MMISP Frontend Documentation

Design Phase

MMISP Frontend Team

Table of contents

1. Intro

The project "Modern MISP Frontend" is implemented as a client-side rendered SvelteKit application.

SvelteKit applications are constructed out of Components. They are `.svelte` files found in the `src` directory of this project. See here for general documentation about Svelte Components.

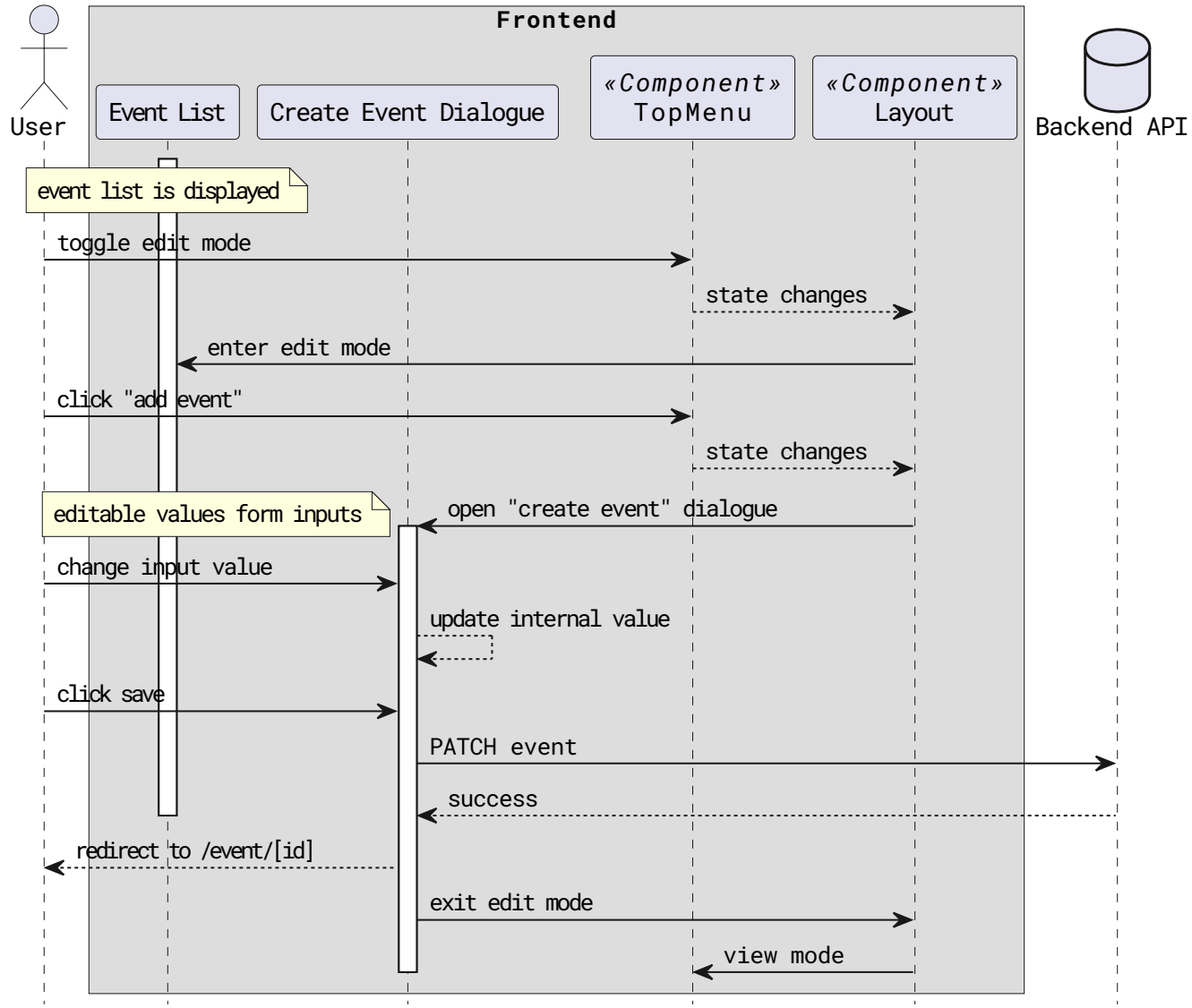
All routes of the application are represented by specific Components called Pages. They are the files called `+page.svelte` in the `src/routes` directory of this project. See here for general documentation about SvelteKit Pages.

Pages "inherit" from specific Components called Layouts placed higher up in the route tree by automatically being placed into the Layout's default slot. They are the files called `+layout.svelte` in the `src/routes` directory of this project. See here for general documentation about SvelteKit Layouts.

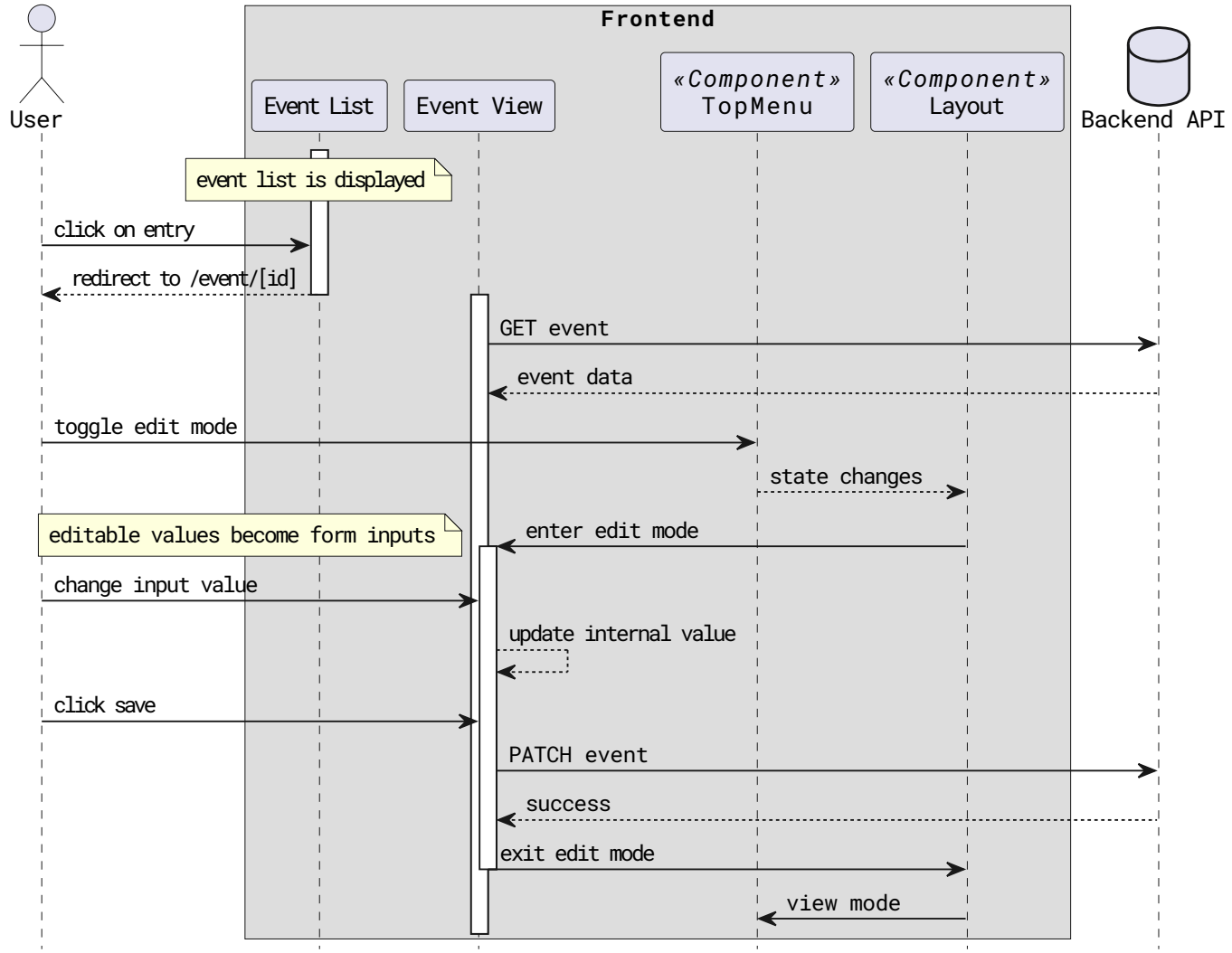
Note that the inter-component dependencies and the specifics of exposed props, slots and events may not fully represent the final state of the application, as agreed upon with our supervisors.

2. Sequence Diagrams

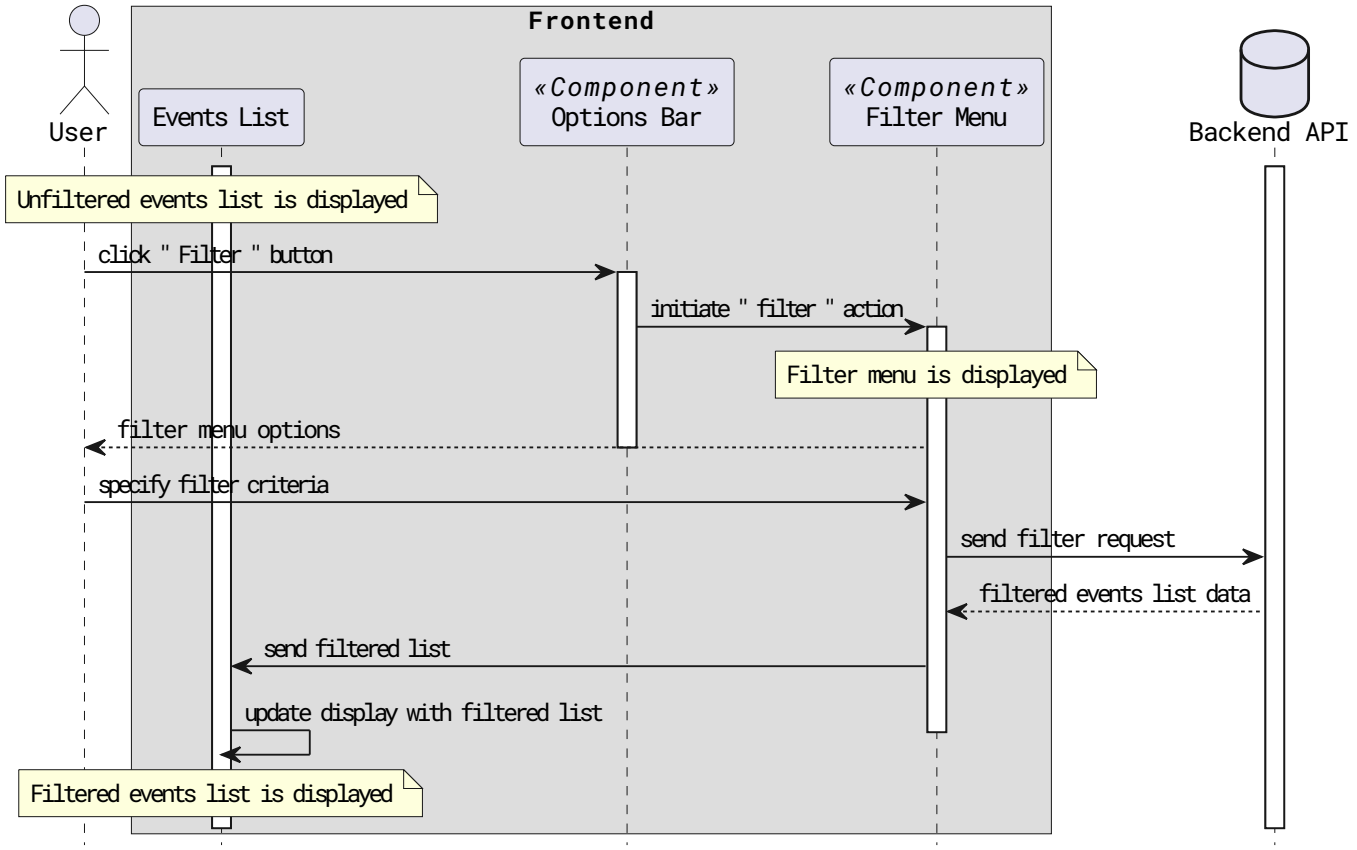
2.1 Add Event



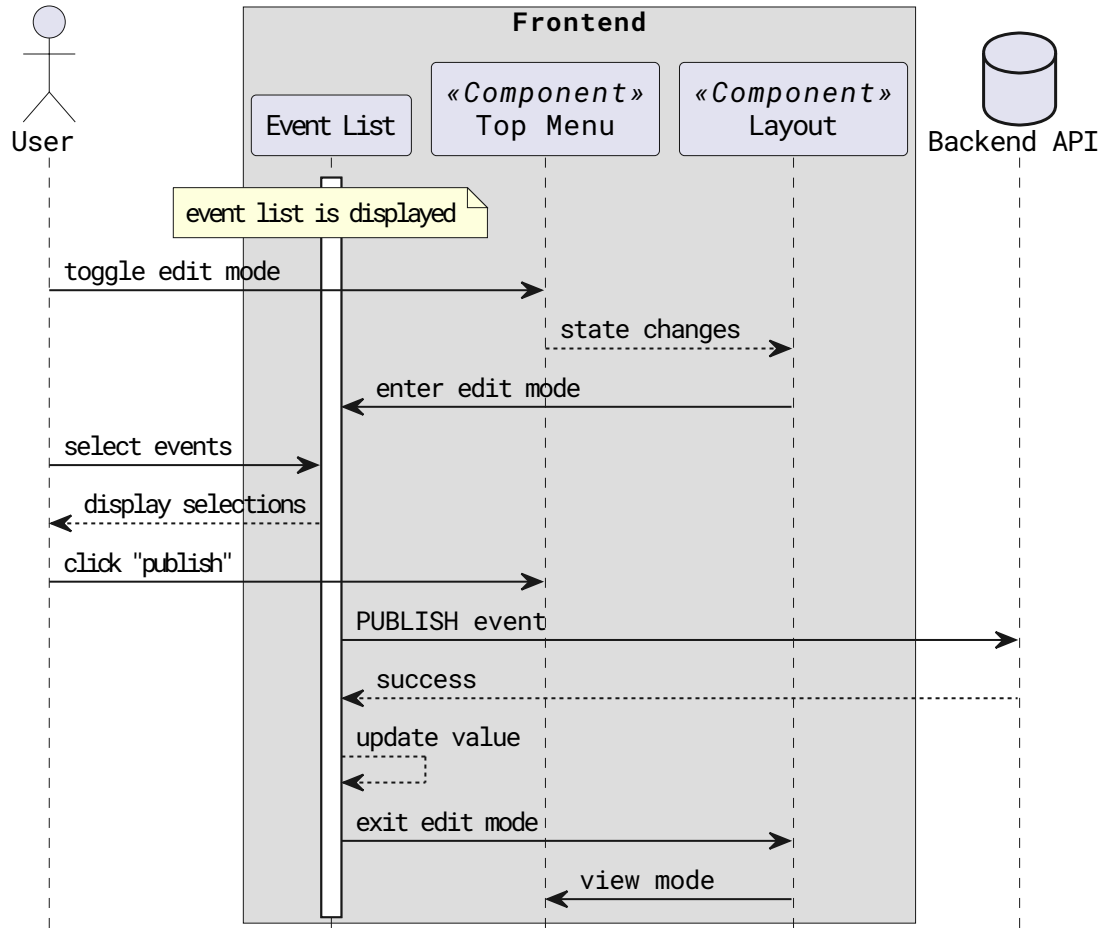
2.2 Edit Event



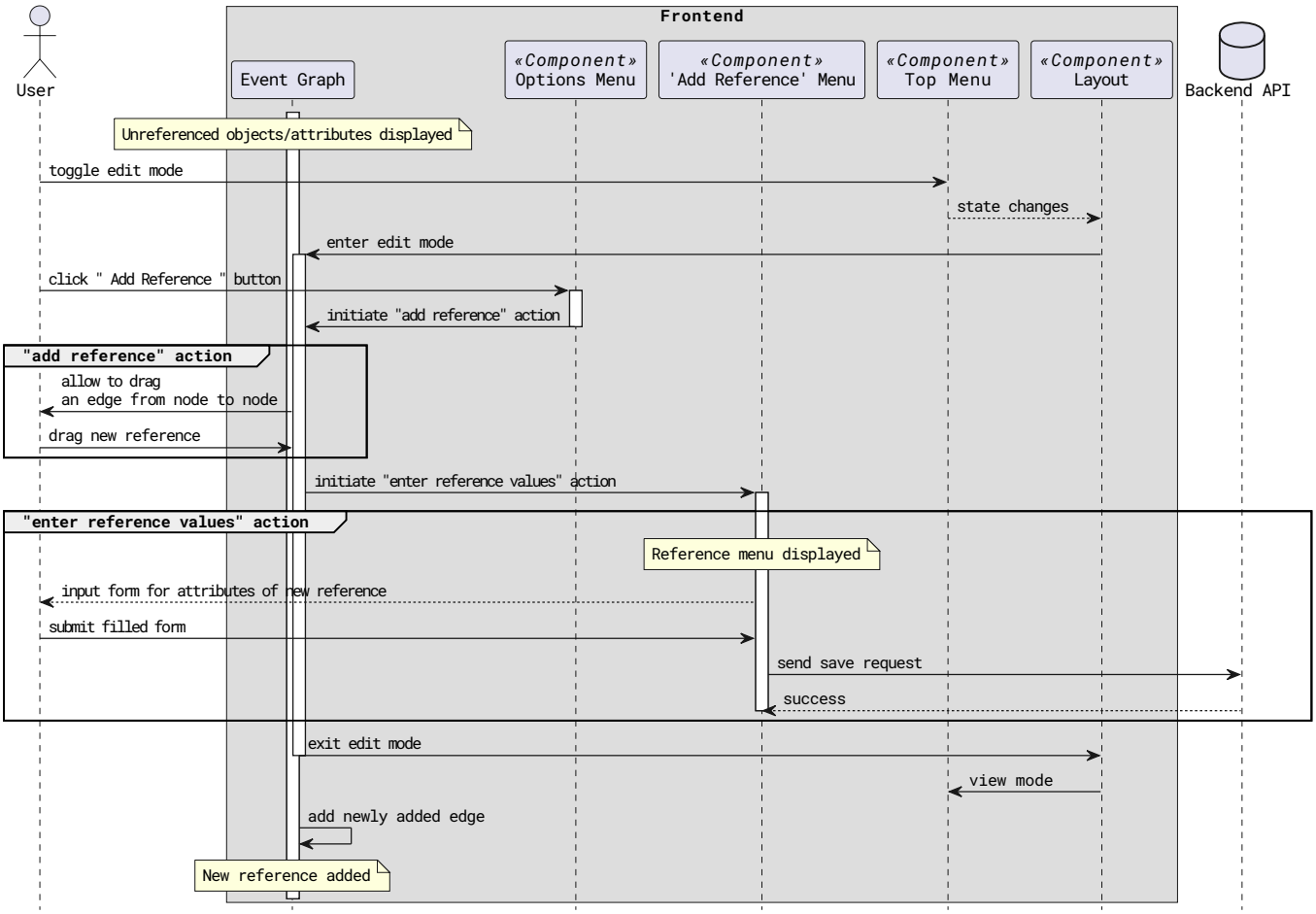
2.3 Filter Events



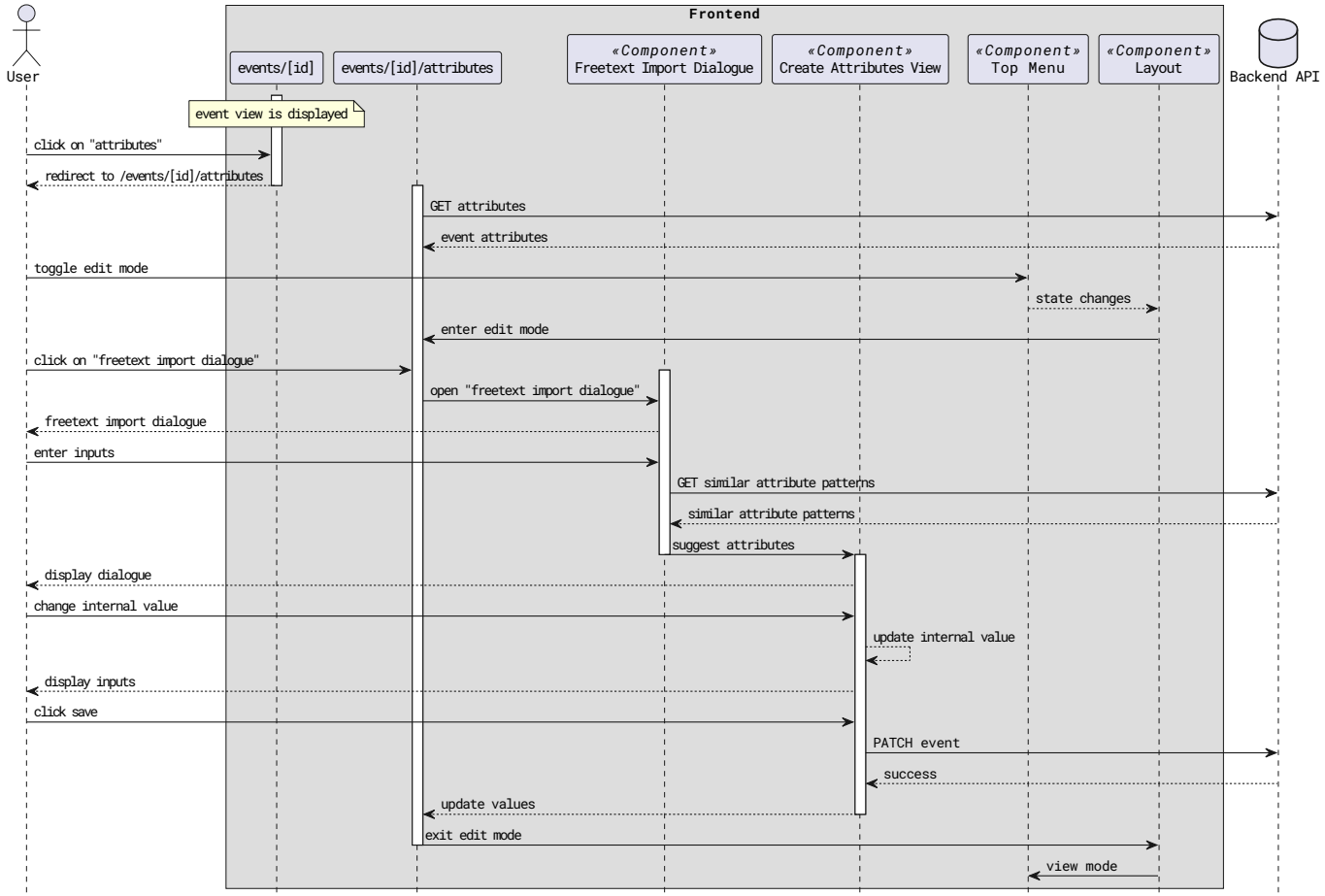
2.4 Publish Event



2.5 Add Reference

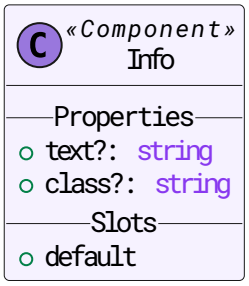


2.6 Freetext Import



3. Components

3.1 Info



Displays a text with a default background color of `surface1`. Also sets the default padding and border radius. You can override this by passing your own classes.

3.1.1 Props

text: `string | undefined`

The text to be displayed.

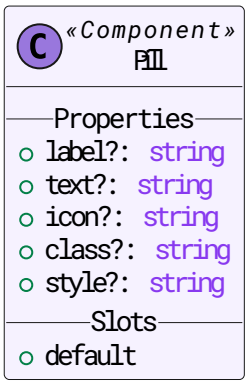
class: `string | undefined`

Additional classes to be applied.

3.1.2 Slots

default

3.2 Pill



A pill component. A pill is a small rounded rectangle with a label and/or text and/or icon.

Slot:

The content of the pill. If no slot is provided, the text prop will be used.

3.2.1 Props

label: `string | undefined`

The label of the pill. Will be placed on the left side of the pill. The background of the label is `bg-crust`.

text: `string | undefined`

The text of the pill. Will be placed in the middle of the pill.

icon: `string | undefined`

The icon of the pill. Will be placed on the left side of the pill. If a label is present, the icon will be placed on the left side of the label.

class: `string | undefined`

Class that should be applied to the pill.

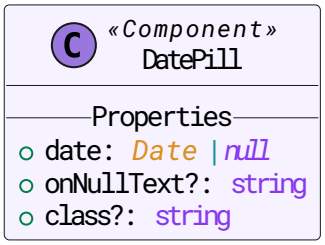
style: `string | undefined`

Some style overrides. When possible, the `class` prop should be used instead.

3.2.2 Slots

default

3.3 DatePill



Uses Pill.

Displays a date in a pill with the default format. The date format can be configured in the `config.ts` file.

3.3.1 Props

date: `Date | null`

The date of the to be displayed.

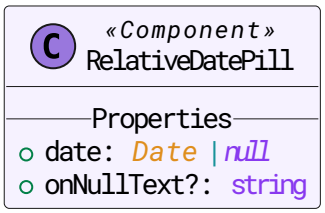
onNullText: `string | undefined`

The text that should be displayed if the date is null.

class: `string | undefined`

Class that should be applied to the pill.

3.4 RelativeDatePill



Uses DatePill.

Displays a relative date in a pill. The color of the pill is based on the date.

- If the date is in the past, the pill will be red.
- If the date is over one week in the future, the pill will be green.
- If the date is less then one week in the future, the pill will be orange.

3.4.1 Props

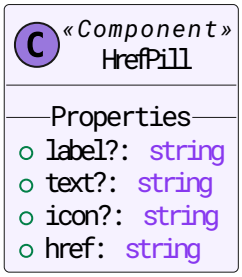
date: `Date | null`

The date of the to be displayed.

onNullText: `string | undefined`

The text that should be displayed if the date is null.

3.5 HrefPill



Uses Pill.

A pill component that acts as a link. This pills text will be blue:

3.5.1 Props

label: `string | undefined`

The label of the pill. Will be placed on the left side of the pill. The background of the label is `bg-crust`.

text: `string | undefined`

The text of the pill. Will be placed in the middle of the pill.

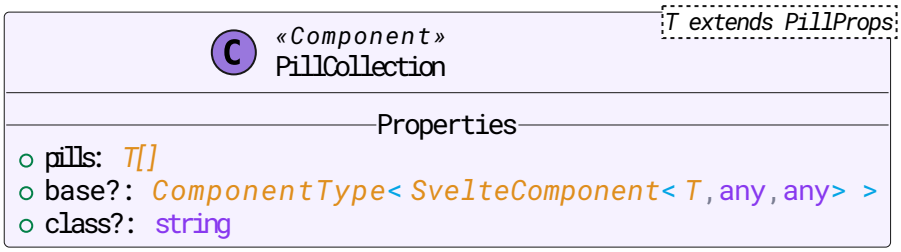
icon: `string | undefined`

The icon of the pill. Will be placed on the left side of the pill. If a label is present, the icon will be placed on the left side of the label.

href: `string`

The target URL of the pill, which will be navigated to when the pill is clicked.

3.6 PillCollection



Uses Pill.

Displays a collection of pills. The pill component that should be used for each pill can be specified by setting the `base` prop.

3.6.1 Props

Generics: `T extends PillProps`

pills: `T[]`

The Pills that should be displayed.

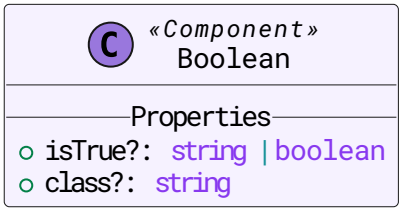
base: `ComponentType<SvelteComponent<T, any, any>> | undefined`

The pill component to be used for each pill.

class: `string | undefined`

The class of the pill wrapper.

3.7 Boolean



Uses Pill.

Displays a boolean value as a text using the Info component. The background is green if the value is true and red if the value is false.

3.7.1 Props

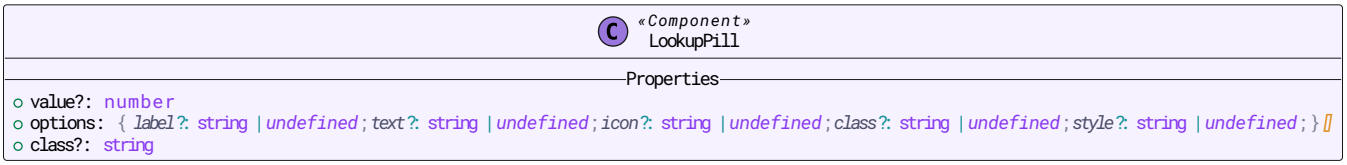
isTrue: `string | boolean | undefined`

Displays a boolean value as a text using the Info component. Also parses strings to booleans. String must be either 'true' or 'false'.

class: `string | undefined`

Additional classes to be applied to the Info component.

3.8 LookupPill



Uses Pill.

Converts the value given by the `value` prop to an entry from the `options` lookup array and displays the result as a pill.

3.8.1 Props

value: `number | undefined`

The index corresponding to the entry in the `options` array.

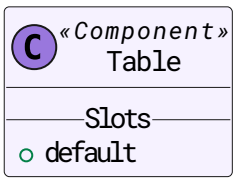
options: `{ label?: string | undefined; text?: string | undefined; icon?: string | undefined; class?: string | undefined; style?: string | undefined; }[]`

Array with props of Pills, indexed by `value`.

class: `string | undefined`

The class of the pill.

3.9 Table



Creates an HTML `table` element with specific styling.

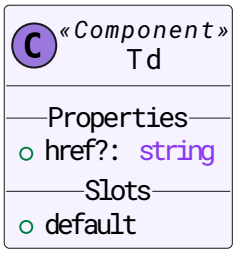
Slot:

The full table.

3.9.1 Slots

default

3.10 Td



Creates an HTML `td` element for the table with specific styling.

Slot:

- The content of the td.

3.10.1 Props

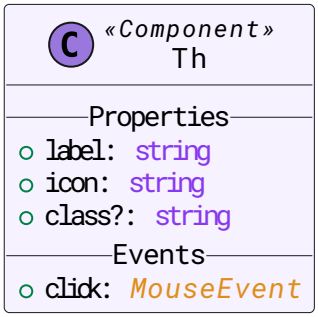
href: `string | undefined`

The url to navigate to when clicking on the `td`.

3.10.2 Slots

default

3.11 Th



Creates an HTML `th` element for the table with specific styling.

Slot:

- The content of the th.

3.11.1 Props

label: `string`

The label of the column.

icon: `string`

The icon of the column.

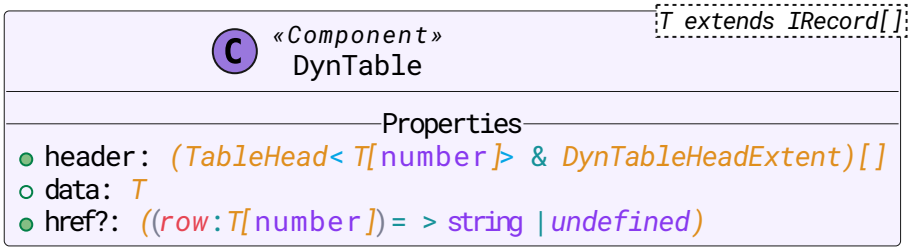
class: `string | undefined`

The class to be applied to the table head.

3.11.2 Events

click: `MouseEvent`

3.12 DynTable



Uses Table, Td, Th.

Creates a dynamic Table using the `header` and `data` props.

The `header` props specifies the columns of the table, while the `data` prop provides rows of data that conform to the structure of the header.

Type safety of this is enforced at compile time using Typescript.

3.12.1 Props

Generics: `T extends IRecord[]`

header: `(TableHead<T[number]> & DynTableHeadExtent)[]`

The header of the table. Also includes the icon and the href. When setting this, it's recommended to use the createTableHeadGenerator util function inside of tableBuilder.util.

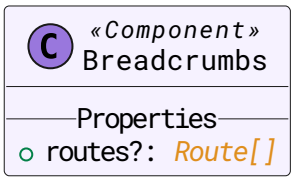
data: `T`

The data that will be displayed in the table.

href: `((row: T[number]) => string | undefined) | undefined`

The callback that will be called when the user clicks on the row.

3.13 Breadcrumbs



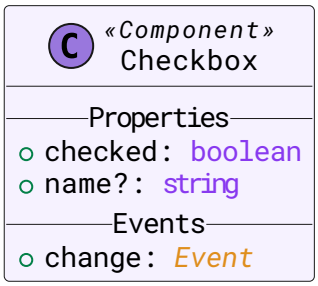
Displays a breadcrumb trail with the given routes.

3.13.1 Props

routes: `Route[] | undefined`

The route that will be displayed in the breadcrumbs.

3.14 Checkbox



A checkbox component. In order to receive changes, the `checked` prop can be reactively bound or the `on:change` event can be listened to for changes.

Internal:

Uses some tailwind css trickery to make the checkbox value to look like a switch. Basically hides the input and sets the focus state via the label. The div is the actual switch and is moved via the peer-checked class where the peer class is set in the input.

3.14.1 Props

checked: `boolean`

Whether the checkbox is checked or not.

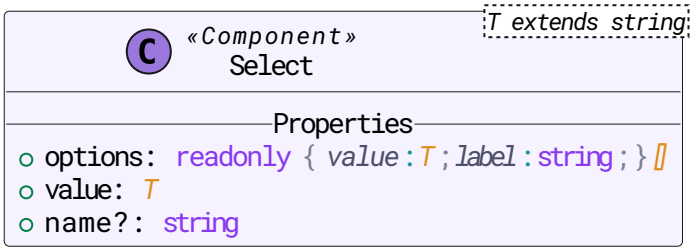
name: `string | undefined`

The form name of this checkbox.

3.14.2 Events

change: `Event`

3.15 Select



A select component that uses the native select element. The options are passed as a prop and the value is bound to the `value` prop. The options prop should be an `as const` array of objects with a value and a label property to allow full type safety.

3.15.1 Props

Generics: `T extends string`

options: `readonly { value: T; label: string; }[]`

The options of the select. The value is the value of the option and the label is the label of the option.

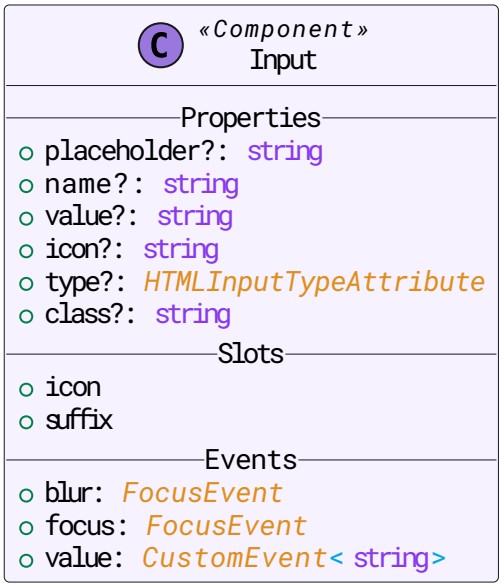
value: `T`

The value that is currently selected. Because of the template variable, full type safety should be enforced if using `const`s as options.

name: `string | undefined`

Name of this `select` element. Used for forms.

3.16 Input



The default input component. A prefix icon can be added inside of the `icon` slot, and/or a suffix icon in the `suffix` slot.

In order to use this component in forms, the `name` prop should be set.

You can also set the value prop, if you want to set an initial value. Or bind to it if you want to use this outside of a form.

3.16.1 Props

placeholder: `string | undefined`

Placeholder of the input.

name: `string | undefined`

The name of the input. Used for the label and for form submission.

value: `string | undefined`

The current value of the input.

icon: `string | undefined`

The icon to be displayed inside of the input.

type: `HTMLInputTypeAttribute | undefined`

The type of the input.

class: `string | undefined`

Additional classes to be applied.

3.16.2 Slots

icon

suffix

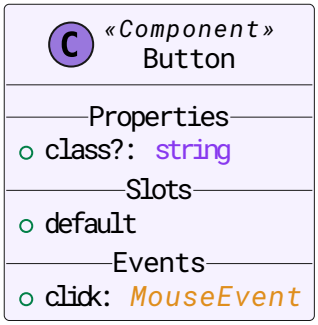
3.16.3 Events

blur: `FocusEvent`

focus: `FocusEvent`

value: `CustomEvent<string>`

3.17 Button



A button with a slot for content.

3.17.1 Props

class: `string | undefined`

Additional classes to be applied to this component.

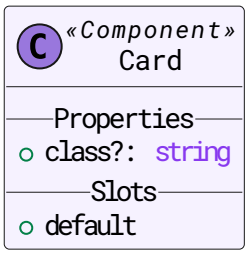
3.17.2 Slots

default

3.17.3 Events

click: `MouseEvent`

3.18 Card



A card with a slot for content. Sets the default padding and border radius. You can override this by passing your own classes.

3.18.1 Props

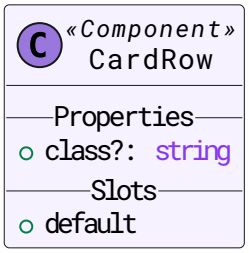
class: `string | undefined`

Additional classes to be applied to this component.

3.18.2 Slots

default

3.19 CardRow



This component should be used to display rows inside of a Card.

It's recommended to only use up to two children in the slot, which will be displayed at both ends of the row.

3.19.1 Props

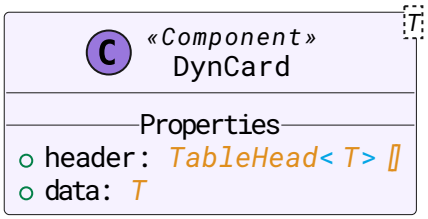
class: `string | undefined`

Additional classes to be applied to this component.

3.19.2 Slots

default

3.20 DynCard



Uses Card, CardRow.

A card that displays the data of the given header.

This works dynamically similar to the DynTable component. So you should probably use the createTableHeadGenerator util function inside of tableBuilder.util to create the header.

3.20.1 Props

Generics: `T`

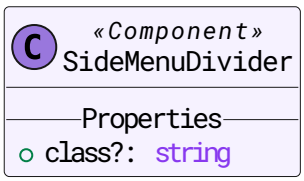
header: `TableHead<T>[]`

The header of the table. Also includes the icon and the href.

data: `T`

The data that will be displayed in the table.

3.21 SideMenuDivider

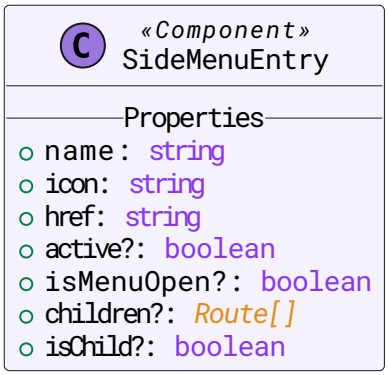


A divider for the side menu.

3.21.1 Props

class: `string | undefined`

3.22 SideMenuEntry



Uses SideMenuDivider.

The side menu entry component.

It can be opened by clicking on it when the parent side menu is open.

When open, all the children will be displayed as subentries using this component.

3.22.1 Props

name: `string`

The name to be displayed in this side menu entry.

icon: `string`

The icon to be displayed in this side menu entry.

href: `string`

The href to be used in this side menu entry.

This is the URL of the page this entry links to.

active: `boolean | undefined`

Whether this side menu entry is active or not.

Active entries are highlighted visually.

isMenuOpen: `boolean | undefined`

Whether the parent side menu is open or not.

children: `Route[] | undefined`

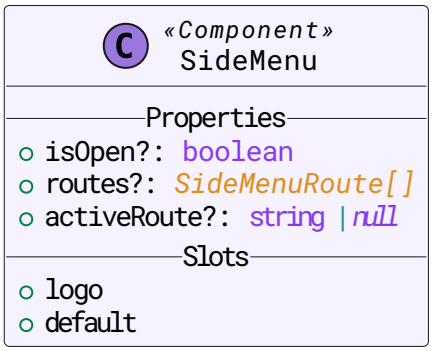
The children of this side menu entry.

Will be displayed as subentries.

isChild: `boolean | undefined`

Whether this side menu entry is a child of another SideMenuEntry, meaning it is a subentry.

3.23 SideMenu



Uses SideMenuDivider, SideMenuEntry.

The side menu component. It contains the SideMenuEntry and SideMenuDivider components.

You can override the default SideMenuEntry list display by using the default slot.

You can also override the logo by using the `logo` slot.

Internal:

When setting a logo, do not forget to set the global `FADE_OPTIONS` constant, otherwise it may look weird.

3.23.1 Props

isOpen: `boolean | undefined`

The current state of the side menu.

Can be bound in order to change the state from other components.

routes: `SideMenuRoute[] | undefined`

The routes to be displayed in the side menu.

activeRoute: `string | null | undefined`

The current route that is active.

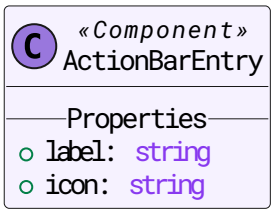
Should usually be the current URL provided by SvelteKit (`$page.url.href`).

3.23.2 Slots

logo

default

3.24 ActionBarEntry



Represents one of the entries of the ActionBar.

3.24.1 Props

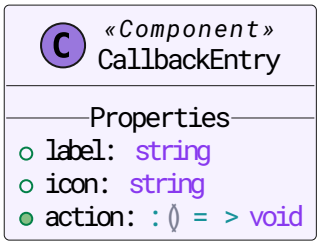
label: `string`

The label of this ActionBar entry.

icon: `string`

The icon of this ActionBar entry.

3.25 CallbackEntry



Uses ActionBarEntry.

An ActionBarEntry with an `on:click` callback action associated with it.

3.25.1 Props

label: `string`

The label of this ActionBar entry.

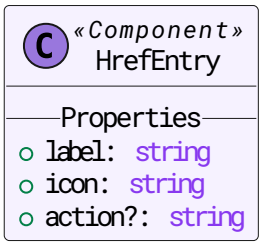
icon: `string`

The icon of this ActionBar entry.

action: `() => void`

Callback function that is executed on click.

3.26 HrefEntry



Uses ActionBarEntry.

An ActionBarEntry that acts as a link to the specified URL.

3.26.1 Props

label: `string`

The label of this ActionBar entry.

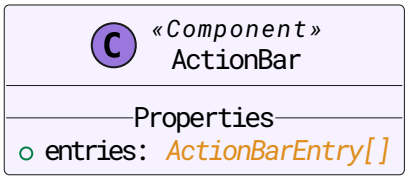
icon: `string`

The icon of this ActionBar entry.

action: `string | undefined`

URL for hyperlink

3.27 ActionBar



Uses CallbackEntry, HrefEntry.

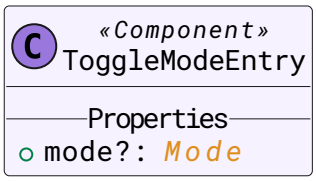
The action bar contains actions that can be performed when in edit mode.

3.27.1 Props

entries: `ActionBarEntry[]`

Actions that are displayed.

3.28 ToggleModeEntry



Uses Checkbox.

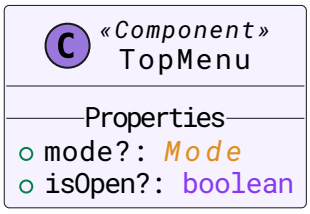
The ActionBar entry responsible for toggling modes.

3.28.1 Props

mode: `Mode | undefined`

The current mode of this Entry.

3.29 TopMenu



Uses Input, ActionBar, ToggleModeEntry.

The top menu component.

The search bar and the ActionBar are located here.

3.29.1 Props

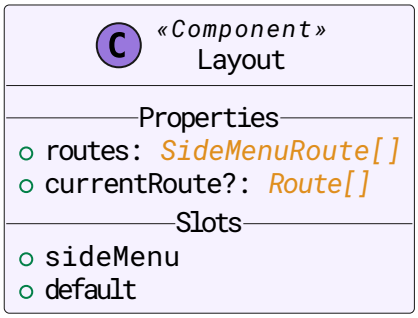
mode: `Mode | undefined`

The mode of the current page. Possible modes are currently "view" and "edit": TODO: maybe extract this to a store?

isOpen: `boolean | undefined`

Whether the side menu is open or not. TODO: probably should search for a better solution for this.

3.30 Layout



Uses SideMenu, TopMenu, Breadcrumbs.

The basic component for the layout of the application.

This Component is intended to be used in Layouts, where the page body will automatically be inserted into the default slot.

You can also override the SideMenu by using the `sideMenu` slot.

3.30.1 Props

routes: `SideMenuRoute[]`

The routes to be displayed in the side menu.

currentRoute: `Route[] | undefined`

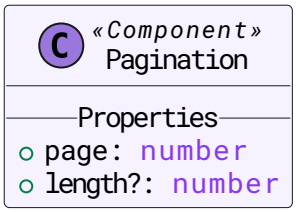
The current route to be displayed in the Breadcrumbs.

3.30.2 Slots

sideMenu

default

3.31 Pagination



A pagination component that allows the user to navigate through pages of a list.

3.31.1 Props

page: `number`

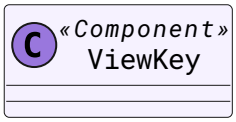
The current page.

length: `number | undefined`

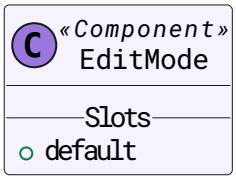
The total number of pages.

3.32 EditKey

3.33 ViewKey



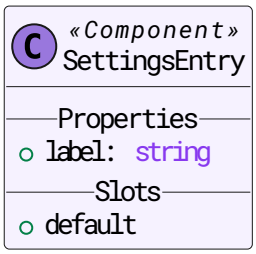
3.34 EditMode



3.34.1 Slots

default

3.35 SettingsEntry



3.35.1 Props

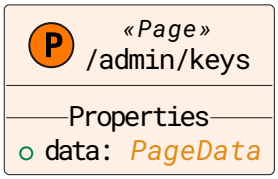
label: `string`

3.35.2 Slots

default

4. Pages

4.1 /admin/keys



Uses DynTable.

4.1.1 Props

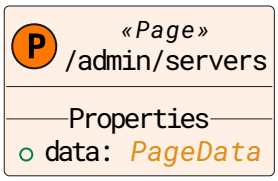
data:

```
{ [x: string]: any; [x: number]: any; [x: symbol]: any; header: ((TableHead<{ AuthKey?: { id?: string | undefined; uuid?: string | undefined; authkey_start?: string | undefined; authkey_end?: string | undefined; ... 6 more ...; last_used?: string | ... 1 more ... | undefined; } | undefined; User?: { ...; } | undefin...
```

4.2 /admin/keys/[id]

Uses EditKey, ViewKey.

4.3 /admin/servers



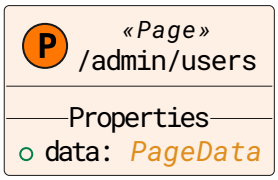
Uses DynTable.

4.3.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; data: { Server?: ({ id?: string | undefined; } & { name?: string | undefined; url?: string | undefined; authkey?: string | undefined; org_id?: string | undefined; ... 20 more ...; cache_timestamp?: boolean | undefined; }) | undefined; Organisation?: ({ ...; } &...`

4.4 /admin/servers/[id]

4.5 /admin/users



Uses DynTable.

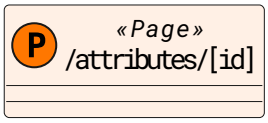
4.5.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; data: { User?: ({ id?: string | undefined; } & { org_id?: string | undefined; server_id?: string | undefined; email?: string | undefined; ... 17 more ...; date_modified?: string | undefined; }) | undefined; Role?: { ...; } | undefined; Organisation?: { ...; } |...`

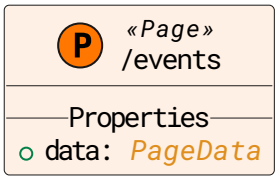
4.6 /admin/users/[id]

4.7 /attributes

4.8 /attributes/[id]



4.9 /events



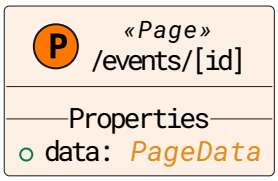
Uses Pagination, DynTable.

4.9.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; header: ((TableHead<{ id?: string | undefined; } & { org_id?: string | undefined; distribution?: "0" | "1" | "2" | "3" | "4" | "5" | undefined; ... 16 more ...; event_creator_email?: string | undefined; } & { ...; }, undefined> & DynTableHeadExtent) | (TableHea...`

Page data

4.10 /events/[id]



Uses Card, DynCard, PillCollection, EditMode.

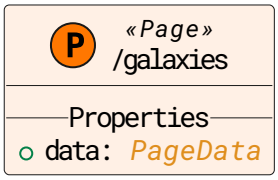
4.10.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; event: ({ id?: string | undefined; } & { org_id?: string | undefined; distribution?: "0" | "1" | "2" | "3" | "4" | "5" | undefined; info?: string | undefined; ... 15 more ...; event_creator_email?: string | undefined; } & { ...; }) | undefined; }`

Page data containing the data of the event with the id in the url

4.11 /events/new

4.12 /galaxies



Uses DynTable.

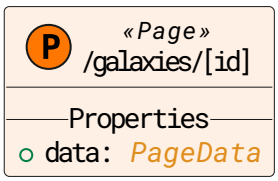
A list of all galaxies.

4.12.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; galaxies: { Galaxy?: { id?: string | undefined; uuid?: string | undefined; name?: string | undefined; type?: string | undefined; description?: string | undefined; version?: string | undefined; icon?: string | undefined; namespace?: string | undefined; kill_chai...`

The data that will be displayed on this page

4.13 /galaxies/[id]

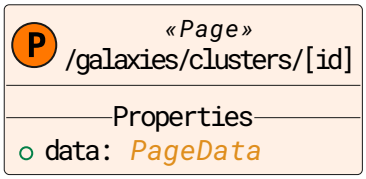


Uses DynCard, DynTable.

4.13.1 Props

data:
`{ [x: string]: any; [x: number]: any; [x: symbol]: any; galaxy: { Galaxy?: { id?: string | undefined; uuid?: string | undefined; name?: string | undefined; type?: string | undefined; description?: string | undefined; version?: string | undefined; icon?: string | undefined; namespace?: string | undefined; kill_chain_...`

4.14 /galaxies/clusters/[id]



Uses Boolean, Card, CardRow, Info, HrefPill, LookupPill, PillCollection, DynTable.

Show all information about a single galaxy cluster, including its elements.

4.14.1 Props

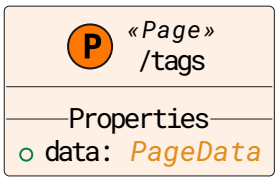
data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; galaxyCluster: { GalaxyCluster?: ({ id?: string | undefined; } & { uuid?: string | undefined; collection_uuid?: string | undefined; type?: string | undefined; ... 17 more ...; GalaxyElement?: { ...; }[] | undefined; } & { ...; }) | undefined; }; tableData: { .....`

Data that is provided +page.ts on page load.

4.15 /settings

Uses Checkbox, Select, SettingsEntry.

4.16 /tags



Uses Pagination, DynTable.

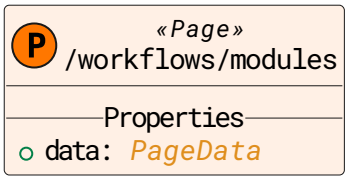
4.16.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; data: { Tag?: ({ id?: string | undefined; } & { name?: string | undefined; colour?: string | undefined; exportable?: boolean | undefined; ... 6 more ...; inherited?: number | undefined; })[] | undefined; }; tableData: ({ id?: string | undefined; } & { name?: st...`

Page data

4.17 /tags/[id]

4.18 /workflows/modules

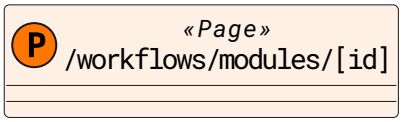


Uses DynTable.

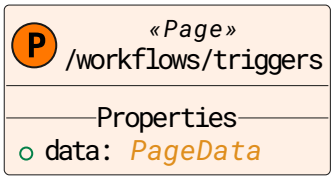
4.18.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; data: Record<string, never> | { AuthKey?: { id?: string | undefined; uuid?: string | undefined; authkey_start?: string | undefined; ... 7 more ...; last_used?: string | ... 1 more ... | undefined; } | undefined; User?: { ...; } | undefined; }[] | ... 41 more .....`

4.19 /workflows/modules/[id]



4.20 /workflows/triggers



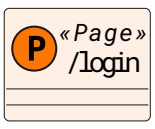
Uses DynTable.

4.20.1 Props

data: `{ [x: string]: any; [x: number]: any; [x: symbol]: any; data: { Galaxy?: { id?: string | undefined; uuid?: string | undefined; name?: string | undefined; type?: string | undefined; description?: string | undefined; version?: string | undefined; icon?: string | undefined; namespace?: string | undefined; kill_chain_or...`

4.21 /workflows/triggers/[id]

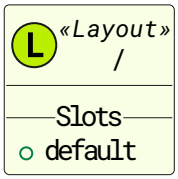
4.22 /login



Uses Button, Input.

5. Layouts

5.1 /

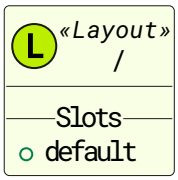


Root layout.

5.1.1 Slots

default

5.2 /



Uses Layout.

5.2.1 Slots

default

6. Error Pages

6.1 /