

Home Assistant: YAML, Automationen und Skripte

Was erwartet Sie in diesem Beitrag?

Was geschieht bei einem *Clubabend*, beim Forum *Smart Home | IoT*? Dieser Beitrag behandelt den Clubabend vom 28. Mai 2025 als Nachlese. Die Grundlagen wurden bereits in den PCNEWS 184 vorgestellt und beim Clubtreffen noch einmal im Detail geübt. Alles ist Schritt-für-Schritt in den Download-Ordnern zu finden, auch eine weitere Einführung zu Home Assistant-Themen. Ferner sind alle Schritte mit Screenshots unterlegt.

Die Themen waren...

- ... ein Überblick über YAML
- ... das Erstellen von Automationen und die verwendeten Editoren
- ... Einträge in der Datei `configuration.yaml`
- ... der Einsatz von ChatGPT
- ... Skripte und Bausteine

In der Langversion steht anstelle von 🙌 viel mehr Text.

YAML



YAML ist die Beschreibungssprache, in der im Home Assistant (HA) Konfigurationen und Anweisungen formuliert werden (können), aber auch veröffentlicht werden. Wer mit HA arbeitet, sollte die Grundlagen von YAML kennen. Der Vergleich mit Konstrukten in JavaScript oder JSON kann manchen Lesern helfen.

YAML ist also keine Programmiersprache, sondern eine für Menschen gut lesbare Form zur seriellen Darstellung von *Daten*, verwendbar in beliebigen Programmiersprachen.



Die Grundlagen von YAML

Jede YAML-Darstellung besteht ausschließlich nur aus:

- Einzelwerten (in JavaScript: Literale)
- Listen (*list*, in JavaScript, Python: Listen, etwas ungenau: Array)
- Assoziativen Listen (*map*, in JavaScript: Objekte, in C/C++: struct)

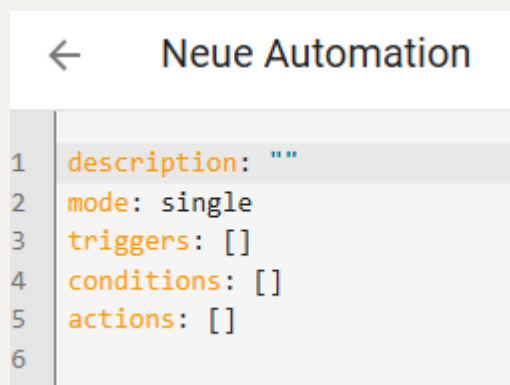
Listen und Maps können selbst wieder alle Datenarten enthalten, also auch weitere Listen oder Maps. 🙌

Automationen

Sie sind *das* zentrale Steuerelement in Home Assistant. 🙌 Automationen können im *visuellen Editor* oder mit *YAML* bearbeitet werden. 🙌

YAML-Editor

Zwischen dem visuellen Editor und dem YAML-Editor kann jederzeit umgeschaltet werden. Dazu wird rechts oben das Drei-Punkt-Menü  aufgerufen. 🙌



```
1 description: ""
2 mode: single
3 triggers: []
4 conditions: []
5 actions: []
6
```

Wie sind diese fünf Zeilen syntaktisch aufgebaut? Sie bilden eine Liste von Maps. Die jeweiligen Keys stehen vor den Doppelpunkten und sind in oranger Schrift hervorgehoben. 🙌 HA arbeitet sehr viel mit Konfigurationsdateien, in denen Geräte, Abläufe, Systemeinstellungen und mehr beschrieben und gespeichert werden. Die Beschreibungssprache ist YAML. *Die Mutter aller Konfigurationen* ist die Datei `configuration.yaml`. Wenn wir *eine einzige* neue Automation erstellen (was in der Praxis kaum vorkommt), könnten wir diese dort unter `automation` eintragen:

```
1 automation:
2   triggers: []
3   conditions: []
4   actions: []
```

Diese Automation (beginnend mit Zeile 1) ist eine Map mit dem Schlüssel `automation`, deren Wert wieder eine Map mit drei Einträgen (Zeilen 2 bis 4) ist. Die Werte der drei Einträge sind selbst leere Listen. 🙌

Automation zum Umschalten einer Lampe in YAML:

```
1 alias: L1 umschalten
2 description: Bsp. 1, Version 1
3 triggers:
4   - domain: mqtt
5     device_id: 2a73abd2b10d069a70ddeb7b4e000a23
6     type: action
7     subtype: single
8     trigger: device
9 conditions: []
10 actions:
11   - type: toggle
12     device_id: a1213e273962d1700c88b6c989213151
13     entity_id: 73a8e2da66a719a3560fac4186c87958
14     domain: light
15 mode: single
```



ChatGPT

Es gibt viele Varianten, wie ein Problem gelöst werden kann. Kann ChatGPT einen Vorschlag machen?

Ich habe eine Lichtsteuerung (siehe YAML) mit Home Assistant erstellt und möchte sie so umstellen, dass beim Drücken des Tasters T1 das Licht in 2 Sekunden die volle Helligkeit erreicht und beim zweiten Tastendruck die Lampe L1 nach 6 Sekunden dunkel wird. Ohne Verwendung einer Hilfsvariablen. 🙌 ChatGPT liefert dazu ein YAML-Programm, das alle gewünschten Funktionen erfüllt. Diese Anfrage an ChatGPT war recht erfolgreich, obwohl noch kleine Änderungen notwendig waren! Allerdings gibt es noch andere Aufgaben, in denen ChatGPT total versagt, zum Beispiel beim Erstellen eines ziemlich einfachen Schaltplans. Aber ChatGPT ist ja noch nicht einmal 3 Jahre alt – das kommt schon noch!

Skripte

Ein Skript kann mit einer Funktion – einem Unterprogramm – in einer höheren Programmiersprache verglichen werden.

Wir haben vorhin eine Automation erstellt, die eine bestimmte Lampe dimmt. Wäre doch nett, diese Idee auf verschiedene Lampen anwenden zu können, ohne jedes Mal eine komplette Automation schreiben zu müssen. Das kann mit einem Skript erreicht werden. 🙌

Bausteine

🙌 Home Assistant bietet viele Bausteine an. 🙌 Einige dieser Bausteine sind schon von den Automationen bekannt:

- `Auf einen Auslöser warten` entspricht dem Sobald-Schritt in Automationen.
- `Bedingung` funktioniert wie bei den Automationen: wird sie nicht erfüllt, endet der Ablauf.

Weitere Bausteine:

`Eine Zeit warten (Verzögerung)` erklärt sich wohl selbst. Zeiten können in Stunden, Minuten, Sekunden und Millisekunden angegeben werden.

Ein interessantes Thema! Aber das passt besser in einen kommenden Clubabend oder Beitrag.

Das alles wurde beim Clubabend am 25. Mai 2025 besprochen und geübt – wer will, kann gerne bei den nächsten Terminen auch dazu kommen. Dieser Text geht aber bei ein paar Punkten noch mehr ins Detail.

Zu den Downloads und zur Langversion

Alle Zusatzinformationen (anklickbare Links, Downloads us.) sind in der Langversion zu finden.

Link und QR-Code: <https://pcnews.at/markdown/index.html>



Erstellt mit QR Code Generator – kostenfrei, ohne Login, ohne Konto: <https://goqr.me/>. Short URL erstellt mit <https://t.ly/>.

Copyright

