

Room Card

Beschreibung:

Mit Der Room Card kann man Entitäten eines Raumes auf eine Karte packen.

Die Room Card ist in 3 Bereiche eingeteilt:

Wir gehen von oben links nach rechts usw. durch

Oben links:

Das Icon des Raumes als Button, das alle Entitäten auf einmal ein oder ausschaltet dessen Entitäten eine an/aus Funktion haben

Oben rechts:

Eine Info Leiste wo zustände der Entitäten dargestellt werden können, wie Temperatur, füllstände, Offen / geschlossenstände etc.

Mitte:

die Geräte des Raumes die man bedienen will oder mehr Informationen möchte.

Einige Geräte brauchen zum Beispiel hier nicht rein. Wenn man zum Beispiel nur einen Temperatursensor hat. Da reicht die Info oben rechts. Nur zur Veranschaulichung das man in die Mitte nicht jedes Gerät auflisten muss ;-)

Die Abschnitte:

Kartendefinition und das das Icon

Eine Vollständige Liste aller MDI Icons (MaterialDesignIcons) <https://materialdesignicons.com/>

```
type: custom:room-card #Kartendefinition
```

```
title: Wohnen # Der Titel
```

```
entity: light.wohnzimmerlichter #die gruppe oder die entität die damit ein oder ausgeschaltet werden kann. Hier sind alle Lichter im Wohnzimmer drin.
```

```
#denn das ist in der Regel das was man schnell ein und Ausschalten möchte. Gruppen können unter Geräte Helfer erstellt werden.
```

```

icon: mdi:sofa # das Icon
show_icon: true #soll das icon angezeigt werden
content_alignment: center # ausrichtung
tap_action: #was soll bei einmal tippen passieren
  action: navigate #weclhe action hier navigierne
  navigation_path: '#lichtwohnzimmer' hier wird ein Popup aufgerufen. Siehe Wiki Bubble Card
double_tap_action: #was bei doppelt click oder tippen passieren soll
  action: toggle #die Aktion toggle schaltet ein gerät ein/oder aus, immer entgegengesetzt seines jeweiligen
aktiven status

```

Nu gehts nach rechts weiter zu den Info Icons.

Tipp Bei dem Entity Wert, den Geräte typ eintragen und ein einen "." setzten dann kommt eine Vorschlagsliste:

In der Liste setht ganz hinten auch der aktuelle Wert des Services drin.

```

10 double_tap_action:
11   action: toggle
12 info_entities:
13   - entity: sensor.
14     format: sensor.battery_level_1 Battery Level 1
15     styles: sensor.battery_level_2 Battery Level 2
16           #mit t
17           #und j sensor.battery_level_3 Battery Level 3
18           #in un sensor.devolo_thermostat_09356_air_temperature Heizung: Air temperature
19           #ist d sensor.devolo_thermostat_09356_battery_level Heizung Battery level
20     templa sensor.devolo_thermostat_09356_battery_level1 Heizuna Battery level
21     if (entity.state < 20) return 'color: light
22     25)return 'color: red': if (entity.state >

```

State: 100

Nun der Code:

```

info_entities:
  - entity: cover.rolllade_kleines_kuchenfenster
    show_icon: true
    icon:
      conditions:
        - icon: mdi>window-shutter
          condition: below #werte above = über,below=darunter oder state damit kann wird halt der text
verglichen vom state mit dem value
          #current position gibt es bei Geräten die ein Position übermitteln wie Rolläden zum beispiel
          #Wenn geräte spezielle Werte zum abfragen haben dann ins Gerät schon schauen,
          #ansonsten braucht nichts angegeben werden, weil dann wird value aus dem standard sensor der
Entität genommen.
          Für eine Batterie wie hier, habe ich das auskommentiert, aber man soll ja wissen das es das gibt.
          #attribute: current_position

```

```
value: 50
styles:
  color: red
- icon: mdi>window-shutter-open
condition: above
#attribute: current_position
value: 51
styles:
  color: green
```

Es können mehrere Entitäten untereinander gepackt werden in den Info Entitäten. Diese werden dann von links nach rechts angezeigt der Reihenfolge nach, der Letzte Eintrag ist ganz rechts.

Nun gehts weiter mit den Entitäten die auch als Button verwendet werden können und einen Status durch Icon wechsel anzeigen können

```
entities:
- entity: light.living_room_main_light
  name: Wohnzimmer Licht
  state_color: true #soll die farbe wiedergeben die das Entity hat, bei licht die die Lichtfarbe wenn es eine RGB
Lampe ist ansonsten
  #wird halt die standard Farbe für und aus genutzt. Wird State color auskommentiert bleibt es bei ner
standardfarbe die nicht wechselt
  #tap_action: #auch hier kann eine action hinterlegt werden, wird keine Action hinterlegt wird die Standard
Karte der Entität in einem Popup geöffnet
  #action: toggle #die Aktion toggle schaltet ein gerät ein/oder aus, immer entgegengesetzt seines jeweiligen
aktiven status
- entity: media_player.fire_tv_192_168_2_61
  name: FireTV
  show_icon: true
  icon: mdi:set-top-box
  state_color: true
- entity: media_player.echo_show_wohnzimmer
  name: Echo
  show_icon: true
  state_color: true
  icon: mdi:speaker-wireless
- entity: fan.zion_air
  state_color: true
```

```
name: Luftreiniger
- entity: fan.ventilator_wohnen
state_color: true
name: Ventilator
- entity: vacuum.roborock_s8_maxv_ultra
state_color: true
name: Staubi
- entity: sensor.zion_pad_2_battery_level
name: Emilys iPad
state_color: true
- entity: sensor.zion_pad_battery_level
name: iPad Pro
state_color: true
```

Beispiele:

Weite Info Karten Entitäten Beispiele:

Temperatur Sensor

```
- entity: sensor.temperatur_wohnzimmer_sonoff_temperature
format: precision1 #auf wie viele stellen hinterm Komma gerundet, hier eine Stelle,
styles:
  #mit templates kann man sein ganz eigenen style betreiben. mit dem wort entity.state wird aus dem oben
angeben sensor der wert gezogen.
  #und je nach status hier eine farbe zurück geben. Es kann jeder Wert geändert werden, der in Styles
verfügbar ist
  #in unserem Beispiel ist die temperatur unter 20 grad dann hellblau
  #ist der wert größer oder gleich 25 dann rot, bei 23 grad orange und bei 21 hellgrün
template: >
  if (entity.state < 20) return 'color: lightblue'; if (entity.state >=
  25)return 'color: red'; if (entity.state >= 23) return 'color: orange';
  if (entity.state >= 21) return 'color: lightgreen';
```

Rauchmelder

```
- entity: binary_sensor.smoke_detected_1  
  show_icon: true  
  state_color: true
```

Version #8

Erstellt: 1 September 2024 11:41:22 von Admin

Zuletzt aktualisiert: 8 September 2024 12:36:20 von Admin