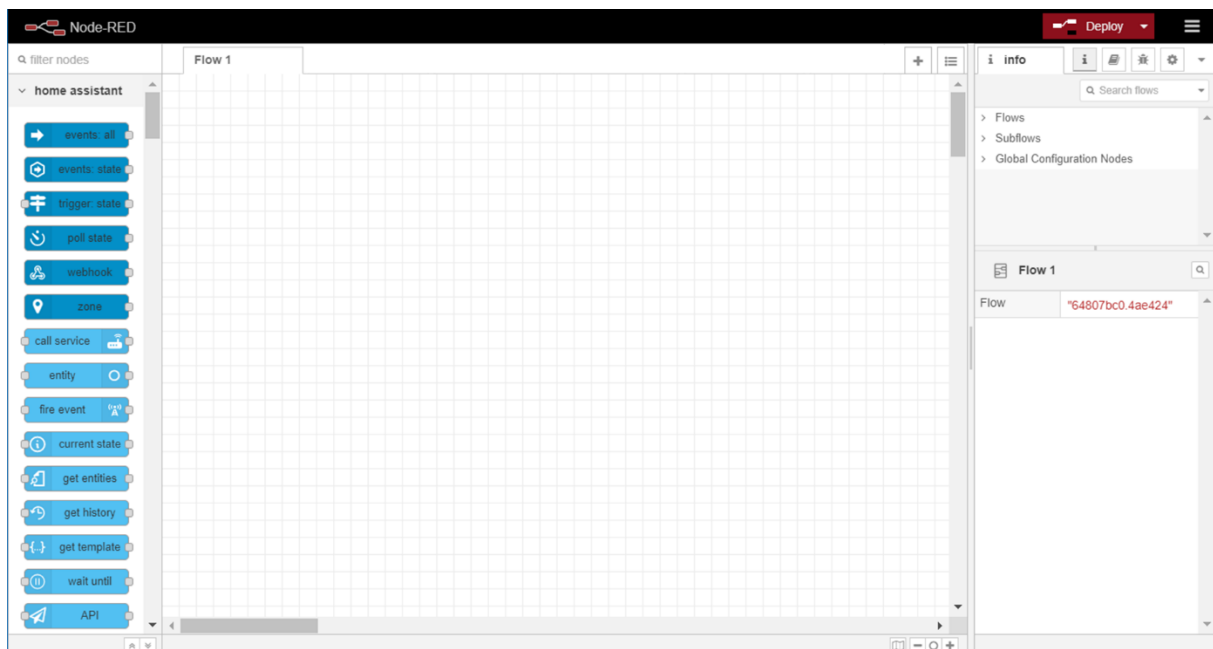


Figur 1: Eksempel på simpelt flow i node-RED

Det viste flow består af tre **noder**, som hver time kontrollerer batteristatus på et batteritryk (f.eks. en Hue Dimmer Switch eller IKEA TRÅDFRI fjernbetjening) og i tidsrummet 7-22 sender en notifikation til en mobiltelefon, hvis der er mindre end 30% batteri tilbage.

De enkelte noder, der er anvendt i eksemplet ovenfor, vil blive beskrevet i det næste afsnit, men først en kort introduktion til brugergrænsefladen i NR.

NR afvikles i en browser og ser således ud, når det er installeret som add-on i HA:



Figur 2: Brugergrænsefladen i node-RED

Node palette

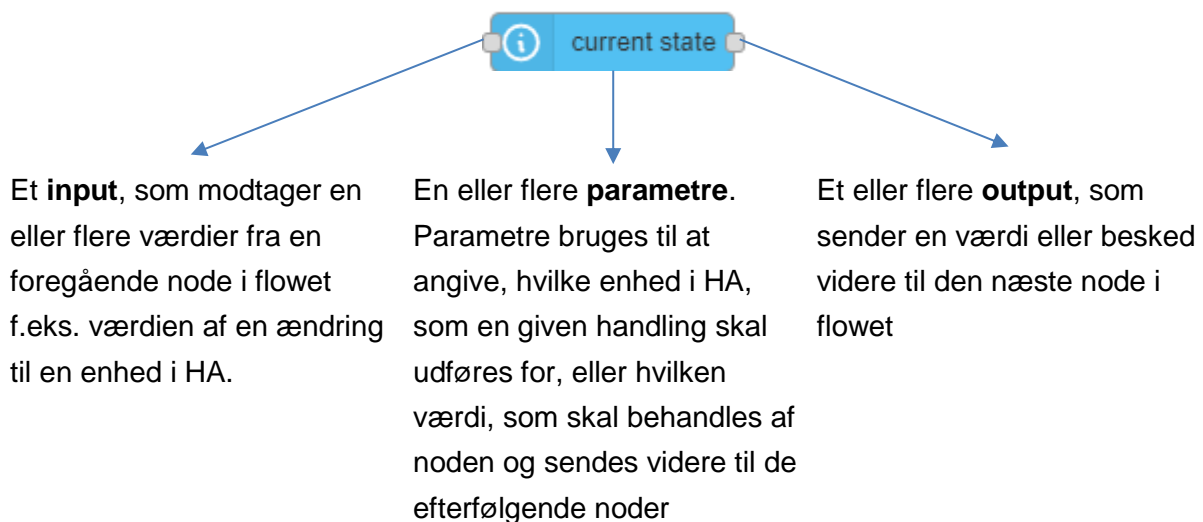
I venstre side af skærmen findes en oversigt (palette) med de forskellige **noder**, som man kan bruge til at bygge sin automatisering med. Hver node er beregnet til at gennemføre en

eller flere specifikke handlinger, f.eks. hente værdien fra en enhed i HA (eksempelvis om en person er hjemme eller ej) eller starte en service i HA (tænd lys, spil musik mv.).

Hvis du har installeret NR addon til HA, vil du øverst i paletten allerede have de nødvendige noder til at kunne bygge automatiseringer, der styrer dine enheder i HA.

En node kan ses lidt som en byggeklods for din automatisering, hvor man sætter flere handlinger sammen i et flow for at opnå det ønskede resultat.

Hver node vil som udgangspunkt bestå af følgende dele:



Det er ikke alle noder, som f.eks. har et input. Noder uden input vil typisk være dem, du starter med i dit flow. Det kan f.eks. være en node, der reagerer på en specifik hændelse i HA på samme måde, som en trigger fungerer i HA's automatiseringer.

Der findes også få noder, som ikke har et output og de vil således afslutte et flow, men man kan også afslutte flows med noder, der har output, men som bare ikke forbindes med en ny node.

Forskellige noder fungerer således som henholdsvis *triggers*, *conditions* og *actions*, som du måske kender fra dine automatiseringer i HA.

I et flow kan du have lige så mange noder, som du ønsker, og derfor kan du bruge mange forskellige *triggers* og *conditions* i det samme flow og samle relaterede automatiseringer i ét flow. Det giver dig et godt samlet overblik over f.eks. al din lysstyring.

Flow designer

Den midterste del af skærmen benyttes til at designe automatiseringer i forskellige **flows**, der vises i hver sin fane. Man kan tilføje et nyt flow ved at klikke på "+" ikonet.

Man *trækker* de nødvendige noder ind fra node **paletten** i venstre side og forbinder de enkelte noders **input** og **output** ved at trække pile mellem dem.

Konfigurationen af den enkelte node foretages ved at *dobbeltklikke* på noden, hvorefter man kan indtaste de **parametre**, som er nødvendige for at bestemme, hvordan handlingen skal udføres.

Nedenstående eksempel er fra den første node i flowet, som er vist i Figur 1 ovenfor. Det er en *Poll State* node, der henter en værdi fra et batteritryk i HA hver time:

The screenshot shows the 'Edit poll state node' configuration window. It includes a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' tab is active. The configuration fields are as follows:

- Name:** Check battery every hour
- Server:** Home Assistant
- Entity ID:** sensor.smart_button_battery_level (labeled with a blue '1')
- Update Interval:** 1 hours (labeled with a blue '2')
- If State:** < 30% (labeled with a blue '3')
- State Type:** Number
- ☐ Output on Changed
- ☐ Output on Connect
- ☐ Expose to Home Assistant
- ☐ Enabled

Figur 3: Eksempel på poll state node

I denne node, er der angivet følgende:

- 1) Hvilken enhed i HA, der skal hentes værdi (state) fra. Navne på enheder hentes fra HA, så man kan vælge dem på en liste, hvis man starter med at skrive navnet på enheden, f.eks. "sensor."
- 2) Hvor ofte værdien skal hentes fra HA
- 3) Hvilken betingelse, der skal være opfyldt, før at output sendes videre og flowet fortsætter

Deploy

I højre side af skærbilledet er der øverst en **Deploy** knap, som man bruger til at aktivere sin automatisering, når flowet er klar til at blive testet. Ændringerne gælder med det samme, så du slipper for at genstarte HA eller genindlæse dine automatiseringer, hvis de er lavet i YAML.

Menu og sidebar

Ved siden af knappen findes **menuen**, hvor man bl.a. kan eksportere og importere flows, så man kan hente inspiration fra eller dine sine automatiseringer med andre brugere.

Man kan også installere flere paletter (under menupunktet *Manage palette*) med ekstra noder med yderligere funktionalitet, f.eks. for at integrere med Sonos, DeCONZ mv. direkte i NR.

Under menuen findes et sidepanel (**sidebar**), hvor man bl.a. kan få vejledning til brugen af de forskellige noder samt et *debug* vindue, som man kan bruge til fejlsøgning. Mere om dette senere.

De mest anvendte noder

Dette afsnit beskriver de noder, som bruges til integrationen med HA samt øvrige relevante noder, som man ofte vil bruge i sine flows. I beskrivelsen er det endvidere angivet, om noden er demonstreret i et eller flere af de efterfølgende eksempler på automatiseringer.

Home Assistant noder

HA's Node-RED addon indeholder i alt 14 forskellige noder, som man kan anvende til at bygge automatiseringer, der anvender enheder og funktionalitet i HA.

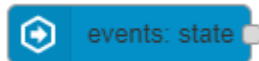
Hver node er kort beskrevet nedenfor med eksempler på, hvornår de kan bruges:



Events: all noden anvendes til at reagere på en hvilken som helst hændelse i HA. **BEMÆRK:** Der bør altid angives en "event_type", da noden ellers vil reagere på alle HA hændelser, hvilket kan overbelaste HA og NR.

Noden kan f.eks. benyttes til at reagere på en hændelse i DeCONZ, Hue, ZHA eller tilsvarende og benyttes ofte til at opfange tryk på knapper på forskellige batteritryk.

Noden er anvendt i efterfølgende eksempel #1 og 2



Events: state noden anvendes til at reagere, når en HA enhed har haft en given status i en nærmere defineret periode.

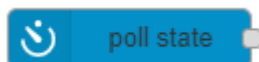
Dette kan f.eks. benyttes til at starte et flow, der slukker lyset 5 minutter efter sidste bevægelse er registreret af en bevægelsessensor.



Trigger: state noden anvendes som navnet antyder som trigger, når en eller flere enheder i HA skifter værdi (state). Der kan tilføjes en eller flere betingelser, som skal være opfyldt, før flowet fortsætter. Det kan f.eks. være state værdien for enheden før ændring og efter ændringen eller en hvilken som helst anden værdi (attribut) for enheden.

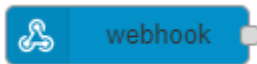
Noden kan f.eks. benyttes til at reagere på, at en sensor (f.eks. en dørsensor) skifter værdi til åben.

Noden er anvendt i efterfølgende eksempel #1



Poll state noden anvendes til løbende at hente en værdi fra en HA enhed efter et nærmere defineret tidsinterval.

Noden kan f.eks. benyttes til at hente gps koordinater fra en tracker hvert 5. minut og bruge dem i en automatisering til at beregne, hvor langt væk hjemmefra en given person er.



Webhook noden anvendes til at reagere på HA webhook hændelser.

Noden kan f.eks. benyttes til at integrere med eksterne websider, som kan sende data til HA såsom IFTT servicen.



Zone noden anvendes til at reagere på, at én eller flere personer ankommer til eller forlader en given zone.

Noden kan f.eks. benyttes til at starte en automatisering når en person forlader arbejdet eller ankommer til hjemme zonen.



Call service noden anvendes til at starte en hvilket som helst service i HA på samme måde som actions i en HA automatisering.

Noden kan f.eks. benyttes til at tænde lys eller sende en besked til en mobiltelefon og alt andet, som er udstillet som en service i HA.

Noden er anvendt i efterfølgende eksempel #1, 2 og 3



Entity noden anvendes til at oprette en ny eller ændre en eksisterende enhed (sensor) med tilhørende værdier i HA.

Noden kan f.eks. benyttes til at lave en ny sensor med data, som hentes fra en web-side.

Noden er anvendt i efterfølgende eksempel #4



Fire event noden anvendes til at sende en hændelse til HA, som kan anvendes som trigger i en HA automatisering eller et andet NR flow.

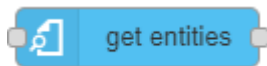
Noden kan f.eks. benyttes til at starte et NR flow fra et andet flow ved at sende en hændelse om f.eks. at en værdi er ændret for en HA enhed. Samme effekt kan opnås ved at ændre værdien med en *entity* node.



Current state noden anvendes på samme måde som *poll state* noden til at hente data fra en given HA enhed, men *current state* noden har både et input og output og kan derfor bruges overalt i et flow til at hente værdier for en HA enhed.

Noden kan f.eks. anvendes til at hente data om solens stilling på himlen og bruge den oplysning i et flow til at vælge, om og hvor meget lys, der skal tændes ved registrering af bevægelse.

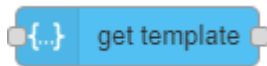
Noden er anvendt i efterfølgende eksempel #2



Get entities noden anvendes til at hente værdier fra flere forskellige HA enheder, f.eks. enheder af en bestemt type. Den kan også benyttes til at tælle antallet af enheder i HA, der opfylder bestemte betingelser, f.eks. antallet af åbne døre og vinduer.

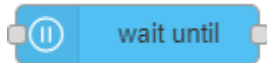
Noden kan f.eks. anvendes til at hente batteristatus for alle batteridrevne enheder og bruge det i et flow til at sende besked om enheder med lavt batteri.

Noden er anvendt i efterfølgende eksempel #3



Get template noden kan bruges til at afvikle Jinja2 template kode og bruge resultatet i en NR automatisering.

Noden kan f.eks. anvendes, hvis du ønsker at konvertere en eksisterende HA automatisering, der bruger templates, til NR.



Wait until noden kan anvendes til at sætte dit flow på pause indtil en bestemt betingelse er opfyldt.

Noden kan f.eks. anvendes til at vente med at starte en automatisering indtil en mobilenhed er tilkoblet til Wi-Fi, når en person kommer til hjemmezonen.



API noden kan anvendes til at kommunikere direkte med HA's API service og benytte funktionalitet, der ellers ikke findes en HA standard service til.

Noden kan f.eks. anvendes til at hente data fra HA's konfigurationsfiler eller fejlbeskeder fra logfilen. Se evt. mere i HA udvikler dokumentationen: <https://developers.home-assistant.io/docs/api/rest>

Øvrige noder

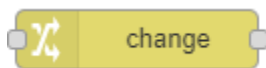
Ud over de HA specifikke noder indeholder NR en lang række andre noder, som man kan anvende i sine automatiseringer. Der er mange at vælge imellem, men dem, som man ofte vil have brug for, er beskrevet her:



Inject noden anvendes til at starte et flow på et givet tidspunkt eller indsætte en værdi, der sendes til den næste node i flowet.

Noden kan f.eks. anvendes til starte en automatisering hver dag på et bestemt tidspunkt eller efter et defineret interval af antal enten timer, minutter eller sekunder.

Noden er anvendt i efterfølgende eksempel #3 og 4



Change noden anvendes til at ændre indhold af input, f.eks. ved at flytte, slette eller ændre data i det modtagne input inden det sendes videre til den næste node i flowet.

Noden kan f.eks. anvendes til at flytte en værdi fra inputtet til et nyt output, så den oprindelige data ikke bliver overskrevet eller fjerne dele af inputtet, f.eks. ved at søge og erstatte tekst.

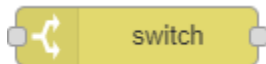
Noden er anvendt i efterfølgende eksempel #4



Template noden anvendes til definere output, så det følger et specifikt format.

Noden kan f.eks. benyttes til at kombinere værdier fra flere forskellige HA enheder i et samlet output, der kan anvendes af de efterfølgende noder til f.eks. at sende en notifikation. Den kan også bruges til at lave en JSON liste med variabler, der skal anvendes på et senere tidspunkt i automatiseringen.

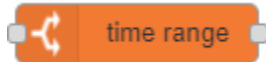
Noden er anvendt i efterfølgende eksempel #2 og 3



Switch noden har som funktion som et sporskifte for et tog. Noden kan have mange forskellige output, så et flow kan fortsætte i forskellige retninger alt afhængigt af hvilken værdi, der modtages som input.

Den kan f.eks. benyttes til at lave forskellige handlinger, når der er nogle hjemme og når alle er ude. Eller til at udføre forskellige handlinger alt afhængigt af hvilken knap, der trykkes ned på et batteritryk.

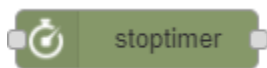
Noden er anvendt i efterfølgende eksempel #1 og 2



Time range noden anvendes til kontrollere, om den aktuelle tid ligger indenfor et defineret tidsinterval.

Noden kan f.eks. anvendes til at lave forskellige automatiseringer om aftenen og om dagen.

Noden er anvendt i efterfølgende eksempel #2



Stoptimer noden anvendes til stoppe udførelsen af det efterfølgende flow i en nærmere defineret periode angivet i enten antal timer, minutter eller sekunder. Nedtællingen starter forfra, hvis noden modtager et nyt input inden nedtællingen er færdig, f.eks. hvis en ny hændelse registreres i HA

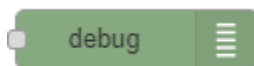
Noden kan f.eks. benyttes til at slukke lyset efter en nærmere fastsat periode uden registrering af bevægelse.

Noden er anvendt i efterfølgende eksempel #1



Function noden anvendes til at køre JavaScript kode, der kan anvendes til at behandle, formatere og ændre data inden det sendes videre til næste node. JavaScript er et selvstændigt programmeringssprog, så der er næsten uanede muligheder, hvis man i forvejen kender til det eller finder eksempler på kode, som kan genbruges / tilrettes. Læs evt. mere om JavaScript på [W3Schools](https://www.w3schools.com/).

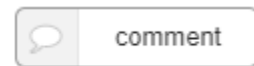
Noden er anvendt i efterfølgende eksempel #4



Debug noden anvendes til at sende output til debug vinduet, så man kan f.eks. se hvilke data, der sendes som output og lave fejlsøgning.

Noden er rigtig god at benytte, mens man bygger sine flows, så man kan se, hvilke data, som sendes videre fra én node til en anden. Det gør det meget lettere at finde og rette evt. fejl i automatiseringen.

Noden er anvendt i efterfølgende eksempel #1 og 4



Comment noden anvendes til at tilføje kommentarer til dit flow. Det kan være relevant at anvende i større flows, der f.eks. indeholder flere forskellige triggere, så man hurtigt kan se, hvad formålet er med den enkelte del af flowet. Man bør også tilføje overordnede kommentarer til hele flowet til kort at beskrive formål og funktion.

Noden er anvendt i efterfølgende eksempel #1 og 4

Ofte kan det ønskede resultat i et flow opnås på flere forskellige måder, så der er ingen rigtig eller forkert måde at opbygge sine flows på. I det efterfølgende afsnit med eksempler, vil jeg vise, hvordan jeg har valgt at bruge flest af ovenstående noder i mine egne automatiseringer til styring af lys, dørklokke, sende notifikation ved lavt batteriniveau og hente data fra en webside.

YAML versus JSON

Nu bliver det en lille smule teknisk, men det er vigtigt for at kunne forstå, hvordan man f.eks. kan konvertere en eksisterende HA automatisering til NR.

Automatiseringer i HA er skrevet i det kodesprog, der hedder YAML mens NR bruger JSON (se definitioner i [HA Palør](#)) til f.eks. at angive data parametre i en *call service* node.

Hvis du har brugt YAML kode før, er du nok vant til, at holde styr på indrykninger i din kode og bruge bindestreger til at lave lister med forskellige værdier osv. I JSON skal koden skrives lidt anderledes, som vist i eksemplet, hvor der sættes en farvekode og lysstyrke for en pære i hhv. YAML og JSON format:

YAML kode

```
brightness: 33
rgb_color:
  - 255
  - 27
  - 44
```

JSON kode

```
{
  "brightness": 33,
  "rgb_color": [
    255,
    27,
    44
  ]
}
```

Der findes heldigvis online værktøjer, der kan hjælpe med at konvertere en kode fra YAML til JSON eller omvendt, så det behøver ikke være så svært, som der ser ud. Brug f.eks. [Code Beautify](#), der også farvekoder YAML og JSON koden.

Eksempler på automatiseringer

Dette afsnit indeholder eksempler på flows, som jeg anvender til forskellige formål, f.eks. styring af lys, dørklokke, sende notifikation ved lavt batteriniveau eller hente data fra en webside.

Eksemplerne er udvalgt for at demonstrere brugen af forskellige noder til at udføre funktionalitet, som de mange vil ønske at automatisere.

Der er i slutningen af hvert eksempel indsat link til Pastebin, hvor flowet kan hentes, så du selv kan teste og bygge videre på eksemplerne.

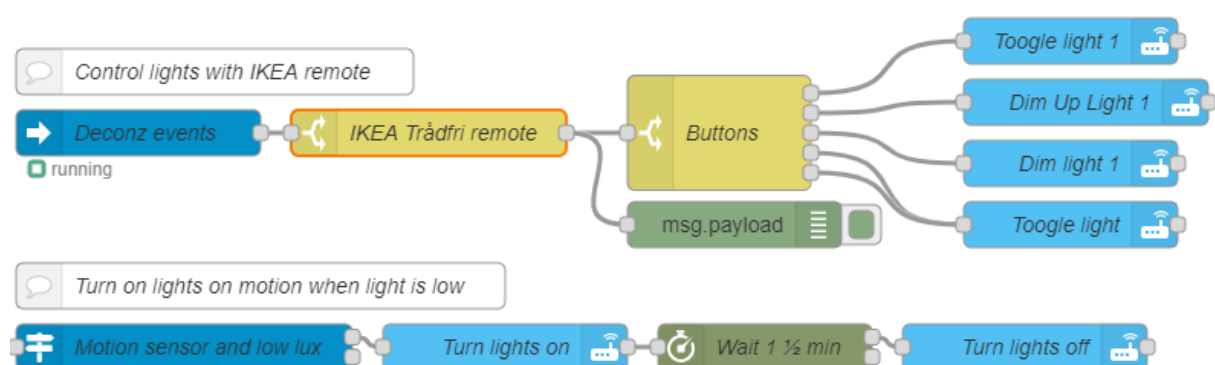
Du kan hente og importere et flow ved at følge de nedenstående trin:

- 1) Klik på linket for at åbne Pastebin
- 2) Klik på **raw** knappen
- 3) Tryk på **Ctrl+A** og **Ctrl+C** for at vælge alt indhold og kopiere
- 4) Åbn Node-RED
- 5) Vælg menupunktet **Import** i menulinjen
- 6) Tryk **Ctrl+V** for at indsætte det kopierede flow
- 7) Tryk på **Import** knappen

Du skal derefter bare tilrette parametre med navne på dine HA enheder for at kunne bruge flowet i din HA installation.

Eksempel #1: Lysstyring

Dette flow er et eksempel på, hvordan flere forskellige automatiseringer til kan samlet i et samlet flow. Flowet indeholder to forskellige dele automatiseringer til lysstyring. Den første automatisering benyttes til at reagere på tryk på en IKEA Trådfri fjernbetjening, mens den anden automatisering reagerer på bevægelse fra en bevægelsessensor med indbygget lysmåler (lux):



Den første automatisering består af følgende noder:

- 1) En *events: all* node reagerer på hændelser fra DeCONZ (kan ændres til Hue, ZHA eller hvad du ellers bruger som gateway til dine smart enheder)
- 2) To *switch* noder tjekker hvilken enhed og hvilken knap, der blev trykket på
- 3) En *debug* node viser hvilke data, der modtages, når der trykkes på en knap på fjernbetjeningen
- 4) Fire *call service* noder benyttes til at tænde og slukke samt skrue op eller ned for lysstyrken på forskellige lys afhængigt af hvilken knap, der blev trykket på

Den anden automatisering består af følgende noder:

- 1) En *Trigger: state* node reagerer på registrering af bevægelse fra en bevægelsessensor. Der er indsat en betingelse, så automatiseringen kun kører ved lavt lysniveau.
- 2) En *call service* node tænder lyset
- 3) En *stoptimer* node pauser automatiseringen i 1 ½ minut. Hvis der registreres ny bevægelse inden tiden er gået, starter der en ny nedtælling.
- 4) En *call service* node slukker lyset igen

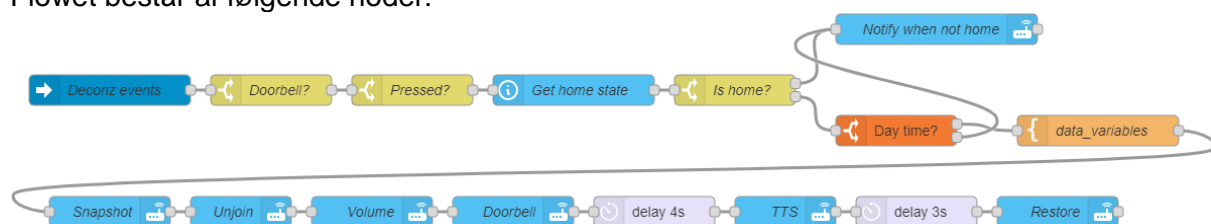
Hent flow på Pastebin: <https://pastebin.com/vcHNuJwD>

Eksempel #2: Ringklokke med lyd over Sonos højttaler

Dette flow benyttes til automatisering af ringklokke, der:

- spiller ringklokke og besked på Sonos, hvis der er nogen hjemme mellem kl. 7.00 – 19.00
- og ellers sendes notifikation til mobil

Flowet består af følgende noder:



- 1) En *events: all* node reagerer på hændelser fra DeCONZ (kan ændres til Hue, ZHA eller hvad du ellers bruger som gateway til dine smart enheder)
- 2) To *switch* noder tjekker hvilken enhed og hvilken knap, der blev trykket på
- 3) En *current state* node henter status for en gruppe af person
- 4) En *switch* node tjekker om der er nogen hjemme eller ej
- 5) En *call service* node sender besked ved at benytte HA's notify service

- 6) En *time range* node tjekker om den nuværende tid er mellem kl. 07.00 – 19.00
- 7) En *template* node benyttes til at angive variabler, som skal bruges af Sonos noder til afspilning af lyd, herunder navn på Sonos højttaler og lydstyrke
- 8) En række *call service* noder (afbrudt af to *delay* noder) benyttes til at spille lyd over en Sonos højttaler. Der spilles en dørklokkelyd samt bruges TTS til at sige, at der er gæster.

Hent flow på Pastebin: <https://pastebin.com/x4ehBLL9>

Eksempel #3: Notifikation ved lavt batteriniveau

Dette flow benyttes til at sende en notifikation hver aften, hvis en eller flere batteridrevne enheder i HA har et lavt batteriniveau. Flowet består af følgende noder:

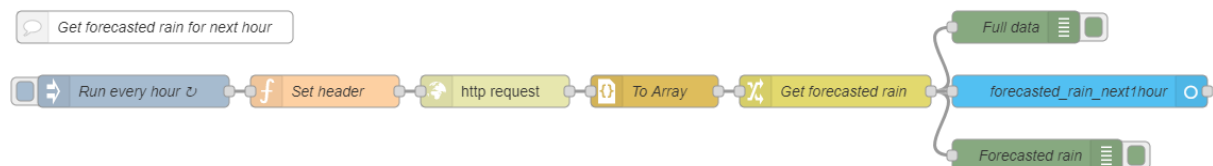


- 1) En *inject* node starter flowet alle dage kl. 20.00
- 2) En *get entities* node henter alle enheder, hvor "device_class" er "battery" og batteriniveaet er mindre end 30%.
- 3) En *template* node laver tekst med navn på enhed og batteriprocent
- 4) En *string* node fjerner teksten "battery level" eller "battery state" fra navnet
- 5) En *join* node samler alle enheder med lavt batteri-niveau i én samlet besked
- 6) En *call service* node sender beskeden ved at benytte HA's notify service

Hent flow på Pastebin: <https://pastebin.com/gmGwAqzG>

Eksempel #4: Hent data fra webside

Dette flow henter hver time det forventede nedbør for den næste time i København fra Met.no (YR) og gemmer det i en HA sensor enhed. Flowet består af følgende noder:



- 1) En *inject* node starter flowet hver time
- 2) En *function* node definerer header og parametre (længde- og breddegrader for lokation for vejruddsigten) til brug for webkald til Met.no's API service
- 3) En *http request* node foretager webkald til Met.no's API service
- 4) En *json* node konverterer de modtagne JSON data til et array objekt
- 5) En *change* node henter den ønskede værdi fra objektet

- 6) En *entity* node gemmer værdien (det forventede nedbør i mm i den næste time) i en ny HA sensor entitet, så den kan vises i HA dashboardet eller bruges i en HA automatisering
- 7) To *debug* noder viser værdien af det fulde objekt samt en udtrukne værdi

Hent flow på Pastebin: <https://pastebin.com/xdKb15c6>

Tips og tricks

Dette afsnit indeholder generelle tips og råd til brugen af Node-RED.

Organiser dine flows


Sørg for at samle automatiseringer med samme formål i hvert deres flow, f.eks. et flow til lysstyring, et andet til styring af robotstøvsugeren og et helt tredje til at sende notifikationer.

Indsæt kommentarer

Indsæt en beskrivelse af hvert flow ved at dobbeltklikke på flowets navn. Beskriv formålet og funktionaliteten af flowet samt evt. forudsætninger for at det virker. Du kan også indsætte evt. links til kilder, som du har hentet inspiration fra, så du altid kan finde tilbage til udgangspunktet. Endeligt kan du lave en simpel versionsstyring, så du kan huske, hvornår du har ændret i et flow.

Du kan indsætte kommentarer i de enkelte flow for hurtigt at kunne læse, hvad de enkelte dele af flowet gør.

Brug den indbyggede hjælpefunktion

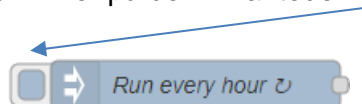
Brug den indbyggede hjælp ved at vælge en node og klikke på  ikonet i **sidebar** i højre side af skærmen. Så kan du få vist vejledning til brug af noden, herunder f.eks. information om input, output og parametre til konfiguration af den valgte node.

Deploy ændringer løbende

Husk at trykke på **Deploy** knappen undervejs, når du arbejder med dit flow. NR gemmer kun, når du laver et deploy og du risikerer derfor at miste ændringer, hvis din browser lukker ned.

Brug inject node til at teste

Når du tester et flow, kan du med fordel indsætte en eller flere *inject* noder, som kan bruges til at starte dit flow fremfor f.eks. at vente på at en given state ændres. *Inject* noden kan bruges til at sende det forventede output (payload) videre til næste node i flowet f.eks. hvert minut mens du tester eller, når du klikker på den firkantede knap på noden:



Disable noder under test

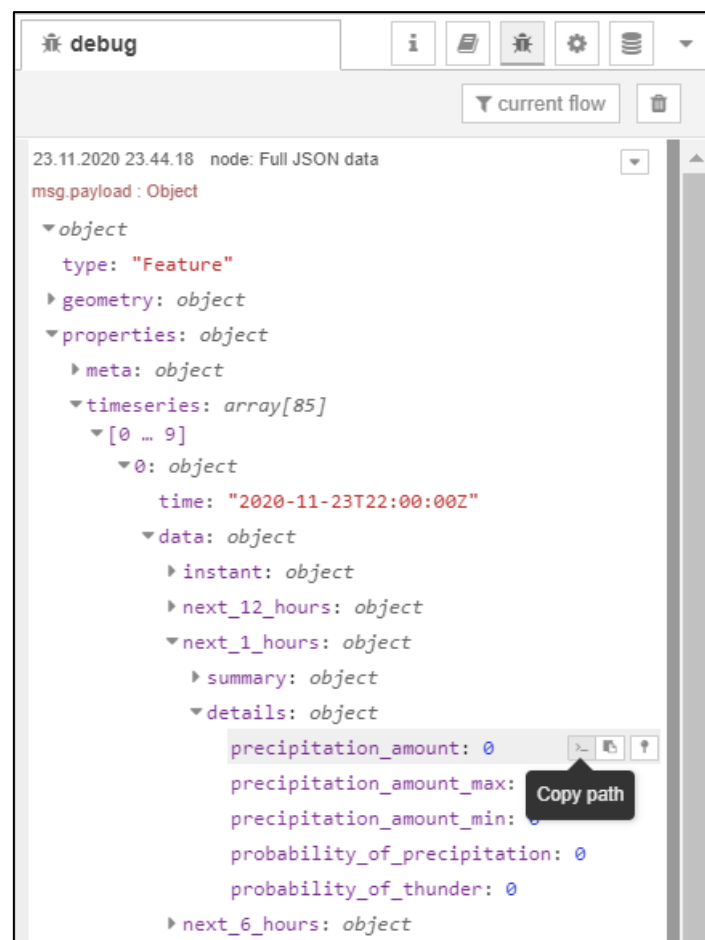
Du kan deaktivere en eller flere noder i dit flow, mens du tester det. På den måde kan du teste én del af et flow ad gangen. Det kan f.eks. være en fordel at vente med at sende beskeder til mobiler før man er helt sikker på at flowet fungerer som forventet. Derfor kan du

enten vente med at indsætte eller linke en call service node til resten af flowet eller blot deaktivere den ved at vælge disable.

Brug debug noder til udvikling og fejlsøgning

Du kan indsætte *debug* noder på udvalgte steder i dit flow, mens du tester. Så kan du se det output, som en node leverer til den næste node i dit flow i *debug vinduet* i **sidebar** i højre side af skærmen.

Du finder *debug* vinduet ved at klikke på  ikonet:



Figur 4: Eksempel på indhold i debug vinduet

TIP: Du kan bruge funktionen **Copy path** til at kopiere stien på data du har markeret (i eksemplet `payload.properties.timeseries[0].data.next_1_hours.details.precipitation_amount`) i stedet for at skrive det selv.


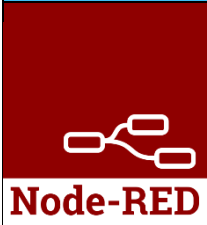
Debug vinduet vil også vise eventuelle fejl i afviklingen af dit flow med beskrivelse af fejlen, så du kan identificere den node, som fejler og den mulige årsag til fejlen.

Opsummering

Vi har nu set på, hvordan Node-RED kan anvendes til automatiseringer til styring af HA enheder og gennemgået nogle konkrete eksempler på flows.

Guiden har dog kun beskæftiget sig med et lille udsnit af de muligheder, som Node-RED tilbyder for at bygge automatiseringer til styring af HA enheder.

Er du stadig i tvivl om NR er det rigtige valg til dine automatiseringer, så kan denne oversigt over fordele og ulemper ved automatiseringer i både HA og NR måske hjælpe til en beslutning:

Platform	Fordele	Ulemper
	<ul style="list-style-type: none">• Kræver ikke ekstra installation• De fleste bruger YAML, så der findes mange eksempler til inspiration	<ul style="list-style-type: none">• Kan være svært at lave fejlsøgning• Kan være svært at overskue automatiseringer, særligt de mere komplekse
	<ul style="list-style-type: none">• Visuel tilgang til programmering og overblik over automatiseringer• Gode muligheder for fejlsøgning• Let at deploye ændringer• Avancerede muligheder for web og database integration	<ul style="list-style-type: none">• Kræver installation af addon og bruger ekstra systemressourcer• Færre brugere og eksempler til inspiration• Kan være svært at lære og mestre

Der er fordele og ulemper ved automatiseringer i både YAML og NR, og kun du kan bestemme, hvad der er den rigtige løsning for dig.

Selvom det kan virke uoverskueligt at sætte sig ind i to forskellige måder at lave automatiseringer på, vil jeg dog anbefale dig, at prøve at lave den samme automatisering i både YAML og NR. På den måde kan du finde ud af, hvad der virker mest logisk og lettest at gå til for dig.

NR automatiseringen fra eksempel #3 kan f.eks. også laves med følgende YAML kode i HA:

```
- id: low_battery_warning
  alias: Low battery notification
  description: Low battery notification example
  trigger:
    - platform: time
      at: "20:00:00"
  condition:
    - condition: template
      value_template: >
        {{ states | selectattr('attributes.device_class', 'eq', 'battery') |
          selectattr('state', 'lt', '30') | map(attribute='attributes.friendly_name') |
          list | count > 0 }}
  action:
    - service: notify.mobile_app_myphone
      data_template:
        message: >-
          En eller flere enheder har lavt batteri:
          {{ states | selectattr('attributes.device_class', 'eq', 'battery') |
            selectattr('state', 'lt', '30') | map(attribute='attributes.friendly_name') |
            list | join(', ') | replace(' battery level', '') | replace(' Battery Level', '')
            | replace(', ', '\n') }}
          Du bør oplade eller skifte batteriet.
  mode: single
```









Det er måske mindre visuelt overskueligt, hvad automatiseringen rent faktisk gør, da der er brugt Jinja2 templates. Omvendt er det meget nemt at kopiere ind og bruge i HA.

Uanset hvad du måtte foretrække, så husk, at du ikke behøver at anvende NR i stedet for dine eksisterende HA's automatiseringer. Du kan sagtens bruge NR som et supplement til dine automatiseringer i HA.

Det kunne f.eks. være, at du foretrækker at holde de mest kritiske eller de mere simple automatiseringer i YAML og så lave de mere komplekse automatiseringer med mange forskellige mulige udfaldsrum (*switch* noder) eller betingelser i NR.

Yderligere inspiration

Hvis du har fået mod på, at starte med at lære NR, er her nogle links med mere inspiration til brug af NR sammen med HA i form af videoer (📺), flow-eksempler ({...}) og guides (📄).

-  [DIY Techie's flows og subflows, til f.eks. brug af Hue/IKEA remotes via DeCONZ](#)
-  [Esben Bücking-Rasmussen's video om integration til MinVolkswagen via NR](#)
-  [JuanMTech - How to get started with Node-RED and Home Assistant](#)
-  [The Hook Up - Node-RED + Home Assistant How-To](#)
-  [The Hook Up - Triggers, Events, Motion Lighting and Actionable Notifications i NR](#)
-  [digiblurDIY - Node-RED Basics with Home Assistant - \[Part 1\]\(#\), \[Part 2\]\(#\), \[Part 3\]\(#\), \[Part 4\]\(#\)](#)
-  [Automate The Things - Node-RED for Automations in Home Assistant](#)
-  [DIY Futurism - Basic Node-Red Flows for Automating Lighting with Home Assistant](#)