

TableViewMerger macro to help us to count registrations by user

With TableViewMerger macro you can do data aggregations and merging. We can supply the sources for the data to be supplied from different sources / forms.

There is a number of tutorials available on this topic already

- [Grouping by month and counting - demo for TableViewMerger](#) (TEST)
 - [tableview-merger](#)
- [Another example of TableView Merger usage](#) (ConfiForms)
 - [tableview-merger](#)
 - [advanced](#)
 - [confiforms](#)
- [Simple way to showing your ConfiForms data with TableView macro in Confluence](#) (ConfiForms)
 - [tableview-merger](#)
- [Merging the data from forms with different structure sharing a common key](#) (ConfiForms)
 - [tableview-merger](#)
- [TableViewMerger macro to help us to count registrations by user](#) (ConfiForms)
 - [tableview-merger](#)
- [Grouping by month and counting - demo for TableViewMerger - counting totals](#) (TEST)
 - [tableview-merger](#)
- [Using Table View merger to return unique rows](#) (TEST)
 - [tableview-merger](#)
- [Using TableViewMerger to show fields in one column](#) (TEST)
 - [tableview-merger](#)
- [Using TableViewMerger to show the data with CONCAT function extended](#) (TEST)
 - [tableview-merger](#)
- [Using TableView merger to find duplicate records](#) (ConfiForms)
 - [tableview-merger](#)
- [Advanced aggregations and calculations with TableView Merger](#) (ConfiForms)
 - [tableview-merger](#)
 - [confiforms](#)
 - [advanced](#)

In this cookbook we show how you can quickly group the data by "key" and count how many times the row with this "key" is present in the dataset

Consider to have a simple form with a given structure (2 fields):

The image shows a screenshot of a ConfiForms form definition interface. At the top, there is a header bar labeled 'ConfiForms Form (Definition) | formName = f'. Below this, the main area contains a 'ConfiForms (Form View) Registrations Control' which is currently empty. Underneath the control, there are two form field definitions: 'ConfiForms Form Field (Definition) | fieldName = name | fieldLab...' and 'ConfiForms Form Field (Definition) | fieldName = age | fieldLabe...'. Each field definition has a small 'ConfiForms' logo and a 'definition' label next to it.

```

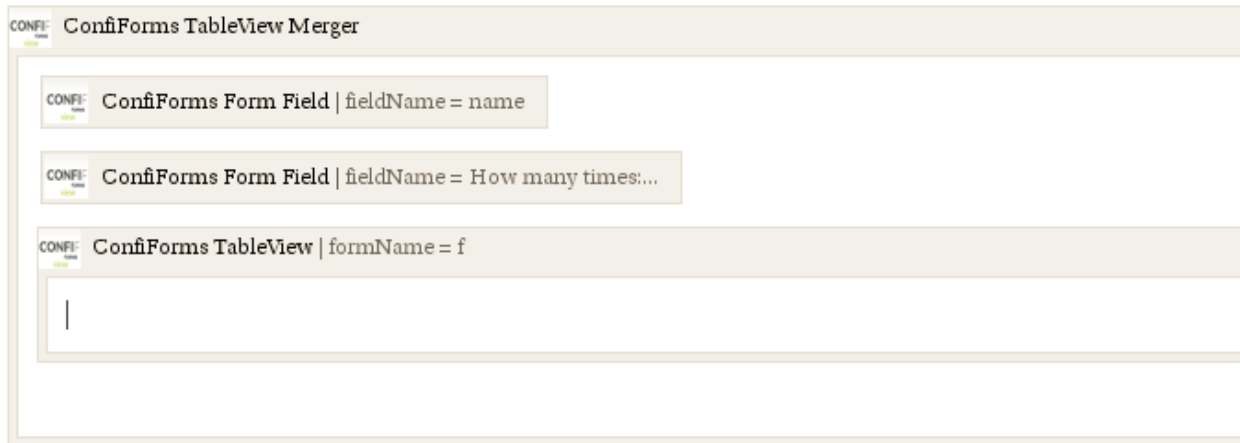
<ac:structured-macro ac:macro-id="f73aa705-b8a4-4cc2-90a6-a41ad564a6a9" ac:name="confiorm" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="7f0d488e-a2a7-4139-87c6-75747865375e" ac:name="confiorm-entry-register" ac:schema-version="1">
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  <p>
    <ac:structured-macro ac:macro-id="c2f4c073-8a47-4827-bdc5-8ff6698602b4" ac:name="confiorm-field-definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">name</ac:parameter>
      <ac:parameter ac:name="fieldLabel">User name</ac:parameter>
      <ac:parameter ac:name="type">text</ac:parameter>
    </ac:structured-macro>
  </p>
  <p>
    <ac:structured-macro ac:macro-id="b2c5bde3-6078-4940-87eb-e08dd75b0d4a" ac:name="confiorm-field-definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">age</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Age</ac:parameter>
      <ac:parameter ac:name="type">number</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>

```

And we need to count how many times in the records the field name has occurred

We will use TableViewMerger to help us with this sort of calculation

And the configuration would be as follows:



Code:

```

<ac:structured-macro ac:macro-id="12e6adff-88dd-44f5-8b56-1267299595f3" ac:name="conform-table-merger" ac:
schema-version="1">
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="accce381-7fc1-44b8-b716-9be2533d2ca6" ac:name="conform-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">name</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="053919d7-7cf9-4f12-9c74-bb054cadca56" ac:name="conform-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">How many times:([count])</ac:parameter>
      </ac:structured-macro>
    </p>
    <ac:structured-macro ac:macro-id="d388ac17-e047-42b5-8055-888ffa427e05" ac:name="conform-table" ac:schema-
version="1">
      <ac:parameter ac:name="formName">f</ac:parameter>
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  </ac:rich-text-body>
</ac:structured-macro>

```

Important bits here are:

- We specify a ConfiForms Field macro with field name set to "name" to point at our "name" fields in the form. This is not an expression (does not have a wrapping () around it's name and does not have a reference to [entry.field_name], so it will be taken as grouping key!)
- For the second column, to count, we define an expression like

```
How many times:([count])
```

It defines a label to be used (How many times) and the expression to calculate ([count]). You can use [Supported math operators, formulas and functions](#) and you can have references to filed values in expressions via [entry.field_name] notations

- [count] and [total] are special context variables available in TableViewMerger. [count] shows the records count for a grouped by key, while [total] holds a total number of records in the dataset

So, for the data like this:

User name	Age
sash	39
vlad	33
sash	32

you will get this calculated with TableViewMerger like this:

User name	How many times
sash	2
vlad	1

You can put the generated table (whole TableViewMerger macro configuration) into a chart macro to build nice charts and graphs over your data. Consult Atlassian documentation on how to use the chart macro: <https://confluence.atlassian.com/doc/chart-macro-163415075.html>

Bonus:

To illustrate how to count registrations by user name, in other words by metafield "createdBy"

You will need to configure your TableViewMerger macro like this

The screenshot shows the configuration interface for the 'ConfiForms TableView Merger' macro. It features a title bar 'ConfiForms TableView Merger' with a 'CONFIGURE' button. Below the title bar, there are three configuration blocks: 1. 'ConfiForms Form Field | fieldName = createdBy' with a 'CONFIGURE' button. 2. 'ConfiForms Form Field | fieldName = How many times:...' with a 'CONFIGURE' button. 3. 'ConfiForms TableView | formName = f1' with a 'CONFIGURE' button. The interface is designed to allow users to configure the macro by adding and configuring form fields and a table view.

What if you want to count the records, but have multiple fields as a grouping key?

The concept is the same. Fields without expressions are taken in to form a unique key for the record. Let's say we want to group the records by createdBy and our "name" field

The configuration for TableViewMerger will be then

The screenshot shows the configuration interface for the 'ConfiForms TableView Merger' macro, similar to the previous one but with an additional field. It features a title bar 'ConfiForms TableView Merger' with a 'CONFIGURE' button. Below the title bar, there are four configuration blocks: 1. 'ConfiForms Form Field | fieldName = createdBy' with a 'CONFIGURE' button. 2. 'ConfiForms Form Field | fieldName = name' with a 'CONFIGURE' button. 3. 'ConfiForms Form Field | fieldName = How many times:...' with a 'CONFIGURE' button. 4. 'ConfiForms TableView | formName = f1' with a 'CONFIGURE' button. The interface is designed to allow users to configure the macro by adding and configuring form fields and a table view.

Sometimes you don't need a field value that forms a grouping key to be present in the resulting table. This is possible by configuring the field not to display

Edit 'ConfiForms Form Field' Macro

Relevant only for dropdown fields (single or multi-select (simple, smart, db)). Filter is applied on the dataset this field is connected to (in case of smart/db fields). Same as in 'Apply filter' rule

☐ Show label on own line
To be used when 'Show value with label' is set to true and you need a label to be placed on own line

☒ Hide if empty (or used as key only)
Useful, when you do not need to show the field when it does not have a value (in ListView/CardView) or in TableView Merger macro when used as a key-only

☐ Do not show description with the field
This is sometimes needed when you

So the resulting table with configuration like this

CONFIFORMS

ConfiForms TableView Merger

CONFIFORMS

ConfiForms Form Field | fieldName = createdBy | hid...

CONFIFORMS

ConfiForms Form Field | fieldName = name

CONFIFORMS

ConfiForms Form Field | fieldName = How many times:...

CONFIFORMS

ConfiForms TableView | formName = fl

will have 2 columns in teh result, not 3 as defined in the configuration (because one is set to be hidden)

User name	How many times
sash	2
vlad	1