

Multiple ways to show your form data in Confluence with ConfiForms



In this tutorial you will learn how to create a simple form and show the data stored in the form in different ways. We call these macros "**views**"

- as WYSIWYG list
- as Card view (card-like table)
- as simple HTML table
- as table with inline search
- as editable table
- as Calendar
- as PlainView (showing data as-is, to mix with HTML and other code)
- as ValueView (shows particular field value)



If you are new to ConfiForms, please take a few minutes to read the [Basic concepts](#) Guide, or go through the video tutorial on that page. 😊

- [WYSIWYG list](#)
- [CardView](#)
- [Simple HTML table](#)
- [Table with instant search](#)
- [Editable table](#)
- [Calendar](#)
- [Show as PlainView, shows values as-is](#)
- [Using ValueView](#)
- [Using TableView with ConfiForms EntryViewer macro](#)

ConfiForms is very flexible at how it can show the data your store. Define form once and show it multiple ways (via ConfiForms Registration Control) and show the data stored with this form on any page, any data subset and in the way and layout you want, with sorting and styling you need.

Let's start with the form. We will define a form with 3 fields: text, textarea and date

We will also have a default "ConfiForms Registration Control" macro which will enable default way of registering records with dialog form.

The screenshot shows the ConfiForms Form configuration interface. At the top, it says "CONF: ConfiForms Form | formName = f". Below this, there are three "CONF: ConfiForms Field Definition" entries, each with a field name and a field label. The first entry has fieldName = t and fieldLabel = ... The second entry has fieldName = ta and fieldLabel = ... The third entry has fieldName = d and fieldLabel = ... Below these, there is a "CONF: ConfiForms Registration Control" entry, which is a macro that can be used to register records with the dialog form.

```

<ac:structured-macro ac:macro-id="4c4be3cf-91c4-4de1-a0b0-ff9f567739cd" ac:name="conform" ac:
schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="cb785dd3-2e33-4626-9759-dc3c5dfbd5a7" ac:name="conform-
field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">t</ac:
parameter>
        <ac:parameter ac:name="fieldLabel">My Text<
/ac:parameter>
        <ac:parameter ac:name="type">text</ac:
parameter>
        </ac:structured-macro>
      </p>
      <p>
        <ac:structured-macro ac:macro-id="59117dc0-60e6-43f8-be55-828ea7664a0c" ac:name="conform-
field-definition" ac:schema-version="1">
          <ac:parameter ac:name="fieldName">ta</ac:
parameter>
          <ac:parameter ac:name="fieldLabel">My
textarea</ac:parameter>
          <ac:parameter ac:name="type">textarea</ac:
parameter>
          </ac:structured-macro>
        </p>
        <p>
          <ac:structured-macro ac:macro-id="c695eala-d03c-4e2a-ac80-85982ca97172" ac:name="conform-
field-definition" ac:schema-version="1">
            <ac:parameter ac:name="fieldName">d</ac:
parameter>
            <ac:parameter ac:name="fieldLabel">My date<
/ac:parameter>
            <ac:parameter ac:name="type">date</ac:
parameter>
            </ac:structured-macro>
          <ac:structured-macro ac:macro-id="f24efal4-a7c6-4738-bf8c-721d5bb75923" ac:name="conform-
entry-register" ac:schema-version="1">
            <ac:parameter ac:name="atlassian-macro-
output-type">INLINE</ac:parameter>
            <ac:rich-text-body>
              <p> </p>
            </ac:rich-text-body>
          </ac:structured-macro>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>

```

Let's add couple of rows of data

Register  Form: f ▼

My Text	My textarea	My date	
another entry	another text	Dec 09, 2015	Edit Delete
test entry	some text for textarea	Dec 08, 2015	Edit Delete

[CSV](#) | [XLS](#) | [JSON](#) | [XML](#) | [RAW](#) | Records: 2 Size: 1.4 KB

exactly 2 rows, like in the screenshot above

Now we will create a page for each of the "views", let's start with **WYSIWYG list** (or the ListView, ConfiForms ListView macro)

WYSIWYG list

CONFI: ConfiForms ListView | formName = f | pageTitle = multiple ways to show the data

CONFI: ConfiForms Field | fieldName = d

CONFI: ConfiForms Field | fieldName = t

 Info

CONFI: ConfiForms Field | fieldName = ta

The details for ListView macro are like this - we reference the form "f" from our page named "multiple ways to show the data"

Edit 'ConfiForms ListView' Macro

List view for Confluence ConfiForms plugin [Documentation](#)

ConfiForms Form name *

f

Page name where the form is defined


multiple ways to show the data

Leave blank if the form is defined on the same page

Filter to apply on entries dataset

See examples in macro preview on the left, under the spoiler

Ordering/Sorting rules to apply on entries dataset

 Preview

Configuration helper for ListViews, TableViews, CardViews, CalendarViews and Registration Control macros

[Hide](#)

The following fields are defined in the form f and can be referenced from here.

- id
- createdBy
- ownedBy
- ownedByName
- createdByName
- created
- dateCreatedFormatted
- t
- ta
- d


Use **ConfiForms Field** macro for that. Available field names you can see above.

[Select macro](#) [Save](#) [Cancel](#)

This is how the page looks in the "view mode"


Dec 08, 2015

test entry

 some text
for textarea

Dec 09, 2015

another entry

 another text

CardView

Also, adding a page to Confluence and placing CardView macro on it

CONFID: ConfiForms CardView | formName = f | pageTitle = multiple ways to show the data | sort = d ASC

CONFID: ConfiForms Field | fieldName = d

CONFID: ConfiForms Field | fieldName = t

CONFID: ConfiForms Field | fieldName = ta

We added sorting here, just for fun and the basics are all the same: referencing a form from our page named "multiple ways to show the data"

ConfiForms Field macros tell us which fields to show and in what order. Important note: for TableViews and CardViews you cannot include the design. The macro body is taken as metadata only, to extract field names and their order appearance

This is how it looks like on the page in "view mode"

My date	Dec 08, 2015
My Text	test entry
My textarea	some text for textarea
My date	Dec 09, 2015
My Text	another entry
My textarea	another text

Simple HTML table

To show the data from ConfiForms forms as table is a responsibility of ConfiForms TableView macro. Configuration is very similar to CardView. List of ConfiForms Field macros tell the order of the fields to include into the resulting table

CONFID: ConfiForms TableView | formName = f | pageTitle = multiple ways to show the data

CONFID: ConfiForms Field | fieldName = d

CONFID: ConfiForms Field | fieldName = t

How the page looks like in the view mode:

My date	Add a Comment (m)	My Text
Dec 08, 2015		test entry
Dec 09, 2015		another entry

In this example we intentionally skipped the Textarea field and included only date and text field. Just to demonstrate the ability to show only the fields you want from a saved dataset.

Table with instant search

Same as above, uses ConfiForms TableView macro, but changes the defaults to enable instant search, and some highlighting

CONFID: ConfiForms TableView | formName = f | pageTitle = multiple ways to show the data | enableSearch = true

CONFID: ConfiForms Field | fieldName = d


CONFID: ConfiForms Field | fieldName = t

CONFID: ConfiForms Field | fieldName = ta

CONFID: CSS Rules for ConfiForms Fields | condition = * | fieldName = ...

We have added CSS rules to show the textarea field values in green colour... and you can change the colour to your own of course 😊

Also , an instant search is enabled via the parameter on TableView (easy to search big tables, including the ones that are paged, shown in pages)

Search 

My date	My Text	My textarea
Dec 08, 2015	test entry	some text for textarea
Dec 09, 2015	another entry	another text

Editable table

Very similar to simple HTML table but will add ConfiForms Edit Controls to enable edits (we can enable edits via dialogs or via inline edit; in this example we will place default "ConfiForms Edit Controls" macro, which will have a delete option enabled as well)



And after saving the page the table will look like

My date	My Text	
Dec 08, 2015	test entry	Edit Delete
Dec 09, 2015	another entry	Edit Delete

with enabled "Edit/Delete" controls for users who have permissions to edit/delete records (form administrators or record owners)

 You can use "ConfiForms Edit Controls" macro with any "views" macro such as TableView, ListView, CalendarView and CardView

Calendar

Another way of showing the data which you can group by date (datetime, timestamp) field is CalendarView

ConfiForms CalendarView macro is responsible for it

Insert 'ConfiForms CalendarView' Macro

Calendar view for Confluence

ConfiForms plugin [Documentation](#)

Name of the field to use as a grouping field *

d

Must be of type Date/Datetime/Datetime Interval

ConfiForms Form name *

f

Page name where the form is defined

multiple ways to show the data

Leave blank if the form is defined on the same page

Filter to apply on entries dataset

Preview

Select macro

Insert

Cancel

Please note that we choose field "d" which is our date field to be the grouping field for the calendar view

CONFI: ConfiForms CalendarView | groupingFieldName = d | formName = f | pageTitle = multiple ways to show the data

CONFI: ConfiForms Field | fieldName = t

Info

CONFI: ConfiForms Field | fieldName = ta

And here is how the calendar view looks like in the view mode

<

>

today

December 2015

month

week

day

test entry

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
		12a test ent	12a another			
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Info

some text for textarea

In this tutorial we have demonstrated few different ways of how you can show the data stored with ConfiForms forms. And this is just "scratching the surface" as there are many many other options and features to help you with creating amazing custom views over your data with ConfiForms!

Show as PlainView, shows values as-is

ConfiForms has a macro to show the data as-is. The macro is called ConfiForms PlainView.

We can mix the contents with HTML tags for example

|

HTML

```
<ul>
```

CONFIF ConfiForms PlainView | formName = f | pageTitle = multiple ways to show the data

```
<li>[entry.t] - [entry.ta] - [entry.d]</li>
```

HTML

```
</ul>
```

This configuration will show a bullet-list of the data we have stored with ConfiForms

Using ValueView

In ConfiForms we have a macro to conveniently show the value of just one field (it can be 1 row or many, depending on a filter you set)

Macro name is ConfiForms ValueView.

Here is how it can be used

Edit 'ConfiForms ValueView' Macro

ConfiForms plugin [Documentation](#)

ConfiForms Form name *

Page name where the form is defined

Leave blank if the form is defined on the same page

Filter to apply on entries dataset

See examples in macro preview on the left, under the spoiler

Field name *

Name of the field defined on the form

[Select macro](#)

You specify a form name, page where form is located, a filter (if you want to show field value for particular record) and a field name

Also, you can apply some CSS if you want to.

Using TableView with ConfiForms EntryViewer macro

Sometimes you want to show a subset of data in some view and then allow users to see the rest of the data. You can do that with ConfiForms EntryViewer macro. It works with TableView, CardView, ListView/CleanView and CalendarView. Also it works with TableViewMerger macro when this macro is used for merging the data from different forms (does not work when TVM is used to render an aggregated view)



EntryViewer can have a default view (card-like view or you can have a custom view, just like you do with ConfiForms Registrations Control / FormView - [ConfiForms Form with custom layout](#))

EntryViewer is rendered as button by default but could be set to render as link if necessary



Need help with ConfiForms or would like to ask for the advice or share a feature request?

Let us know!