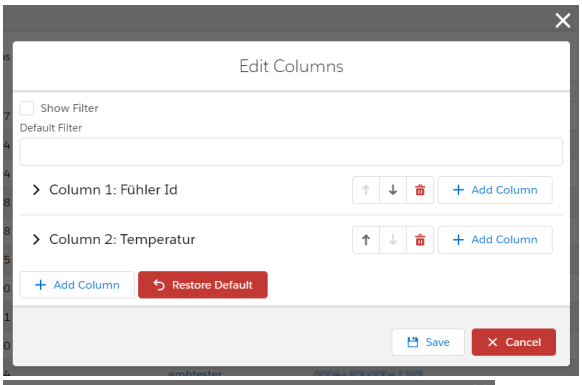
Table Config

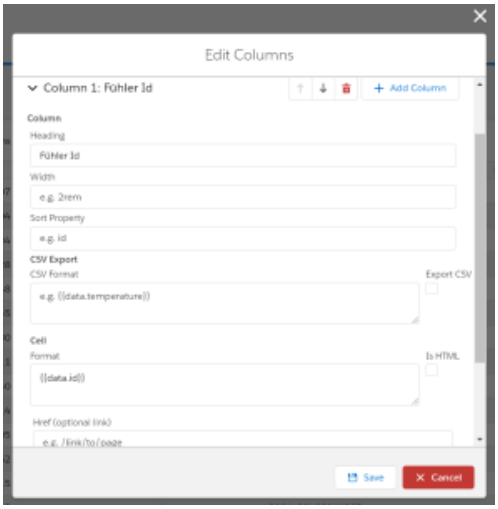
Audience: Platform Admins

Platform Admins can manage Table Configs for different tables in the Platform. (Currently only for Device Data) The table config is shared between all devices of a device type.

Navigate to:

- Configuration -> Device Types -> Select a Device Type -> Data Table Config
- Data -> Gear Icon above table
- Devices -> Select a Device -> Device Data -> Gear Icon above table





Column Fields:

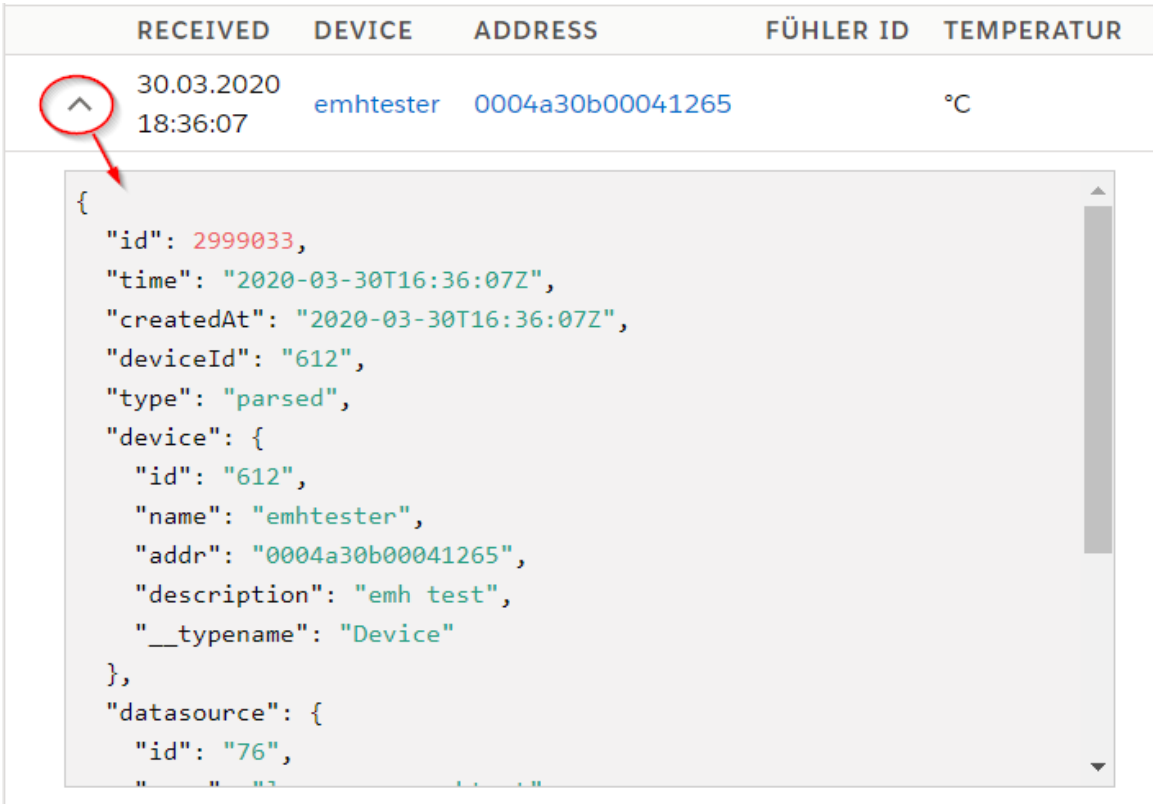
- **Heading:** Title of the column
- **Width:** Optional width of the column (use CSS)
- **Sort Property:** Optional: Path to property to allow sorting
- **CSV Export:**
- **Export CSV**: Check to enable CSV export for the column
- **CSV Format:** Format for the CSV export (supports Format Tamplates)
- **Cell:**
- **Format:** Format of the value inside the table (supports Format Tamplates)
- **Is Html:** HTML is not escaped. This is unsafe. Do not use!
- **Href:** Add a link to another page to the value (supports Format Tamplates)

Sort/Filter Property

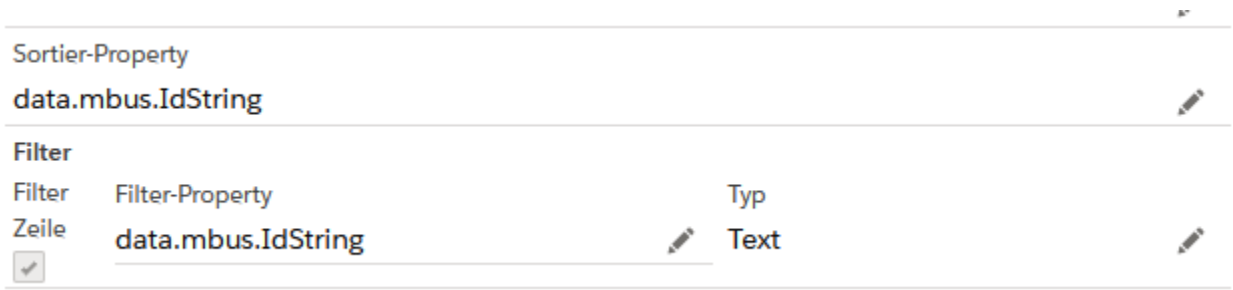
The property is entered as Path inside the JSON column Object:

Table of Contents

- Sort/Filter Property
- Format Templates
  - Accessing Arrays
  - Handlebar Template Helpers
    - Cascading helpers
    - date
    - duration
    - durationHumanize
    - durationAs
    - fromNow
    - typeof
    - toFixed (number of decimals)
    - numberFormat
    - CSV (requires platform > v1.8.0)
    - Math
    - Boolean operations
    - If Clauses
    - icon



To enable sorting/filtering on device name use device.name as property. In the devices Properties are field present that have a point inside the fieldname. These points needs to be protected by quotes. So for the device.vbat property use: properties.device"."vbat.value



To enable a Filter

- tick the "Filer Column" box.
- Enter the path of the property based on which the filter is applied
- choose a filter Type.

The Text filter will display a Text Box enable to search every entry containing the word entered. (Case insensitive) Alternativly you can enter a leading star (*word) to only search entry that end with "word" or a trailing star (word*) to only search for entries that start with "word".

In the boolean Filter you get a drop down Menu where you can choose if no filter is applied or you want to filter on "true" or "false".

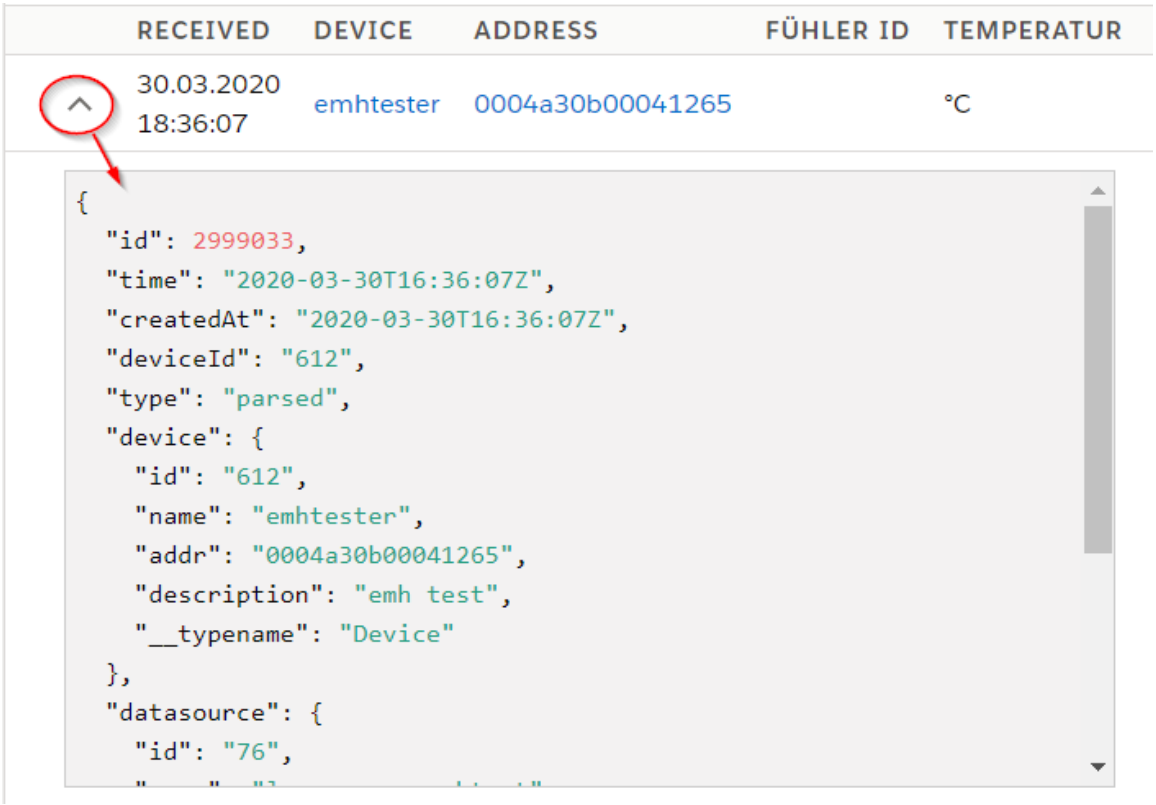
Formats are handlebar templates. See: "Format Template" below.

Format Templates

Formatting values in HTML or CSS output is done via Handlebar Templates. Please also have a look into the Handlebar Documentation.

You can not use HTML directly for security reasons. There is a checkbox `Is HTML` to skip the escaping. We do not recommend using it and it might get removed in future versions.

The underling data structure can be seen when expanding the table:



Parser output is saved in the `data` field. But you can also render all other fields. To replace part of the template with variable data you need to put the variable in double curly brackets: e.g. `{{variable}}`

Example

To display the `temperature` field of the parser output you can simply write `{{data.temperature}}`.

For more advanced formatting you will need Handlebar Helpers.

Accessing Arrays

Array elements can be reached with `.[<idx>]`

> The JavaScript syntax with [idx] (e.g. data[1]) does not work in handlebars! A dot in front of the [x] is needed.

Example

Template
`{{data.[1]}}`

Data

`{`
`  "data": [`
`  "a", "b", "c"`
`  ]`
`}`

Output
b

Handlebar Template Helpers

Handlebar Helpers can be used to apply advanced formatting to data outputs. Starting from conditionals to simple calculations.

All built-in Handlebar Helpers can be used.

Helpers are used in the following format:
`{{helper_name <required_parameters> [optional_parameters]}}`

Cascading helpers

To cascade helpers use round braces

Example
`{{toFixed (div data.vsysCurrent_mV 1000) 1}}`

date

Format a given time and date.

`{{date <date> [format]}}`

- `date` must be a valid input for moment(). When using timestamps it must be in milliseconds.
- `format` Optional parameter, not supported for CSV export. ⚠️ **Deprecated:** Please use the named parameter (see below).
- Optional Named parameters:
  - `format` must be a valid moment.format() string. (will be ignored by csv export job yet [25.11.2022])
  - `tz` Timezone Name, e.g. "Europe/Berlin". See also: List of time zones

Example

Template

```
{{date data.time}}

//Example with optional params
{{date data.time format='dddd, MMMM Do YYYY, h:mm:ss a' tz='Europe/Berlin'}}
```

Data

```
{
  "data": {
    "time": 1585823609000
  }
}
```

Output

```
02.04.2020 10:33:29
```

duration

Format given duration.

```
{{duration <duration> [format]}}
```

- `duration` must be a valid input for moment.duration(). When using timestamps it must be in milliseconds.
- `format` must be a valid moment.format() string.

Example

Template

```
{{duration duration}}
```

Data

```
{
  "data": {
    "duration": 10000
  }
}
```

Output

```
00:00:10
```

durationHumanize

Format a given duration in a human readable format.

```
{{durationHumanize <duration>}}
```

- `duration` must be a valid input for moment.duration(). When using timestamps it must be in milliseconds.

Example

Template

```
{{durationHumanize data.duration}}
```

Data

```
{
  "data": {
    "duration": 10000
  }
}
```

Output

```
10 seconds
```

durationAs

Get a given duration in a certain unit of time.

```
{{durationAs <duration> [unit_of_time]}}
```

- `duration` must be a valid input for moment.duration(). When using timestamps it must be in milliseconds.
- `unit_of_time` must be a valid input for moment.duration().as()

Example

Template

```
{{durationAs data.duration "milliseconds"}}
```

Data

```
{
  "data": {
    "duration": 10000
  }
}
```

Output

```
10000
```

fromNow

Display the time between now and a given time. See also: moment().fromNow().

`{{fromNow <date>}}`

- `date` must be a valid input for moment(). When using timestamps it must be in milliseconds.

Example

Template

```
{{fromNow data.time}}
```

Data

```
{
  "data": {
    "time": 1585823609000
  }
}
```

Output

```
3 days ago
```

typeof

Display the js type of a value. Useful for debugging.

`{{typeof <value>}}`

- `date` must be a valid input for moment(). When using timestamps it must be in milliseconds.

Example

Template

```
{{typeof data.value}}
```

Data

```
{
  "data": {
    "value": 1337
  }
}
```

Output

```
number
```

toFixed (number of decimals)

Display a number with given amount of decimal places. For more advanced formatting see `numberFormat`.

```
{{toFixed <value> [decimals]}}
```

- `decimals` number of decimal places (Default: 2)

Example

Template

```
{{toFixed data.value 3}}
```

Data

```
{
  "data": {
    "value": 1.2345
  }
}
```

Output

```
1.234
```

numberFormat

Format a number.

```
{{numberFormat <value> [options]}}
```

options:

- `thousandsSep` separator between 3 digits (Default: locale dependent - ',' for germans)
- `decimalSep` decimal separator (Default: locale dependent - '.' for germans)
- `decimals` number of decimal places (Default: 2)

Example

Template

```
{{numberFormat data.value thousandsSep="." decimalSep=","}}
```

Data

```
{
  "data": {
    "value": 1024.2345
  }
}
```

Output

```
1.024,23
```

CSV (requires platform > v1.8.0)

Format an array as CSV string.

`{{csv <value> [options]}}`

options:

- `separator` separator between entries (Default: ';')
- `decimalSep` decimal separator (Default: locale dependent - '.' for germans))

Example

Template

```
{{csv data.value separator=";" decimalSep=","}}
```

Data

```
{
  "data": {
    "value": [1024.2345, 5]
  }
}
```

Output

```
1024,23;5
```

`replace`

Replaces a part of a string by an alternativ. Use "" as replaceWith to delete searched part. Will return the input when applied on values that are not of type string.

`{{replace "String to search in" "searchFor" "replaceWith"}}`

`Example`

```
//use
{{replace "Hallo Lobaro welcome to the Internet" "Lobaro" "oraboL"}}
output: "Hallo oraboL welcome to the Internet"

//for a data field in json object
{{replace data.value  "wmbusapp-" ""}}

//object to apply:
{
  "data": {
    "value": "wmbusapp-v101"
  }
}

//output
"v101"
```

Math

Mathematical operations.

- `{{ceil <value>}}` round up to integer
- `{{floor <value>}}` round down to integer
- `{{div <nom> <denom>}}` returns nom / denom
- `{{mul <a> <b>}}` returns a * b (requires Platform > v1.8.0)
- `{{max <list> [property_path]}}` returns the biggest element from the list. Compares the given `property_path`.

Boolean operations

Boolean operations, useful for `#if` conditions.

- `{{not <value>}}` negate the value
- `{{eq <v1> <v2>}}` v1 == v2
- `{{lt <v1> <v2>}}` v1 < v2
- `{{lte <v1> <v2>}}` v1 <= v2
- `{{gt <v1> <v2>}}` v1 > v2
- `{{gte <v1> <v2>}}` v1 >= v2

Example

```
{{#if (not value)}}
{{value}} is falsy
{{else}}
{{value}} is truthy
{{/if}}
```

If Clauses

See also: https://handlebarsjs.com/guide/builtin-helpers.html

```
{{#if value}}
{{value}} is truthy
{{else}}
{{value}} is falsey
{{/if}}
```

Example, don't return "NaN": {{#if (not (eq data.value NaN))}}{{data.value}}{{/if}}

icon

Render an Icon

```
{{icon <name> [category]}}
```

- `name` name of the icon. Find all icons here: SLDS Icons
- `category` category of the icon (Default: "utility").

Device

confi

{{config }}