

Widgets




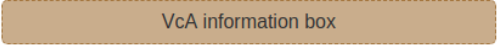

There are several already implemented widgets (see [Technical Documentation](#)). Here you can find all information you need to use the widgets.

Widgets are **complete small software pieces**. Thus, do not try to rework its functionality or user interface design. If you need a different style or functionality try to use the REST-APIs of the corresponding back end systems and implement your own user interface. Furthermore, you can contact the software developers of the widget.

These are the different widget projects, which implement one or more reusable UI elements:

- [vca-widget-base](#)
- [heureka-widget-navigation-2021](#)
- [vca-widget-user](#)

You will find all required technical details for a widget in the readme files of the widget projects. There are data binding and event propagation, but also nesting of UI elements by using slots.

Widget	Project	Image
VcA Frame - The frame describes the base of a Pool ² page. Content is organized in VcA Boxes and these boxes are part of the VcA Columns which are positioned by the VcA Frame. Additionally, it is possible to add a title.	vca-widget-base	
VcA Column - Columns organize the VcA boxes in a vertical order.	vca-widget-base	
VcA Box - Boxes contain the main content of a Pool ² page. They can be used for any kind of content and help you to cluster your information and functions. Please try to develop a small concept how you want to use these boxes, since the usage of boxes will influence the usability of your microservice and thus, of the whole Pool ² .	vca-widget-base	
VcA Info Box - This box will be used to raise the users attention and inform the user about important facts.	vca-widget-base	
VcA Filter Tag - Tags can be used to display key-value structured information. For example, it is used in Drops to show the selected search criteria of the user search (user model attribute and the search key).	vca-widget-base	

Widget

Heureka! menu - The menu implements basic HTML and a bit of CSS code to show the buttons of the main menu. During the initiation of it, the widget requests the content of the menu by an ajax call to Dispenser. That means all buttons with labels and targets, but also the structure of the menu are received from a database managed by Dispenser. Its designed to stay at the top of the page.

Heureka! Footer - The footer contains a menu with secondary priority. It is designed to be the bottom of the page.

Plain crew name - Since microservices should be loosely coupled, ideally they only save references to data objects managed by other microservices. Therefore, you need widgets to print a data object, if you only have a reference. This widget prints a crew name by a given crews `UUID`.

Crew select - It allows the user to select a crew. The widget fires an event after selection that returns the `UUID` of the selected crew.

Tag - Using this widget, you can show user or crew names as small (removable) tags. On click, the user will be redirected to the profile page of the user that is represented by the tag.

Avatar - Shows the avatar (profile image) of a user.

VcA Role - Shows the role of a user as a tag.

Project

heureka-
widget-
navigation-
2021

heureka-
widget-
navigation-
2021

vca-
widget-
user

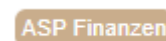
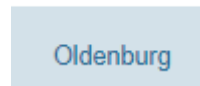
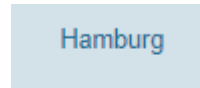
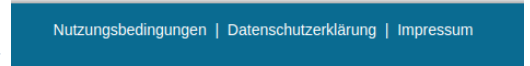
vca-
widget-
user

vca-
widget-
user

vca-
widget-
user

vca-
widget-
user

Image



Widget

Project

Image

User Widget - Visualizes a user in various forms.

vca-
widget-
user



User Widget List - Shows a set of users and implements functions to search through, sort and paginate the set. There are four different forms of visualizing the set: as large user widget, as medium user widget, as small user widget and as table columns.

vca-
widget-
user



User Widget Autocomplete - Allows to select a list of users by an autocomplete field.

vca-
widget-
user



Grav was </> with ❤️ by Trilby Media.