

# 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 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
<code>cols   select</code>	yes	These are the attributes you want to display. These are the same names you used in the Data Entry part
<code>head   header   headers</code>	no	If specified, these names will be used in the table headers instead of the column names
<code>max   limit</code>	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
<code>sort   order</code>	no	By what column should the table be sorted initially? Prepend a ^ to reverse the sorting
<code>filter   where   filterand   and</code>	no	Filter by a column value. You may specify this more than once, multiple filters will be ANDed.
<code>filteror   or</code>	no	Like filter, but multiple instances will be ORed
<code>dynfilters</code>	no	Set to 1 to enable a row of input fields for dynamically filtering the table
<code>summarize</code>	no	Set to 1 to calculate sum of columns
<code>align</code>	no	List of column alignments. The alignments can be left (l), center © or right ®
<code>rownumbers</code>	no	Set to 1 to show row numbers
<code>widths</code>	no	Specifies column widths e.g. 50px, 20em, -, 10%. Skip column by dash -.

For filtering, multiple comparators are possible:

Comparator	Meaning
<code>=</code>	Exact match
<code>!= or &lt;&gt;</code>	Does not exactly match
<code>&lt;</code>	Less than
<code>≤</code>	Less or equal than
<code>&gt;</code>	Greater than
<code>≥</code>	Greater or equal than
<code>~</code>	Wildcard match. Use a * as wildcard. Like <code>Apple*</code> to match <code>Apple Pie</code> and <code>Apple Computer</code> ; e.g. <code>dessert~ *Pie</code> . Case insensitive.
<code>*~</code>	Wildcard match. Look for matches containing search term; e.g. <code>dessert*~ Pi</code> match <code>Apple Pie</code>
<code>!~</code>	Negative Wildcard match. Select everything that does not match the expression.

Comparator	Meaning
~~	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
%user%	match against the currently logged in user
%group%	match against the groups of the current user; use the ~~ comparator
%now%	compare with the current date
%lang%	Language code of your default wiki language. e.g. en
%trans%	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
%pageid%	The whole pagename of a page
%title%	The "title" of a page. This is the first headline of a page.
%class%	The class of a data entry
# ↓ <b>Details</b> <b>Beschreibung</b> <b>Version</b>	
Nichts gefunden	