

# Lovelace Dashboard - Von scratch erstellen

- [Dashboard Grundgerüst und installation Custom Cards](#)
- [Room Card](#)
- [Swipe Card](#)
- [Bubble Card](#)

# Dashboard Grundgerüst und installation Custom Cards

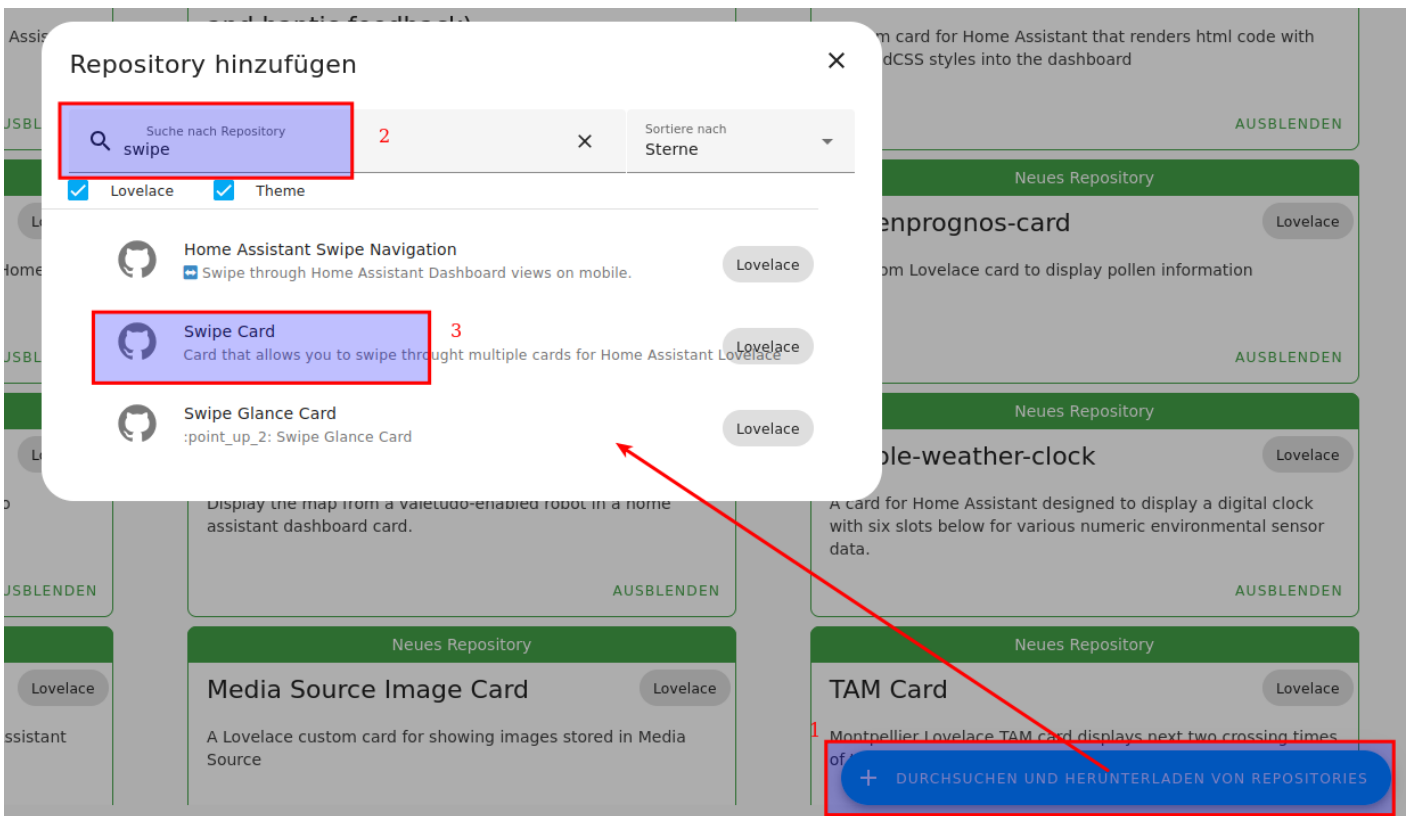
## Beschreibung:

Ein Dashboard von Grund auf erstellen.

Dazu benutzen wir ein paar custom cards und ein Themepack aus dem HACS.

In HACS gehen dann Frontend und dann auf Repo und Suchbegriff eingeben.

Diese Karten werden wir in den nächsten Kapiteln verwenden



welche Karten sind dies :

Addon GIT URL	Suchbegriff in HACS unter repo hinzufügen	Beschreibung
<a href="https://github.com/thomasloven/lovelace-layout-card">https://github.com/thomasloven/lovelace-layout-card</a>	lovelace-layout-card	Dies ist eine Grid Card, mit der Man Karten anordnen kann

<a href="https://github.com/bramkragten/custom-ui/tree/master/swipe-card">https://github.com/bramkragten/custom-ui/tree/master/swipe-card</a>	swipe	Eine Karte in der man mehrere Karten legen kann und einfach durch Wischen von einer zur anderen Karte wechseln, aber nur den Platz einer Karte braucht
<a href="https://github.com/piitaya/lovelace-mushroom/">https://github.com/piitaya/lovelace-mushroom/</a>	mushroom	Eine Sammlung von Karten für diverse Geräte. Die Rolläden Karte finde ich besonders Chic: Mushroom Cover-Kartenkonfiguration
<a href="https://github.com/wassy92x/lovelace-digital-clock/">https://github.com/wassy92x/lovelace-digital-clock/</a>	digital-clock	Eine schlichte moderne Uhr Karte mit Datum
<a href="https://github.com/marcokreeft87/room-card/">https://github.com/marcokreeft87/room-card/</a> entwicklung eingestellt wurde geforkt vom Eintrag drunter	room-card	Eine Karte mit der sich Räume mit einem Oberbegriff Icon ausstatten lässt, was dann eine Gruppe an/ausschalten kann. Dann eine Info Symbolzeile hat und dann noch eine Geräte Liste beinhalten kann.
<a href="https://github.com/viseradius/room-card">https://github.com/viseradius/room-card</a> ist ein Fork von dem drüber	room-card	Eine Karte mit der sich Räume mit einem Oberbegriff Icon ausstatten lässt, was dann eine Gruppe an/ausschalten kann. Dann eine Info Symbolzeile hat und dann noch eine Geräte Liste beinhalten kann.
<a href="https://github.com/berrywhite96/lovelace-shutter-row">https://github.com/berrywhite96/lovelace-shutter-row</a>	shutter row	Eine Karte mit mehr Einstellungsmöglichkeiten als die Mushroom Rolläden Karte.
<a href="https://github.com/Cloos/Bubble-Card/">https://github.com/Cloos/Bubble-Card/</a>	bubble-card	Eine Karte die als Aktion aufgerufen wird. Sprich ein Dashboard was in einem Popup aufgerufen wird.
<a href="https://github.com/bramkragten/weather-card">https://github.com/bramkragten/weather-card</a>  wird von jemand anders auch gepflegt, beide repos sind aktiv  <a href="https://github.com/r0bb10/weather-card">https://github.com/r0bb10/weather-card</a>	weather card	Eine Wetterkarte mit vielen Details und animierten Icons
<a href="https://github.com/ricardoquecria/caul-e-themes-pack-1">https://github.com/ricardoquecria/caul-e-themes-pack-1</a>	caule	Ein Theme mit mehreren Varianten
<a href="https://github.com/custom-cards/button-card">https://github.com/custom-cards/button-card</a>	button-card	Eine Card die einen Button beinhaltet wo man aber ein Icon hinzufügen kann, mit blinken oder anderen States etc.

# Designen

## Grundgerüst:

Den Kiosk mode aktivieren, wenn man dies denn möchte, ansonsten überspringen

```
kiosk_mode:  
  user_settings:  
    - users:  
      - wall-unit  
    kiosk: true  
    hide_header: true  
    hide_sidebar: true  
    hide_menu_button: true  
    hide_overflow: true
```

Danach unser Layout:

Wir wollen 3 Spalten und die Anzahl der Zeilen soll sich nach unten automatisch erweitern. Ist die maximale breite des Fensters nur noch 93 Pixel oder weniger dann nehme eine zweite Anordnung.

Hier bei uns alles untereinander. Wofür ist das. Wenn das Fenster kleiner als 93 Pixel ist zum Beispiel auf dem Smartphone soll alles untereinander angezeigt werden. Und schon ist das Dashboard responsive

Beschreibung des Codes

```
views:  
  #Unser Theme aus dem Theme pack, Schwarz mit Blau find ich ganz gut  
  - theme: Caule Black Blue  
  #Title vom Tab  
  title: Home  
  path: grid  
  type: custom:grid-layout  
  layout:  
    #die Breiten der Spalten  
    grid-template-columns: 25% 40% 35%  
    Die Breite der Zeilen, sollen hier automatisch angepasst werden  
    grid-template-rows: auto  
    #Unsere Areas, sprich Spaltennamen.  
    Werden zwei Spaltennamen gleich benannt unter / übereinander so werden die zellen zu einer großen
```

## Verbunden

#Wir denken da an Excel.

#Wir haben die Erste Zeile ist unser Message bereich, ähnlich wie die Statusleiste beim Smartphone

#Headerlinks header mitte und headerrechts kann man reinpacken was man möchte das gleich bei den headern untern

#Dann haben wir einen Trenner für die Footer. Falls man unten noch was drin haben will, angeleht an eine Webseite mit Footern.

#Dafür stehen uns auch Zwei Zeilen zur Verfügung

grid-template-areas: |

"messages messages messages"

"headerl headerm headerr"

"headerlu headerm headerru"

"divider divider divider"

"left middle right"

"footer1 footer2 footer3"

"footer4 footer5 footer6"

mediaquery: # ist die maximale breite nur noch 93 pixel oder weniger dann nehme diese anordnung. hier bei uns alles untereinander

'(max-width: 932px)':

grid-template-columns: 100%

grid-template-areas: |

"messages"

"headerl"

"headerm"

"headermu"

"headerr"

"headerlu"

"headerru"

"divider"

"left"

"middle"

"right"

"footer1"

"footer2"

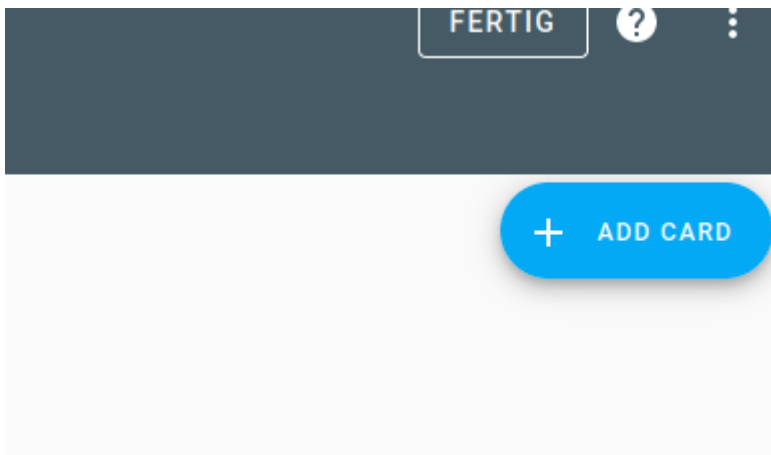
"footer3"

"footer4"

"footer5"

"footer6"


Nun können Karten in den Platzhaltern positioniert werden.  
Jetzt verlassen wir wieder den Code-Editor.  
Dann einfach eine Karte hinzufügen.



Zum testen die Wettervorhersage



Diese steht jetzt am Anfang, wir müssen ihr noch die Position mitteilen

komplettneu 

← HOME GERUEST  → +



**Sonnig**

Forecast Home

**17,9 °C**

25,4 °C / 17,9 °C

So



25,4°

17,9°

Mo



27,3°

14,1°

Di



26,2°

17,3°

Mi



22,8°

18,4°

Do



26,2°

17,3°

BEARBEITEN




1



Nun gehen wir auf der Karte auf bearbeiten

komplettneu 

← HOME GERUEST 



Sonnig

Forecast Home

17,9 °C

25,4 °C / 17,9 °C

So



25,4°

17,9°

Mo



27,3°

14,1°

Di



26,2°

17,3°

Mi



22,8°

18,4°

Do



26,2°

17,3°

BEARBEITEN



1



und in den Code Editor

## Wettervorhersage-Kartenkonfiguration

KONFIGURATION
SICHTBARKEIT

Entität (erforderlich)\*  
 Forecast Home

Name

Zweites Infoattribut

Theme (optional)

Vorhersagetyp auswählen

Täglich

Stündlich

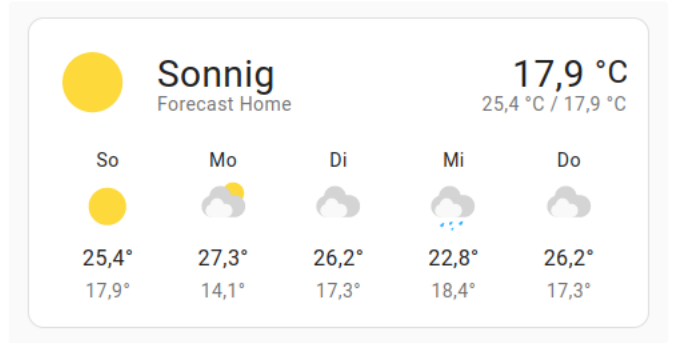
Anzuzeigendes Wetter

Aktuelles Wetter und Vorhersage anzeigen

Nur aktuelles Wetter anzeigen

Nur Vorhersage anzeigen

CODE-EDITOR ANZEIGEN
ABBRECHEN   SPEICHERN



Nun sehen wir den Editor und fügen folgendes darunter:

## Wettervorhersage-Kartenkonfiguration

```

1 type: weather-forecast
2 entity: weather.home
3 forecast_type: daily
4

```

VISUELLEN EDITOR ANZEIGEN
ABBRECHEN   SPEICHERN

Der Positions-code

view\_layout:

grid-area: <positionsname>

Das waren unsere definierten Psoitionen:

"messages messages messages"

"headerl headerm headerr"

"headerlu headerm headerru"

"divider divider divider"

"left middle right"

"footer1 footer2 footer3"

"footer4 footer5 footer6"

Beispiel:

view\_layout:

grid-area: headerr

Der Komplette Code der Karte:

type: weather-forecast

entity: weather.home

forecast\_type: daily

view\_layout:

grid-area: headerr

Nun steht unsere Karte im der headerr rechts Position

The screenshot shows a smart home dashboard interface. At the top, there is a dark navigation bar with the text 'komplettneu' and a pencil icon on the left, and 'FERTIG' with a question mark and a three-dot menu icon on the right. Below the navigation bar, there is a light gray area with a navigation bar containing '< HOME GERUEST >' and a plus sign. On the right side of the dashboard, a weather forecast card is displayed. The card has a yellow sun icon and the text 'Sonnig Forecast Home'. The current temperature is '17,9 °C' with a range of '25,4 °C / 17,9 °C'. Below this, there is a table of daily forecasts for the next five days (So, Mo, Di, Mi, Do) with icons and temperature ranges. At the bottom of the card, there is a 'BEARBEITEN' button and a minus sign with a '1' in a circle, followed by a plus sign and a three-dot menu icon. A blue 'ADD CARD' button is located at the bottom right of the dashboard.

So	Mo	Di	Mi	Do
25,4° 17,9°	27,3° 14,1°	26,2° 17,3°	22,8° 18,4°	26,2° 17,3°



# 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
```

# Swipe Card

## Beschreibung:

Die Swipe Karte ist eine Karte auf der die Größe einer Karte einnimmt aber darauf mehrere Karten hinterlegt werden können. In einer Art Karussell kann man dann durch die Karten wischen. Wie bei einer Diashow von Webseiten wo man ja auch durch die Bilder wischen kann. Hier kann man dann noch angeben ob Punkte, Striche oder gar nichts angezeigt werden soll auf welcher Seite man sich befindet.

## Installation:

Aus dem HACS Store die Swipe Karte installieren. Suchbegriff: swipe card

Auf Add Karte klicken und die Swipe Karte auswählen.

Leider ist die Swipe Karte nur mit YAML konfigurierbar, aber trotzdem simpel.

Unser Basiscode nachdem wir die Karte hinzugefügt haben.

Nicht sehr spektakulär.

```
type: custom:swipe-card
cards: []
```

Als erstes vergeben wir wieder eine Position, vorausgesetzt Ihr nutzt die Grid Card, sonst natürlich nicht

```
type: custom:swipe-card
view_layout: #unsere positionsangabe
  grid-area: middle
cards: []
```

Nun stellen wir die Eigenschaften unserer Swipe Karte ein.

Hier eine Komplettiliste aller Parameter auf der Seite des Entwicklers:

<https://swiperjs.com/swiper-api#parameters>

```
type: custom:swipe-card
view_layout:
  grid-area: middle
parameters:
```

```

spaceBetween: 8 #abstand zwischen den swipe Karten beim wischen siehe Screenshot unten.
effect: coverflow #der typ der animation beim wechseln / wischen der karte
grabCursor: false #Auf System mit Mauzeiger unterstützung wird dann hier das Handsymbol angezeigt,
#ich deaktiviere es, weil beim anklicken von Elementen doch stören kann, auf mobilen geräten gibt es eh keinen
Cursor
#außer es ist eine maus angeschlossen.
centeredSlides: false #da es dann probleme mit Vertikalen und Horizontalen SStack Karten innerhalb der Swipe
Karte kommt und ich
#diese Karten aber darin einbetten möchte
slidesPerView: auto #Dies hab ich auf Auto gelassen
coverFlowEffect: #wie der Flow aussehen soll. Einfach ausprobieren oder so lassen.
  rotate: 50
  stretch: 0
  depth: 100
  modifier: 1
  slideshadows: true
pagination:
  type: progressbar #wie der Seitenzähler angezeigt werden soll, mögliche werte: []'progressbar' | 'bullets' |
'fraction' | 'custom'

cards: []

```

Damit hätten wir die Grundeinstellung erledigt.

Nun können wir darin, unsere Karten hinzufügen.

Beispiel mit einer Vertikal Stack Karte und einer Rollanden Karte.

Ein Nachteil hat die Swipe Karte. Man kann darin nicht den Visuellen Editor für die darin enthaltenden Karten nutzen.

Tipp: Die Karte erst fertig konfigurieren, Code kopieren und in die Vertical / bzw. Sipe Karte einfügen, je nach dem.

Im Ersten Kart Abschnitt der Vertical Stack und im Vertical Stack die Karte für die Rollade. Wir können jetzt weite Karten im Vertical stack hinzufügen. Ich hab den Abschnitt Parameter hier mal rausgelassen, damits übersichtlicher ist

```

type: custom:swipe-card
view_layout:
  grid-area: middle
#hier wären die parameter
cards:
  - type: vertical-stack

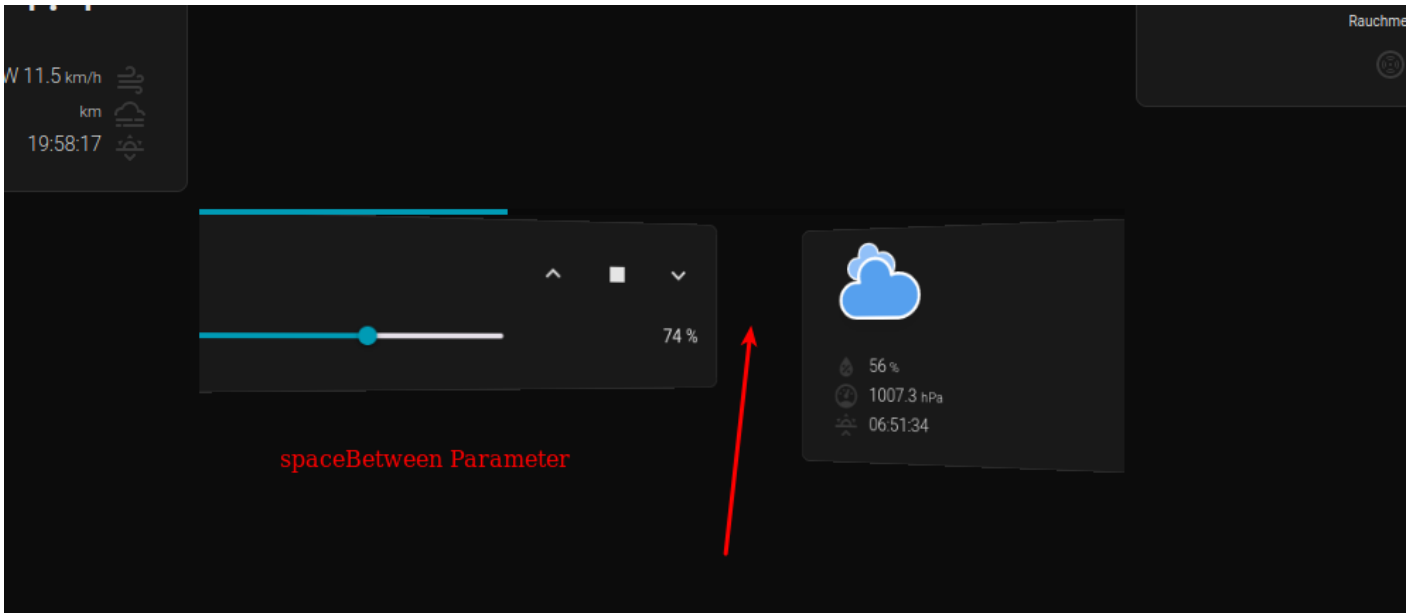
```

```
cards:  
- type: custom:shutter-row  
  move_up_button: {}  
  move_stop_button: {}  
  move_down_button: {}  
  entity: cover.rollade_wohzimmer_cover
```

Weiteren Vertical Stack für die zweite Seite. Da blenden wir jetzt ne Wetter Karte ein. Auch schöne Custom Wetter Card mit Animation.

```
type: custom:swipe-card  
view_layout:  
  grid-area: middle  
  #hier wären die parameter  
cards:  
- type: vertical-stack  
  cards:  
    - type: custom:shutter-row  
      move_up_button: {}  
      move_stop_button: {}  
      move_down_button: {}  
      entity: cover.rollade_wohzimmer_cover  
- type: vertical-stack  
  cards:  
    - type: custom:weather-card  
      entity: weather.home  
      forecast: true  
      details: true  
      current: true  
      hourly_forecast: true
```

Uns so sieht das Ergebnis aus, beim Wischen



Fertig

# Bubble Card

Beschreibung:

Die Bubble Card ist eine Custom Lovelace Card, die es ermöglicht VerticalStack Karten in einem Popup anzuzeigen.