# Addons - Virtuelle Komponenten

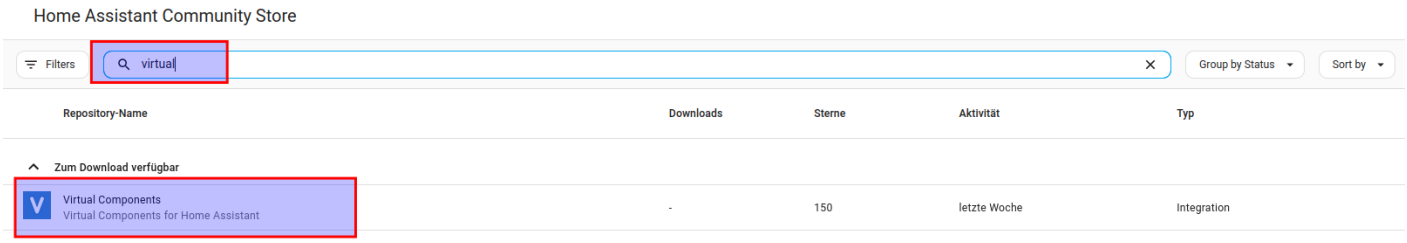
# Installation

Beschreibung:

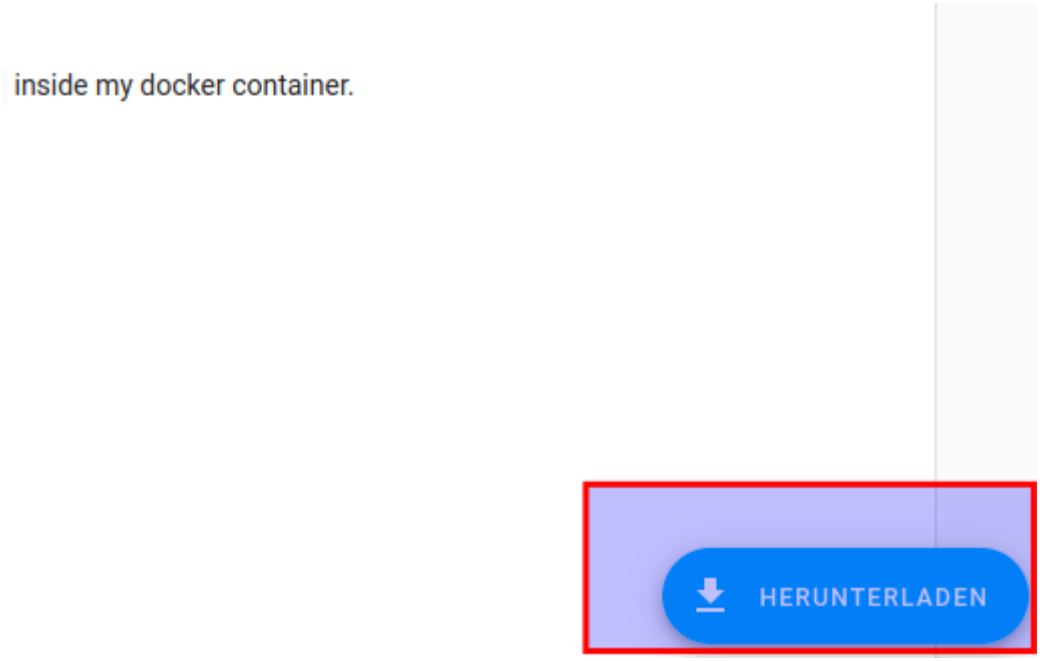
Möchte man was ausprobieren oder Automatisierungen schreiben, für Geräte die noch gar nicht angeschafft wurden, kann man sich Virtuelle Geräte bauen. In unserem Beispiel ein Rauchmelder. Ich besitze noch keinen, möchte aber schon mal die Automatisierungen bauen. So das ich nachher nur noch die Entitäten gegen den echten Rauchmelder austauschen muss.

Installation:

Im HACS Store gibt es eine Integration Virtual Component.
Dazu gehen wir auf HACS und geben Virtual als Suchbegriff ein und klicken dann auf Virtual Component.

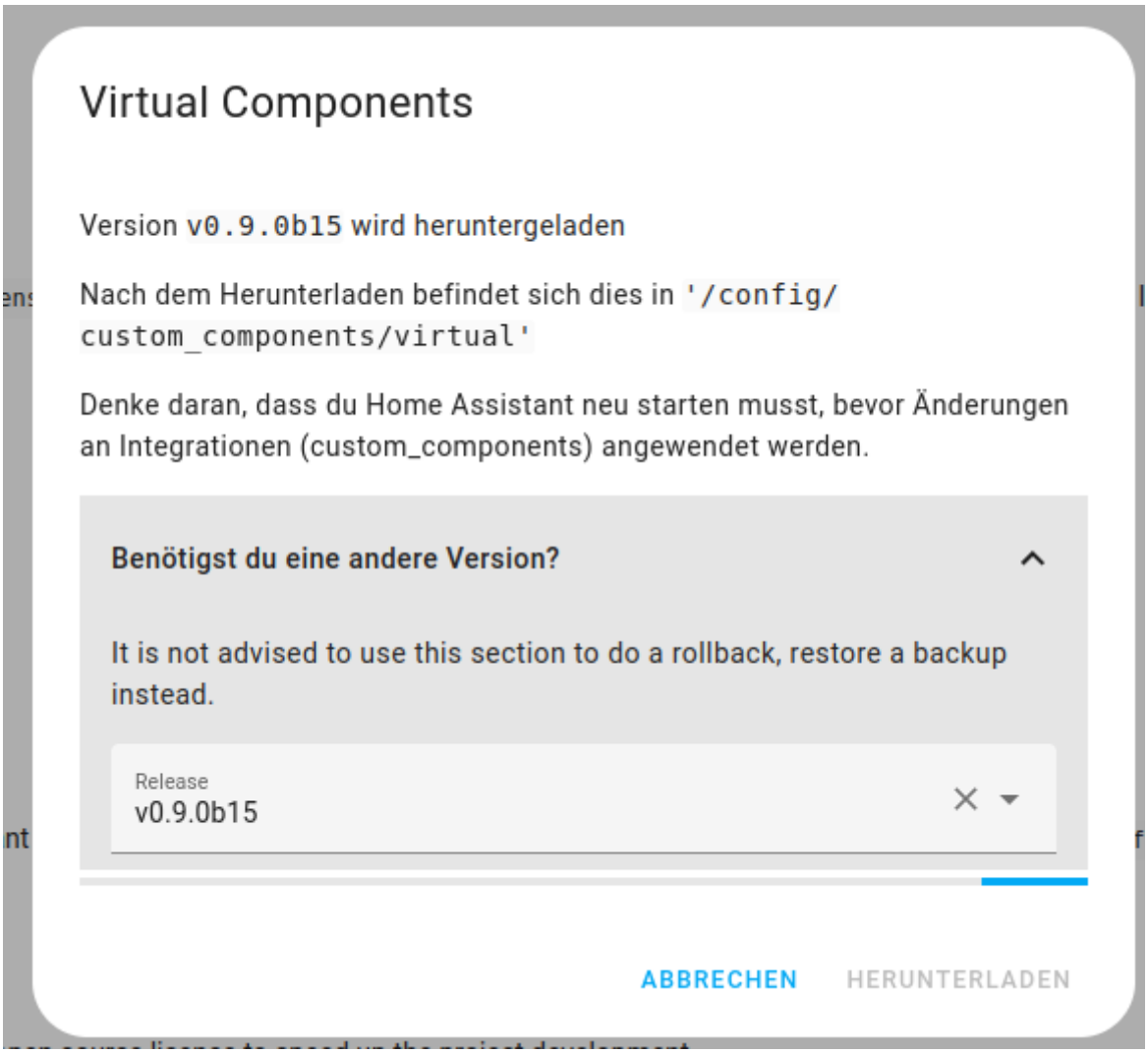



Jetzt wieder unten rechts auf Herunterladen klicken.



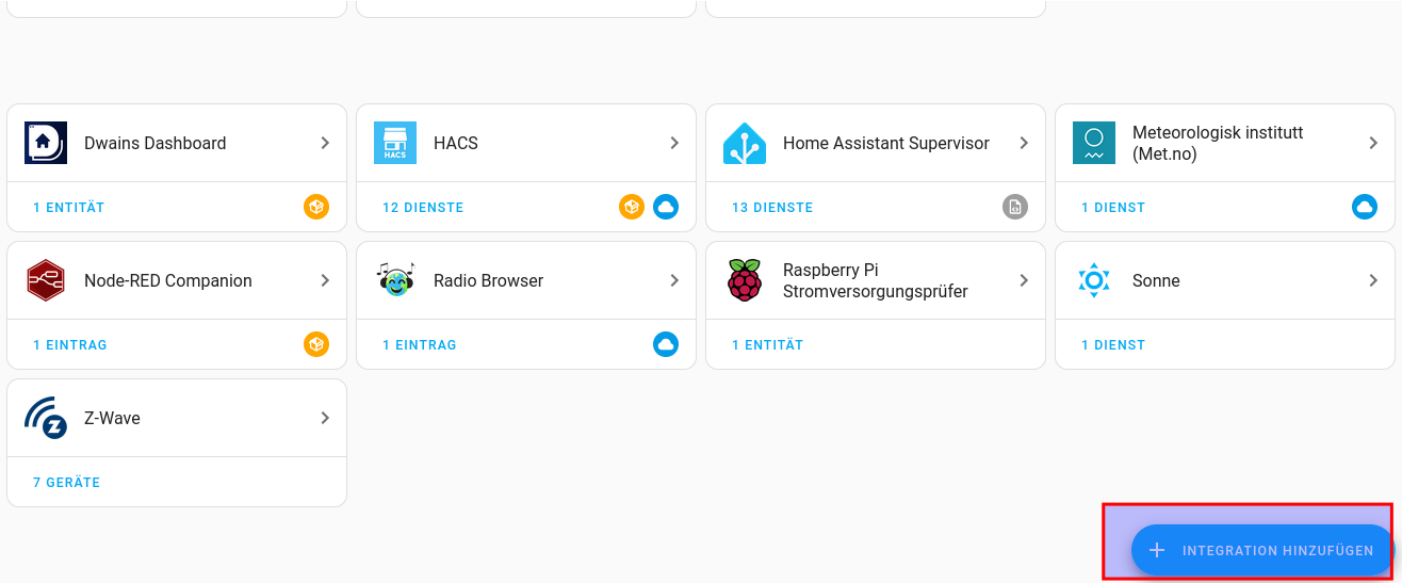
!!! Wir wählen hier die Letzte Preversion 0.9 aus, weil es in 9 einen Syntax wechsel gibt und wir auf aktuellem Stand sein wollen.

Nun einfach nochmal auf Herunterladen klicken

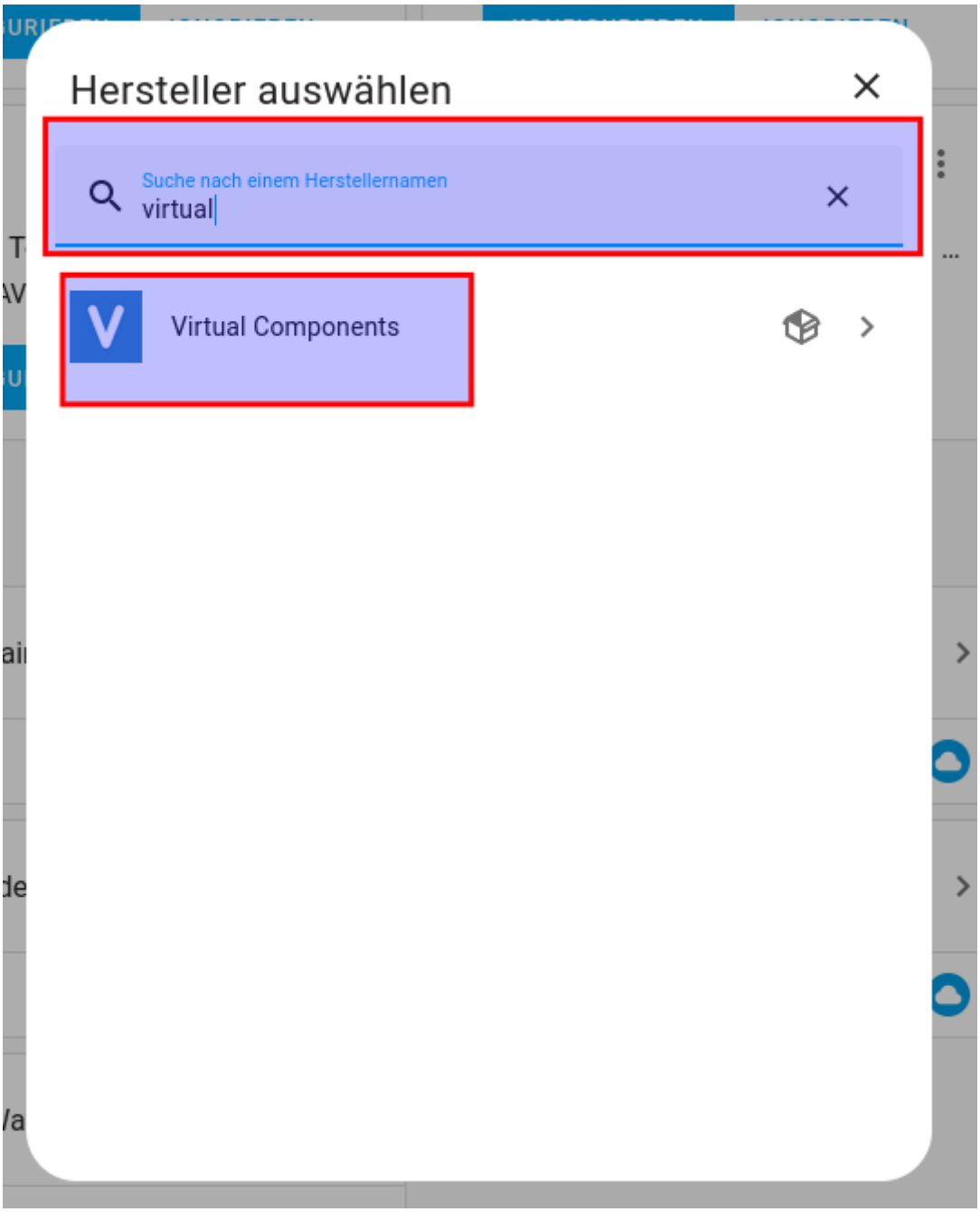



Nun sehen wir Virtual Component in der Liste unter neustart Austehend.'
Jetzt Home Assistant einmal neustarten.

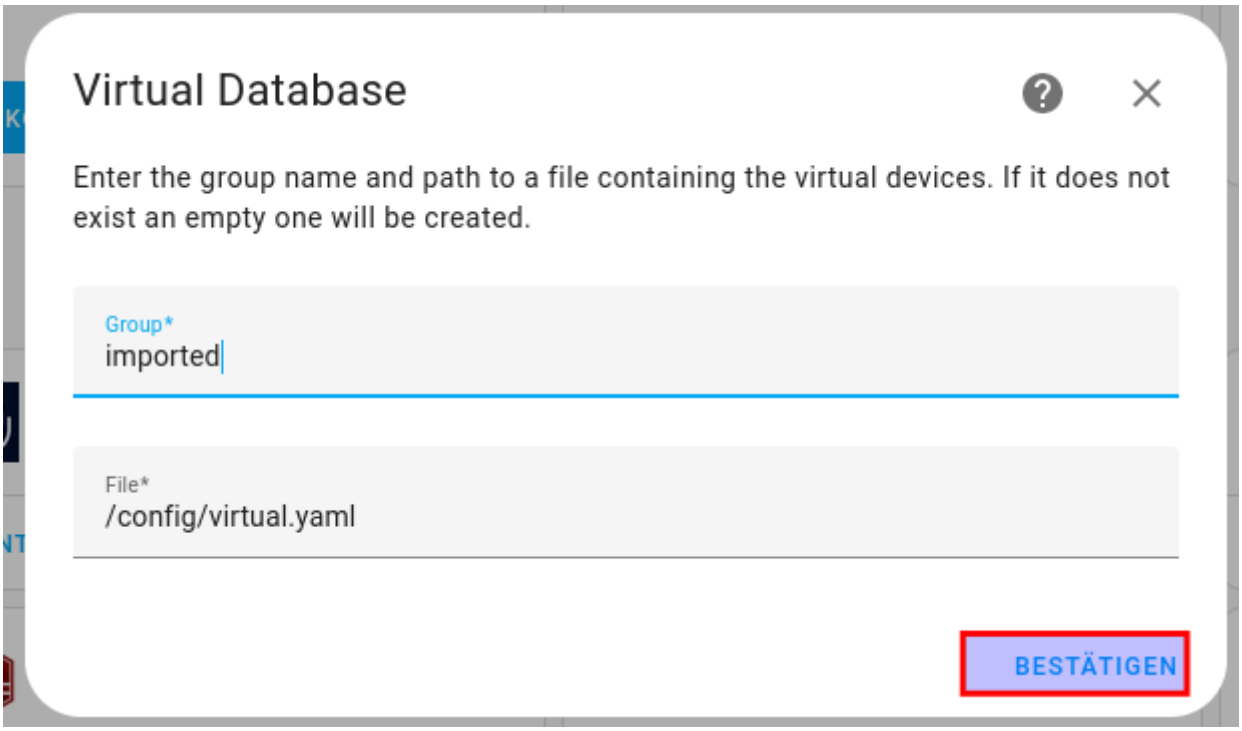
Nun auf Geräte -> Inegrationen hinzufügen klicken




Nun als Suchbegriff Virtual eintippen und dann aus  der Liste Virtual Component auswählen




Nu auf bestätigen klicken, denn das lassen wir so




Dies erstellt gleich eine leere virtual.yml
Fertig

# Geräte anlegen

Beschreibung:

Da wir die Installation nun hinter uns haben, können wir nun unsere ersten Dummy Geräte anlegen.
Dazu müssen wir `virtual.yaml` anlegen, dort wo auch die configration.yml liegt wechseln und am ende unsere Devices anlegen.
Hier anhand unseres Rauchmelders den es ja gar nicht gibt. ;-)

> **Hinweis:** Die Datei hat **.yaml** und nicht **.yml** am Ende

Code:

```
devices:
  Virtual Smoke Detector 1: #Gerätename
    - platform: binary_sensor #Welcher Typ. Typenliste weiter unten. Hier ein bool als an/aus (dieser Wert ist nur read only)
      name: Smoke Detected 1  #der Name der Entität
      initial_value: 'off' #Init state. Also beim starten des gerätes bzw Home Assistant
      class: smoke #Die Klasse, danach wird auch das Icon vergeben. Klassenliste weiter unten

    - platform: binary_sensor #Welcher Typ. Typenliste weiter unten. Hier ein bool als an/aus (dieser Wert ist nur read only)
      name: Tamper Detected 1 #der Name der Entität
      initial_value: 'off' #Init state. Also beim starten des gerätes bzw Home Assistant
      class: tamper #Die Klasse, danach wird auch das Icon vergeben. Klassenliste weiter unten

    - platform: sensor  #Ein sensor, kann also einen Wert haben
      name: Battery Level 1 #der Name der Entität
      initial_value: 100 #Init state. Also beim starten des gerätes bzw Home Assistant
      unit_of_measurement: "%" #Welche Einheit. hier Prozent. Eine Liste der Maßeinheiten dann weiter unten
      class: battery  #Die Klasse, danach wird auch das Icon vergeben.



    - platform: switch
      name: Siren 1 #der Name der Entität
```

```
    initial_value: 'off' #Init state. Also beim starten des gerätes bzw Home Assistant
    #Bei einem Switch gibt es keine Klasse



#Eine Lampe hat besondere Eigenschaften:
#Denn diese hat nur eine Platform einen initwert und der rest ist lampen plattform spezifisch
#Besxhreibung siehe Tabelle weiter unten

  living room main light:
    - platform: light
      initial_value: 'on'
      support_brightness: true
      initial_brightness: 100
      support_color: true
      initial_color: [0,255]
      support_color_temp: true
      initial_color_temp: 255
      support_white_value: true
      initial_white_value: 240

#eine Rollade
  Rollade Wohnzimmer:
  - platform: cover #typ rollade
    initial_value: 'closed' #initialwert open oder closed
    open_close_duration: 10 #zeit wie lange von ganz offen bis ganz geschlossen und umgekehrt. die Zeit gillt für beide Richtungen
    open_close_tick: 1 #Das update interval wenn er beim öffnen oder schließen ist
```

Danach Home Assistant neustarten und schon haben wir ein neues Gerät in diesem fall unseren Rauchmelder. Weitere Geräte einfach drunter anfügen.

Nach jedem ändern muss Home Assistant neugestartet werden. Nur Konfig neuladen reicht nicht.

Platformen, Einheiten und Klassen:

**Plattformtypen**

| **Plattform** | **Beschreibung** |
|---|---|
| binary_sensor | Ein binärer Sensor, der Zustände wie an/aus oder offen/geschlossen überwacht. |
| sensor | Ein allgemeiner Sensor für Messwerte wie Temperatur, Luftfeuchtigkeit, etc. |

| | |
|---|---|
| switch | Ein Schalter zum Ein- und Ausschalten von Geräten. |
| light | Ein Lichtsteuerungselement zum Ein-/Ausschalten, Dimmen und Ändern von Farben. |
| lock | Ein Entitätstyp für das Verriegeln und Entriegeln von Schlössern. |
| camera | Ein Kameramodul, das Video-Feeds bereitstellt. |
| media_player | Ein Entitätstyp zur Steuerung von Mediengeräten wie Lautsprechern und Fernsehern. |
| climate | Ein Steuerungsmodul für Heizungen, Klimaanlagen und Thermostate. |
| cover | Ein Modul zur Steuerung von Abdeckungen wie Rollläden und Garagentoren. |
| fan | Ein Entitätstyp zur Steuerung von Ventilatoren. |

## Einheiten

| **Einheit** | **Beschreibung** |
|---|---|
| °C | Grad Celsius - Temperaturmaß. |
| °F | Grad Fahrenheit - Temperaturmaß. |
| % | Prozent - z.B. für Luftfeuchtigkeit, Akkustand. |
| m | Meter - Länge/Distanzmaß. |
| cm | Zentimeter - Länge/Distanzmaß. |
| mm | Millimeter - Länge/Distanzmaß. |
| km | Kilometer - Länge/Distanzmaß. |
| m/s | Meter pro Sekunde - Geschwindigkeit. |
| km/h | Kilometer pro Stunde - Geschwindigkeit. |
| mph | Meilen pro Stunde - Geschwindigkeit. |
| W | Watt - Leistung. |
| kW | Kilowatt - Leistung. |
| kWh | Kilowattstunde - Energieverbrauch. |
| Wh | Wattstunde - Energieverbrauch. |

| | |
|---|---|
| V | Volt - Spannung. |
| A | Ampere - Stromstärke. |
| Ah | Amperestunde - Kapazität von Batterien. |
| mA | Milliampere - Stromstärke. |
| dB | Dezibel - Lautstärke. |
| lx | Lux - Beleuchtungsstärke. |
| lm | Lumen - Lichtstrom. |
| Pa | Pascal - Druck. |
| bar | Bar - Druck. |
| psi | Pfund pro Quadratzoll - Druck. |
| ppm | Teile pro Million - Konzentration (z.B. CO2). |
| g | Gramm - Gewicht. |
| kg | Kilogramm - Gewicht. |
| mg | Milligramm - Gewicht. |
| l | Liter - Volumen. |
| ml | Milliliter - Volumen. |

## Klassen

| **Klasse** | **Beschreibung** |
|---|---|
| motion | Bewegungssensor (erkennt Bewegung). |
| door | Türsensor (erkennt, ob eine Tür geöffnet/geschlossen ist). |
| window | Fenstersensor (erkennt, ob ein Fenster geöffnet/geschlossen ist). |
| smoke | Rauchsensor (erkennt Rauch). |
| gas | Gassensor (erkennt das Vorhandensein von Gasen). |
| battery | Batteriesensor (überwacht den Batteriestatus). |
| humidity | Feuchtigkeitssensor (misst Luftfeuchtigkeit). |
| temperature | Temperatursensor (misst Temperatur). |

| | |
|---|---|
| illuminance | Helligkeitssensor (misst Lichtstärke). |
| moisture | Feuchtigkeitssensor (misst Bodenfeuchtigkeit). |
| opening | Öffnungssensor (allgemein für Türen/Fenster). |
| power | Leistungssensor (misst Stromverbrauch). |
| presence | Präsenzsensor (erkennt Anwesenheit). |
| pressure | Drucksensor (misst Druck). |
| safety | Sicherheitssensor (allgemein für Sicherheitsüberwachung). |
| sound | Geräuschesensor (erkennt Geräusche). |
| vibration | Vibrationssensor (erkennt Vibrationen oder Erschütterungen). |

## Beschreibung der Konfigurationszeilen der Plattform Licht

| **Konfigurationszeile** | **Beschreibung** |
|---|---|
| living room main light: | Dies ist der Name des virtuellen Geräts. In diesem Fall bezieht es sich auf das Hauptlicht im Wohnzimmer. Der Name dient als Referenz, um dieses Licht im System anzusprechen. |
| - platform: light | Gibt an, dass dieses Gerät eine light-Plattform ist. Es handelt sich also um ein Licht, das über Home Assistant gesteuert werden kann. |
| initial_value: 'on' | Definiert den anfänglichen Zustand des Lichts. Wenn Home Assistant startet, wird dieses Licht automatisch eingeschaltet sein (on). |
| support_brightness: true | Gibt an, dass das Licht die Helligkeitssteuerung unterstützt. Mit dieser Einstellung kann die Helligkeit des Lichts angepasst werden (z.B. dimmen oder aufhellen). |
| initial_brightness: 100 | Definiert die anfängliche Helligkeit des Lichts, wenn es eingeschaltet wird. Der Wert kann zwischen 0 (aus) und 255 (maximale Helligkeit) liegen. In diesem Fall ist der Wert auf 100 eingestellt, was eine mittlere Helligkeit bedeutet. |
| support_color: true | Gibt an, dass das Licht Farbänderungen unterstützt. Das bedeutet, dass das Licht in verschiedenen Farben leuchten kann. |
| initial_color: [0,255] | Definiert die anfängliche Farbe des Lichts in RGB-Werten. Hier bedeutet [0, 255] wahrscheinlich eine grüne Farbe. Normalerweise werden RGB-Werte als drei Zahlen angegeben (z.B. [255, 0, 0] für rot), also könnte dies eine abgekürzte Notation sein, je nach Implementierung. |

| | |
|---|---|
| support_color_temp: true | Gibt an, dass das Licht die Farbtemperatursteuerung unterstützt. Damit kann die Farbtemperatur des Lichts angepasst werden, zum Beispiel von warmem Gelb zu kühlem Blau. |
| initial_color_temp: 255 | Definiert die anfängliche Farbtemperatur des Lichts. Der Wert kann typischerweise in einem Bereich von 153 (kühleres, bläuliches Licht) bis 500 (wärmeres, gelbliches Licht) liegen. In diesem Fall ist der Wert auf 255 eingestellt, was einer neutralen Farbtemperatur entspricht. |
| support_white_value: true | Gibt an, dass das Licht einen Weißwert unterstützt. Dies wird verwendet, wenn das Licht neben der Farbsteuerung auch die Intensität von weißem Licht steuern kann. |
| initial_white_value: 240 | Definiert den anfänglichen Weißwert des Lichts. Der Wert kann zwischen 0 (kein weißes Licht) und 255 (maximale Intensität des weißen Lichts) liegen. Hier ist der Wert auf 240 gesetzt, was fast maximale Intensität bedeutet. |

**Hinweis:** *white_value ist veraltet und wird in zukünftigen Versionen **entfernt.**

# Geräte Werte Temporär ändern ohne neustart

## Beschreibung:

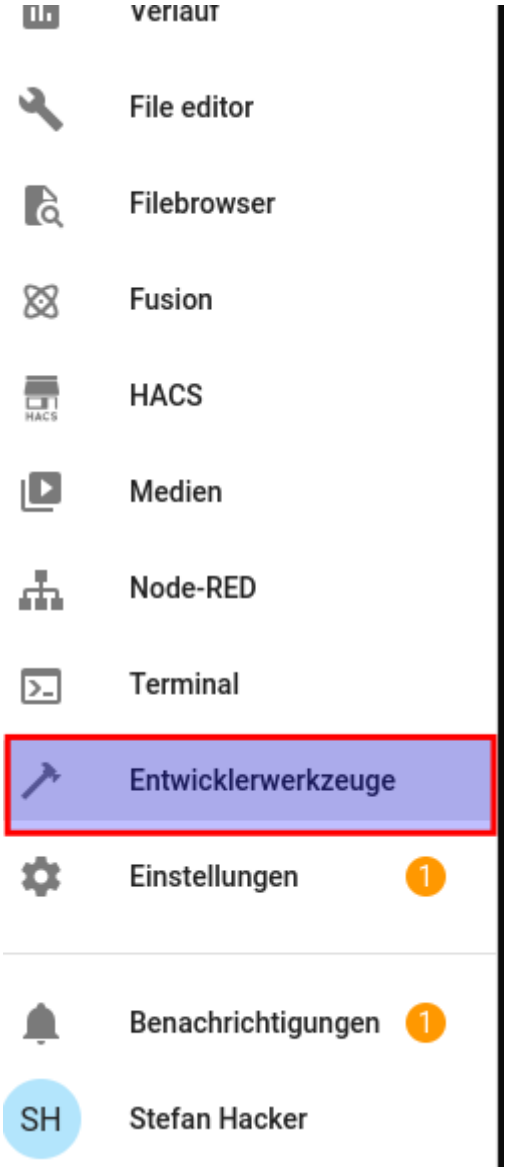
Möchte man sonst Werte ändern, geht das ja nur in der Konfiguration Datei und einem neustart.
Möchte man nur den Batteriestand ändern um eine Automation zu testen, nervt Konfigurationsdatei und ein neustart extrem.
Abhilfe die Entwicklerwerkzeuge.

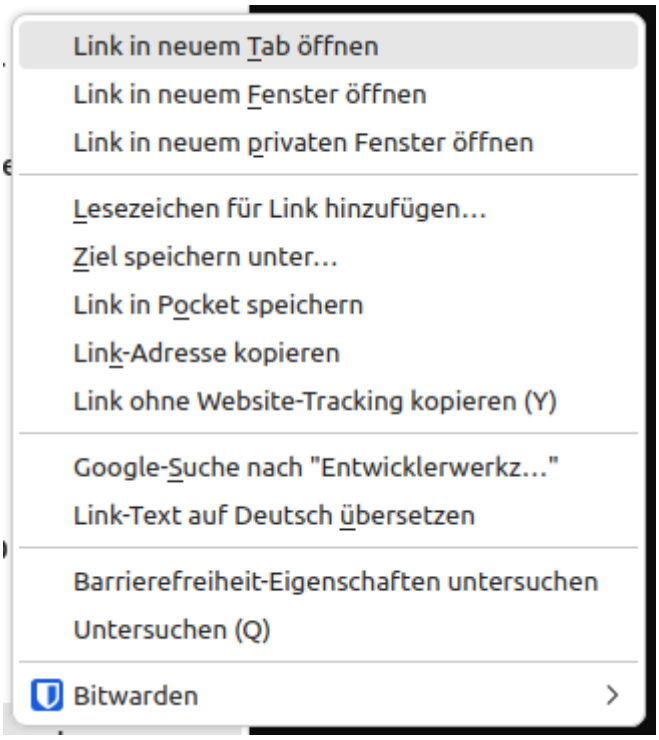
## Anwendung:

Es empfiehlt sich die Entwicklerwerkzeuge in einen Neuen Tab zu öffnen, da das überschreiben nur solange gültig ist bis zum nächsten Updatezyklus der Geräte.
Denn die Geräte senden ja immer noch den Initwert mit dem Sie gestartet wurden.

Deshalb macht es sinn einen zweiten Tab zu nehmen, weil dann kann man schnell einfach wieder auf den Button setzten klicken.

Wir finden die Entwicklerwerkzeuge links unten im Menü
Dort mit rechten Maustaste drauf und im neuen Fenster Öffnen anklicken

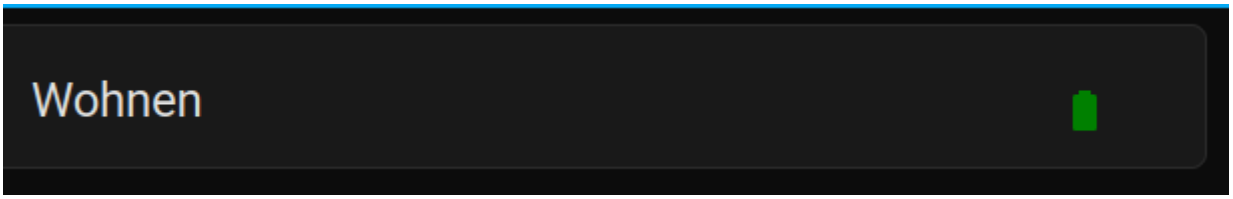

Verlauf
File editor
Filebrowser
Fusion
HACS
Medien
Node-RED
Terminal
Entwicklerwerkzeuge
Einstellungen
1
Benachrichtigungen
1
SH
Stefan Hacker


Link in neuem Tab öffnen
Link in neuem Fenster öffnen
Link in neuem privaten Fenster öffnen
Lesezeichen für Link hinzufügen…
Ziel speichern unter…
Link in Pocket speichern
Link-Adresse kopieren
Link ohne Website-Tracking kopieren (Y)
Google-Suche nach "Entwicklerwerkz…"
Link-Text auf Deutsch übersetzen
Barrierefreiheit-Eigenschaften untersuchen
Untersuchen (Q)
Bitwarden

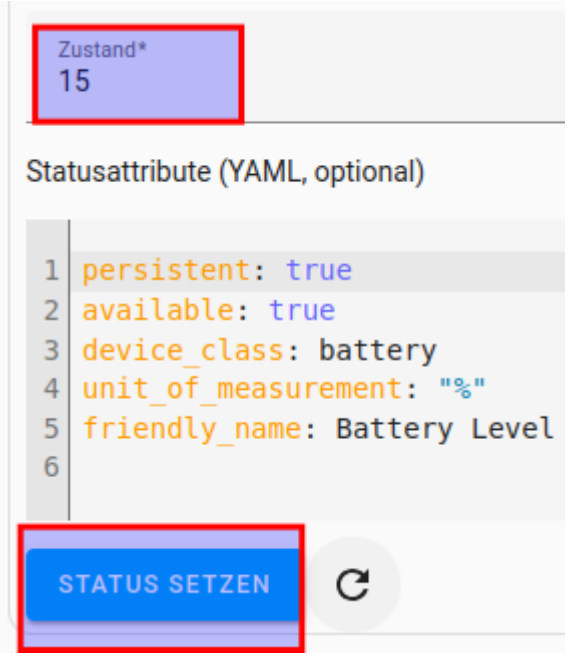
Nun in den Tab wechseln und in den Entwicklerwerkzeugen auf Zustände klicken und dann auf Status setzten klicken.


Nun kann aus der Liste die Entität ausgewählt werden. In der liste kann auch wieder ein Wort eingeben werden das dann als Teilsuche fungiert. Batt für Battery.
Wir wählen eine Batterie. Als Wert wird immer der aktuelle Wert angegeben, hier 100


Schauen wir uns unser Dashboard im anderen Tab an, wo das Batterie Icon ist. Das ist zur Zeit noch Grün.

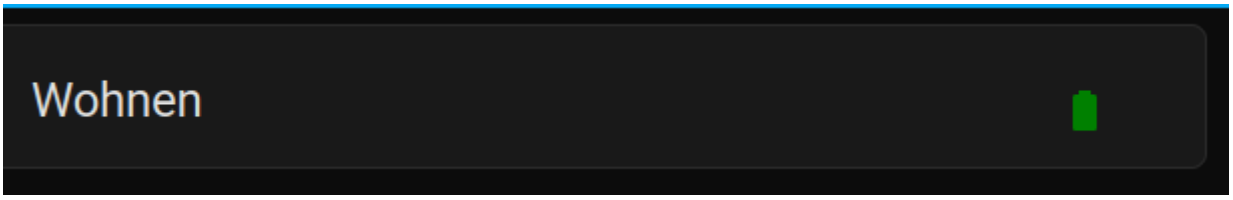



Nun gehen wir in den Tab mit den Entwicklerwerkzeugen zurück und ändern den Wert 100 in 15 ab und klicken auf status setzten




Nun ändert sich für ca 5 Sekunden der Status, dann wird er vom Gerätezyklus Update zurück gesetzt.
Nun hat die Batterie einen Füllstand von 15%

Nach den 5-6 Sekunden geht die Entität in seinen Ursprünglichen Status zurück.
Dies funktioniert auch bei echten Geräten, nur das diese dann natürlich den wirklichen Wert den Sie gerade haben dann wieder zurückgeben.




Fertig