

Inhaltsverzeichnis

Like the dataentry, the values in the columns can also be formatted by postfixing the column names with the typealias. Like the deadline_dt column above. Another example shows website, as an url is given label the entries in cols with e.g. _url to enable linking in output.

The datalist works like the datatable above and accepts the same parameters. But instead of displaying a table, an unordered list is shown. Each result row is one list item, all selected columns are default joined with a space. When sepbyheaders is not set, the header parameter is completely ignored. No pagination is used.

The keyword before the colon is a configuration option and the value behind is the actual setting. To make it more fault tolerant often multiple option names are possible. Here is a list of all available options:

Option(s)	Required?	Description
cols select	yes	These are the attributes you want to display. These are the same names you used in the Data Entry part
head header headers	no	If specified, these names will be used in the table headers instead of the column names
max limit	no	How many rows should be displayed. If more rows are available the table will be made browsable. If not given all matching rows are shown
sort order	no	By what column should the table be sorted initially? Prepend a ^ to reverse the sorting
filter where filterand and	no	Filter by a column value. You may specify this more than once, multiple filters will be ANDed.
filteror or	no	Like filter, but multiple instances will be ORed
dynfilters	no	Set to 1 to enable a row of input fields for dynamically filtering the table
summarize	no	Set to 1 to calculate sum of columns
align	no	List of column alignments. The alignments can be left (l), center (c) or right (r)
rownumbers	no	Set to 1 to show row numbers
widths	no	Specifies column widths e.g. 50px, 20em, -, 10%. Skip column by dash -.
sepbyheaders	no	Set to 1 to use headers as separators between the values. In headers you can use additional spaces, when enclosed by quotes e.g. „ - “

For filtering, multiple comparators are possible:

Comparator	Meaning
=	Exact match
!= or <>	Does not exactly match
<	Less than
≤	Less or equal than
>	Greater than

Comparator	Meaning
\geq	Greater or equal than
\sim	Wildcard match. Use a * as wildcard. Like Apple* to match Apple Pie and Apple Computer; e.g. dessert~ *Pie. Case insensitive.
$*\sim$	Wildcard match. Look for matches containing search term; e.g. dessert*~ Pi match Apple Pie
$!\sim$	Negative Wildcard match. Select everything that does not match the expression.
$\sim\sim$	Looks if contained in given comma separated list; e.g. dessert ~~ diner,dessert

You may use the special variables in a filter to make it match against current logged in user or current date e.g. filter: `username=%user%`

Variable for filtervalue	Meaning
<code>%user%</code>	match against the currently logged in user
<code>%group%</code>	match against the groups of the current user; use the $\sim\sim$ comparator
<code>%now%</code>	compare with the current date
<code>%lang%</code>	Language code of your default wiki language. e.g. en
<code>%trans%</code>	Language code of the localisation of the current page.

Requires the Translation Plugin. For page link `de:test:page`, it is replaced by `de`. Fallback to default lang. Empty without Translation Plugin.]

There are a few variables available to be used as field names:

Variable for fieldname	Meaning
<code>%pageid%</code>	The whole pagename of a page
<code>%title%</code>	The “title” of a page. This is the first headline of a page.
<code>%class%</code>	The class of a data entry

Nichts gefunden