Managing Free Text Filters

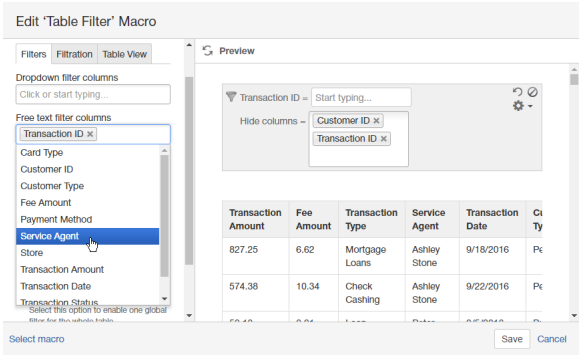
This version of the app's documentation is outdated. Please find the information you're looking for here:

- Filter types available

Managing Free Text Filters
Table Filter for Confluence add-on is equipped with the free text filters. You can add this filter type for any column of your table. It allows you to enter text queries for filtration of data in the selected column of your table. You can also enter the regular expressions for flexible data filtration in the column.

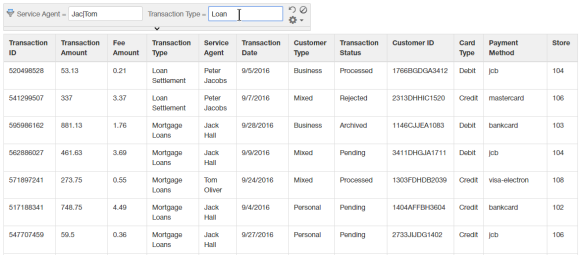
Adding the Free Text Filter

1. Edit the page.
2. Insert the Table Filter macro and paste the table within the macro body.
3. Select the macro and click **Edit**.
4. In the **Free text filter columns** box, select the columns for filtration with free text filters.
5. Save the macro.
6. Save the page.



Filtering the table with free text filters

1. Open the page with the table for filtration.
2. On the filtration pane, locate the free text filters and position the mouse pointer within the appropriate one.
3. Enter the appropriate query or regular expression for filtration.



> You can also filter tables with @-user mentions by the current user (write or select **@currentUser**).

Using regular expressions

You can use JavaScript-style regular expressions in free text filters.

Regular Expression	Matched Values
[Dd]oe	Doe, doe
colo(u)?r	color, colour
Developer|Scientist	Developer, Scientist

Table Filter macro uses the **OR** operator for table filtration. You can use the **AND** operator in the free text and global filters. Enter the '**&**' ('ampersand') between values to match two or more values in the cell at once.

The full list of regular expressions is available here.

Free text filters allow you to use regular expressions while filtering table data. They allow you to write powerful filtration queries.

Quick Recipes with Regular Expressions

<table>
<tr><th>Regular Expression</th><th>Explanation</th><th>Example</th></tr>
<tr><td>\d\.\d</td><td>This query will look for two digits separated by the point (for example, '8.1', '9.5'). Entries with more than one digit after the point will be also included.</td><td>If applying the '\d\.\d' query to the Operating System column, you will get three values with 'OSx 5.7', 'cOS 7.2' and 'aOS 2.8' after filtration.
<table><tr><th>User</th><th>Operating System</th></tr><tr><td>Mike</td><td>OSx 5.7</td></tr><tr><td>Jane</td><td>SuperOS</td></tr><tr><td>Stewart</td><td>OSx 9</td></tr><tr><td>Jane</td><td>cOS 7.2</td></tr><tr><td>Andy</td><td>aOS 2.8</td></tr></table></td></tr>
<tr><td>\d\d\:\d\d\:</td><td>You can use this query if you want to filter dates having time and dates without time.</td><td>If applying the '\d\d\:\d\d\:' query to the Sync Time column, you will get three values with '11/17/2015 10:35:58', '11/17/2015 12:24:54' and '11/19/2015 16:47:22' after filtration.
<table><tr><th>User</th><th>Sync Time</th></tr><tr><td>Mike</td><td>11/17/2015 10:35:58</td></tr><tr><td>Jane</td><td>11/28/2015</td></tr><tr><td>Stewart</td><td>11/17/2015 12:24:54</td></tr><tr><td>Jane</td><td>11/30/2015</td></tr><tr><td>Molly</td><td>11/16/2015 17:28</td></tr><tr><td>Andy</td><td>11/19/2015 16:47:22</td></tr></table></td></tr>
<tr><td>^[0-9]{2}$

<b>^[0-9]{3}$</td><td>You can use these queries if you want to filter people by age depending on the number of digits.</td><td>If applying the '^[0-9]{2}$' query to the Age column, you will get three numbers comprised of two digits.
<table><tr><th>User</th><th>Age</th></tr><tr><td>Mike</td><td>120</td></tr><tr><td>Jane</td><td>75</td></tr><tr><td>Stewart</td><td>57</td></tr><tr><td>Jane</td><td>111</td></tr><tr><td>Molly</td><td>85</td></tr></table></td></tr>
<tr><td>\b([1-1][1-1])

\b([1-1][2-2])

\b([1-1][3-3])

and so on</td><td>You can use this query if you have a list with timestamps and you would like to see how many operations were performed at a particular hour,</td><td>If applying the '\b([1-1][1-1])' query to the Login time column, you will get three entries of login attempts.
<table><tr><th>User</th><th>Login Time</th></tr><tr><td>Mike</td><td>11:30:25</td></tr><tr><td>Jane</td><td>13:31:25</td></tr><tr><td>Stewart</td><td>11:32:25</td></tr><tr><td>Jane</td><td>12:40:25</td></tr><tr><td>Molly</td><td>11:47:25</td></tr><tr><td>Stewart</td><td>12:20:25</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>h(n|a)s</td>
<td>You can use this query if you want to filter last names having either 'n' or 'a' between h and s.</td>
<td>If applying the 'h(n|a)s' query to the Last Name column, you will get four entries (Johns, Johnston, Johnson and Johason).
<table>
<tr><th>First Name</th><th>Last Name</th></tr>
<tr><td>Alexander</td><td>Johns</td></tr>
<tr><td>John</td><td>Johnston</td></tr>
<tr><td>Mike</td><td>Johney</td></tr>
<tr><td>Alice</td><td>Johnson</td></tr>
<tr><td>Pavel</td><td>Johason</td></tr>
<tr><td>Jane</td><td>Johannesen</td></tr>
<tr><td>Jannet</td><td>Johndrow</td></tr>
<tr><td>Paul</td><td>Johnting</td></tr>
</table>
</td>
</tr>
<tr>
<td>Ruby & Python</td>
<td>You can use '&' (ampersand) to filter cells consisting two or more values standing in different positions in the cells.

This is a custom enhancement in the add-on that simplifies the usage of the regular expression to filter cells containing two or more values at a time.</td>
<td>If applying the 'Ruby & Python' query to the Skills column, you will get two names of employees who have these skills.
<table>
<tr><th>User</th><th>Skills</th></tr>
<tr><td>Mike</td><td>Java, JavaScript, HTML5, CSS, ASP, Agile</td></tr>
<tr><td>Jane</td><td>Java, PHP, CSS, HTML5, Agile, Ruby</td></tr>
<tr><td>Stewart</td><td>SOAP, JSON, PHP, CSS, Java, REST, C++</td></tr>
<tr><td>Jane</td><td>Java, Ruby, Python, PHP, CSS, HTML5</td></tr>
<tr><td>Molly</td><td>PHP, CSS, Java, JavaScript, HTML5</td></tr>
<tr><td>Stewart</td><td>Ruby, HTML5, PHP, CSS, JavaScript, Python</td></tr>
</table>
</td>
</tr>
</table>