

How to use voting controls in ConfiForms



Very simple tutorial to demonstrate how voting control can be used. Please be aware that "voting control" field works only for authenticated users (logged-in Confluence users!)

Quick intro

We have a simple form with 2 fields:

- choice1 - to hold a text for users to vote on (field type: text)
- votes - to count votes (field type: voting)



The demo will work only for logged-in users!!!!

Configuration

The screenshot displays the ConfiForms configuration interface. At the top, a header bar reads "ConfiForms Form Definition | formName = voting". Below this, there are two sub-sections, each with its own header bar: "ConfiForms Form Field Definition | fieldName = choice1 | field..." and "ConfiForms Form Field Definition | fieldName = votes | fieldLa...". The bottom section is titled "ConfiForms (Form View) Registrations Control" and contains a large, empty rectangular area for configuration.

Field type for "votes" is "Voting control"

Edit 'ConfiForms Form Field D

Field for ConfiForms dynamic forms

[Documentation](#)

Field name *

votes

Must be unique within the form and can be alphanumeric (contain only latin symbols and/or numbers, no spaces and umlauts!)

Field label *

Votes

☐ Required *

Field type

Voting control

Rendered as **BUTTON** in view mode!

Caption for quick voting button

Storage format for the form

```

<ac:structured-macro ac:macro-id="61430c46-979f-486a-98f7-33666ecc779e" ac:name="conform" ac:schema-version="1">
  <ac:parameter ac:name="formName">voting</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="8ddfe4e8-085f-485a-a31e-8580a0106e5f" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">choice1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Choice 1</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="516d13d9-3b30-4ba3-967c-902b7c5169f9" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">votes</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Vote!</ac:parameter>
        <ac:parameter ac:name="values">Vote for me!</ac:parameter>
        <ac:parameter ac:name="extras">Remove my vote</ac:parameter>
        <ac:parameter ac:name="type">voting</ac:parameter>
      </ac:structured-macro>
      <ac:structured-macro ac:macro-id="7fb066a8-6d3a-4165-b6c6-32860fb77573" 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>

```

Using a ConfiForms ListView macro to show the data, "choice1" field followed by "votes" field

Votes


ConfiForms ListView | formName = voting


ConfiForms Form Field | fieldName = choice1


ConfiForms Form Field | fieldName = votes

Votes

vote for me

Another option

As the voting field holds users who has voted then we can easily list the users who has voted the way we want

With the help of [Virtual functions](#), and "asUserNames" in particular. So, using "votes.asUserNames" will show the list of usernames who has voted for this particular "choice"

Who has voted



Who has voted

vote for me

sash

Another option

Storage format for the last "ListView"

```

<ac:structured-macro ac:macro-id="b0f059b6-8caa-4846-85ae-30f793032132" ac:name="confiorm-list" ac:schema-
version="1">
  <ac:parameter ac:name="formName">voting</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="71b8430e-27de-4fea-a1d0-856cee1a5728" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">choice1</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="a9069365-764c-4835-acbf-399e91121e55" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">votes</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <strong>
        <ac:structured-macro ac:macro-id="9cb95250-6983-4260-b749-c5c29880aa6c" ac:name="confiorm-field" ac:
schema-version="1">
          <ac:parameter ac:name="fieldName">votes.asUserNames</ac:parameter>
        </ac:structured-macro>
      </strong>
    </p>
    <p>
      <strong>
        <br/>
      </strong>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>

```

As with ANY multi-value field (and voting control is the multi-value holding field, and it holds list of users) you can use "transform" function and for example list the emails of the voted users ([Virtual functions](#))

```
votes.transform(email).asList
```

Here we use transform function on "user" fields and get the "email" property of each user field and put it into the CSV list (with asList function)