

Documentation

Documentation for the ConfiDoc Plugin for Confluence



See also [tutorials](#) for easy to follow examples on how to use ConfiDoc in different scenarios

TableView: <i>confidoc-table</i>	The <i>confidoc-table</i>, <i>confidoc-list</i>, and <i>confidoc-card</i> macros are used to present the REST service content (or RSS, CSV, XML) as a table, list (WYSIWYG), or card, each with filtering and data sorting/ordering.			
	Name of the service	serviceId	Services are defined by the Confluence administrator using the plugin's configuration screen	
	Root element	root	Apply additional filters to the returned JSON data using field names as inputs	
	ListView: <i>confidoc-list</i>	Sorting /Ordering	sort	See examples below
		Filter	filter	Apply a filter on returned data; see examples below
	CardView: <i>confidoc-card</i>	Cache resources locally	useCache	Request results can be stored locally in the Confluence cache in order to improve page loading times and to reduce the number of requests to external servers; default local cache duration is 30 minutes
		Show service configuration dialog	enableLookup	Users will be able to configure their own parameters for a configured service request
		Define configuration parameters	lookupLabels	Define labels for the configuration dialog using key-value pairs, separated by "&" (as in URL notation), where <u>key</u> is the parameter name, and <u>value</u> is the text to be used as the label
		Caption for configuration button	lookupBtnLabel	Provide a helpful pointer under the configuration dialog submit button
Set HTML element ID		elementId	Useful, when you want to have a predictable and constant element id in HTML for generated content. Important: "_viewport" will be added to the name	

ConfiDoc Field <i>confidoc-field</i>	This macro defines a field to display returned data within the ConfiDoc TableView or ListView macros.	
Field name	name	The name of the field in the returned data to be displayed by the TableView or ListView macros
Field label	label	The label for the field to be displayed inside the table or list
Field type	type	Choose the field type from these supported formats: <ul style="list-style-type: none"> • Link • Date • Number • Image • 'as-is' (blank: the value will be shown without transformation)
Input format	inputFormat	Define the format in a returned data field in order to reformat using Output format (below), for example, to transform returned date values into more readable form
Output format	outputFormat	Define the transformation rules for Output to be applied to data in the Input format (above)
CSS	css	Define CSS rules to be applied on returned data in a rendered field
Access control	restrictions	Control access to this field with a comma-separated list of group- or user names; leave blank if there are no restrictions
Set HTML element ID	elementId	Useful, when you want to have a predictable and constant element id in HTML for generated content

<p>ConfiDoc Excerpt</p> <p><i>confidoc-excerpt</i></p>	<p>Use this macro to include in the Confluence page the HTML or binary content (such as images) from an external webserver</p> <table border="1"> <tr> <td data-bbox="289 180 508 258">Name of the service</td> <td data-bbox="508 180 618 258">serviceld</td> <td data-bbox="618 180 1484 258">Services are defined by Confluence administrators with the ConfiDoc plugin configuration</td> </tr> <tr> <td data-bbox="289 258 508 422">Selector</td> <td data-bbox="508 258 618 422">root</td> <td data-bbox="618 258 1484 422"> <p>A CSS-like selector, but limited to a form: element attribute="value".</p> <p>To select elements by name, give the element's name; for example: div will select all the divs in the page</p> <p>To select all div elements in the page with the <u>myclass</u> class name, use: div class="myclass"</p> </td> </tr> <tr> <td data-bbox="289 422 508 606">Download and cache resources locally</td> <td data-bbox="508 422 618 606">downloadResources</td> <td data-bbox="618 422 1484 606"> <p>You can:</p> <ul style="list-style-type: none"> • leave links to external resources "as is" • download external resources (such as images) to the Confluence server and adds them as attachments to the page • proxy external resources though local proxy </td> </tr> <tr> <td data-bbox="289 606 508 684">Cache resources locally</td> <td data-bbox="508 606 618 684">useCache</td> <td data-bbox="618 606 1484 684">Store requests locally in the Confluence cache to improve page-loading times and to reduce the number of requests to external servers (default cache duration is 30 minutes)</td> </tr> <tr> <td data-bbox="289 684 508 747">Content Type</td> <td data-bbox="508 684 618 747">contentType</td> <td data-bbox="618 684 1484 747">Force ConfiDoc to use the given content type</td> </tr> <tr> <td data-bbox="289 747 508 825">Custom CSS styles</td> <td data-bbox="508 747 618 825">css</td> <td data-bbox="618 747 1484 825">Custom CSS rules to apply on the rendered element</td> </tr> <tr> <td data-bbox="289 825 508 1066">Include as</td> <td data-bbox="508 825 618 1066">type</td> <td data-bbox="618 825 1484 1066"> <p>This option provides a way to control how the macro outputs the returned data. Available types include:</p> <ul style="list-style-type: none"> • Link • Date • Number • Image • 'as-is' (blank: the value will be shown without transformation) </td> </tr> <tr> <td data-bbox="289 1066 508 1165">Show service configuration dialog</td> <td data-bbox="508 1066 618 1165">enableLookup</td> <td data-bbox="618 1066 1484 1165">Users will be able to configure their own parameters for a configured service request</td> </tr> <tr> <td data-bbox="289 1165 508 1264">Define configuration parameters</td> <td data-bbox="508 1165 618 1264">lookupLabels</td> <td data-bbox="618 1165 1484 1264">Define labels for the configuration dialog using key-value pairs, separated by "&" (as in URL notation), where <u>key</u> is the parameter name, and <u>value</u> is the text to be used as the label</td> </tr> <tr> <td data-bbox="289 1264 508 1365">Caption for configuration button</td> <td data-bbox="508 1264 618 1365">lookupButtonLabel</td> <td data-bbox="618 1264 1484 1365">Provide a helpful pointer under the configuration dialog submit button</td> </tr> <tr> <td data-bbox="289 1365 508 1442">Caption</td> <td data-bbox="508 1365 618 1442">downloadButtonLabel</td> <td data-bbox="618 1365 1484 1442">When the content to be loaded is binary, a download link will be rendered; this parameter sets the label on that link</td> </tr> <tr> <td data-bbox="289 1442 508 1518">Set HTML element ID</td> <td data-bbox="508 1442 618 1518">elementId</td> <td data-bbox="618 1442 1484 1518">Useful, when you want to have a predictable and constant element id in HTML for generated content. Important: "_viewport" will be added to the name</td> </tr> </table>	Name of the service	serviceld	Services are defined by Confluence administrators with the ConfiDoc plugin configuration	Selector	root	<p>A CSS-like selector, but limited to a form: element attribute="value".</p> <p>To select elements by name, give the element's name; for example: div will select all the divs in the page</p> <p>To select all div elements in the page with the <u>myclass</u> class name, use: div class="myclass"</p>	Download and cache resources locally	downloadResources	<p>You can:</p> <ul style="list-style-type: none"> • leave links to external resources "as is" • download external resources (such as images) to the Confluence server and adds them as attachments to the page • proxy external resources though local proxy 	Cache resources locally	useCache	Store requests locally in the Confluence cache to improve page-loading times and to reduce the number of requests to external servers (default cache duration is 30 minutes)	Content Type	contentType	Force ConfiDoc to use the given content type	Custom CSS styles	css	Custom CSS rules to apply on the rendered element	Include as	type	<p>This option provides a way to control how the macro outputs the returned data. Available types include:</p> <ul style="list-style-type: none"> • Link • Date • Number • Image • 'as-is' (blank: the value will be shown without transformation) 	Show service configuration dialog	enableLookup	Users will be able to configure their own parameters for a configured service request	Define configuration parameters	lookupLabels	Define labels for the configuration dialog using key-value pairs, separated by "&" (as in URL notation), where <u>key</u> is the parameter name, and <u>value</u> is the text to be used as the label	Caption for configuration button	lookupButtonLabel	Provide a helpful pointer under the configuration dialog submit button	Caption	downloadButtonLabel	When the content to be loaded is binary, a download link will be rendered; this parameter sets the label on that link	Set HTML element ID	elementId	Useful, when you want to have a predictable and constant element id in HTML for generated content. Important: "_viewport" will be added to the name
Name of the service	serviceld	Services are defined by Confluence administrators with the ConfiDoc plugin configuration																																			
Selector	root	<p>A CSS-like selector, but limited to a form: element attribute="value".</p> <p>To select elements by name, give the element's name; for example: div will select all the divs in the page</p> <p>To select all div elements in the page with the <u>myclass</u> class name, use: div class="myclass"</p>																																			
Download and cache resources locally	downloadResources	<p>You can:</p> <ul style="list-style-type: none"> • leave links to external resources "as is" • download external resources (such as images) to the Confluence server and adds them as attachments to the page • proxy external resources though local proxy 																																			
Cache resources locally	useCache	Store requests locally in the Confluence cache to improve page-loading times and to reduce the number of requests to external servers (default cache duration is 30 minutes)																																			
Content Type	contentType	Force ConfiDoc to use the given content type																																			
Custom CSS styles	css	Custom CSS rules to apply on the rendered element																																			
Include as	type	<p>This option provides a way to control how the macro outputs the returned data. Available types include:</p> <ul style="list-style-type: none"> • Link • Date • Number • Image • 'as-is' (blank: the value will be shown without transformation) 																																			
Show service configuration dialog	enableLookup	Users will be able to configure their own parameters for a configured service request																																			
Define configuration parameters	lookupLabels	Define labels for the configuration dialog using key-value pairs, separated by "&" (as in URL notation), where <u>key</u> is the parameter name, and <u>value</u> is the text to be used as the label																																			
Caption for configuration button	lookupButtonLabel	Provide a helpful pointer under the configuration dialog submit button																																			
Caption	downloadButtonLabel	When the content to be loaded is binary, a download link will be rendered; this parameter sets the label on that link																																			
Set HTML element ID	elementId	Useful, when you want to have a predictable and constant element id in HTML for generated content. Important: "_viewport" will be added to the name																																			
<p>ConfiDoc Excerpt Anchor</p> <p><i>confidoc-excerpt-anchor</i></p>	<p>This macro helps you to group content elements inside a block and easily reference this block later with the ConfiDoc Excerpt macro</p> <table border="1"> <tr> <td data-bbox="289 1575 638 1652">Alphanumeric AnchorID (no spaces)</td> <td data-bbox="638 1575 727 1652">anchorId</td> <td data-bbox="727 1575 1484 1652">Reference this block from the ConfiDoc Excerpt macro using the "div id="AnchorID"" selector</td> </tr> </table>	Alphanumeric AnchorID (no spaces)	anchorId	Reference this block from the ConfiDoc Excerpt macro using the "div id="AnchorID"" selector																																	
Alphanumeric AnchorID (no spaces)	anchorId	Reference this block from the ConfiDoc Excerpt macro using the "div id="AnchorID"" selector																																			
<p>ConfiDoc Show-For (IF) Macro</p> <p><i>confidoc-show-for</i></p>	<p>This macro displays included content only for given users and/or groups</p> <table border="1"> <tr> <td data-bbox="289 1759 565 1812">Content is visible only to</td> <td data-bbox="565 1759 686 1812">restrictions</td> <td data-bbox="686 1759 1230 1812">A comma-separated list of group names and/or user names</td> </tr> </table>	Content is visible only to	restrictions	A comma-separated list of group names and/or user names																																	
Content is visible only to	restrictions	A comma-separated list of group names and/or user names																																			

ConfiDoc Globals <i>confidoc- globals</i>	This macro provides a convenient way to store and display global variables in Confluence		
	Variable name	vars	Variable name
CSS Rules for ConfiDoc Fields <i>confidoc-field- css</i>	This is a helper macro to format fields visually based on their values		
	Field value or expression	con dition	Use values or expressions in order to match returned data much as you would for filters; ConfiDoc filters are very similar in functionality to ConfiForms Filters (also developed by Vertuna LLC)
	Field name	field Na me	Define the name of the field to which this rule should be applied (if blank, then the rule is applied to the row)
	CSS to be applied	css	CSS to apply when the above condition is met

Using filters

ConfiDoc filters employ syntax similar to the queries in Apache Lucene for expressions, which supports grouping using brackets (currently, ConfiDoc filters support only outer or single bracket sets, not inner or sub-brackets). Filters support matching for expressions that start with a wild-card, and filtering per field as well as for free text search (seeking a match against any field value).

The operators AND and OR are case-sensitive!

Filters support negative results with "not notation" to filter records that do not match the given filter with the '!'. For example: `!f1:[empty]` will search for records where the f1 field value is not empty (has some value).

Filters can perform simple arithmetic operations on dates (for example, to add or subtract days from today's date). Here are some examples:

To match records where f1 contains 'success' and f2 is after yesterday or where f3 equals 'accepted':

`(f1:*success AND f2:>[yesterday]) OR (f3:accepted)`

To match records where f1 contains 'success' and where f2 is after yesterday or where f3 is not empty:

`(f1:*success AND f2:>[today]-1) OR (!f3:[empty]) -`

This filter would be invalid because ConfiDoc does not yet support inner brackets in expressions:

`(f1:*success AND (f2:>[yesterday] OR (f3:accepted)))`.

Reserved words to use in expressions:

[empty]	To match when a particular field in the returned data is empty; for example: field1:[empty] will match records where field1 is empty (does not have a value even though the form defines this field)
[now]	To match on the current time and date; often useed with '<' and '>' for comparing with dates stored. For example: someDateField:<[now] will match records where the field 'someDateField' has a value which is in the past compared to now (current time)
[today]	To match on the current date, much like [now], but without hours and minutes
[tomorrow]	To compare the returned data against tomorrow's date ([today]+1 could also be used)
[yesterday]	To compare the returned data against yesterday's date ([today]-1 could be also used)
[end-of-month], [start-of-month], [now], [today], [yesterday], [tomorrow], [today]+N, [today]-N	Functions added / revised in ConfiDoc version 3.4.6
'<' and '>'	These can be used together with date and datetime fields, as well as to compare values in the returned data with values stored in numeric fields
'=<' and '>='	Added in ConfiDoc 3.4.6
!	To reverse or negate the filter condition; for example: !field1:[empty] will find records that have data in 'field1'

Additional examples:

- assuming field1 is of type date (or datetime), this filter will return records where the returned values of field1 are not older than 5 days before now:
field1:[today]-5
- assuming field1 is of type date (or datetime), this filter will return records where the returned values of field1 is not after 10 days from now
field1:[today]+10

Using sorting and limiting the number of records returned

ConfiDocs supports sorting results in ascending (ASC) and descending (DESC) directions. The syntax is similar to that in SQL, and the ordering rules must be coma separated.

Order directions ASC and DESC are case-sensitive!

Some examples: to limit the number of records returned, a LIMIT operator is used; for example,

f1 ASC, f2 DESC

will sort the dataset by the field f1 in ascending order and by f2 in descending order.

f1 ASC, f2 DESC LIMIT 2

will sort the dataset by the field f1 in ascending order and by f2 in descending order, and will limit the number of records returned to 2.

LIMIT 5

will limit the number of records returned to 5.