Using ConfiForms app as template engine to create pages in Confluence



In this tutorial:

- · how to create a form which will create new pages in Confluence based on the template
- how to set labels to newly created pages (also dynamic labels)
- our pages will have a status field which will show the status of the content on this page (we will have 3 statuses: draft, completed,
- our form will have a link to all created pages
- use formula to have a LastUpdated date field stored for us automatically



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. \bigcirc



- Creating Confluence pages with ConfiForms
- Why ConfiForms creates a broken page from Confluence template
- Why ConfiForms "auto-wraps" everything with ConfiForms ListView on a created page

Creating Confluence pages with ConfiForms

ConfiForms plugin is very versatile in functionality. And one of it's features is to help you with creation of dynamic page templates.

It is not a secret that you can create Confluence pages based on the data entered in ConfiForms, there is a good tutorial about that - Easy employee checklists in Confluence with ConfiForms

This tutorial is focusing on same feature, but from a bit different angle.

We will create pages which are labeled and have a status field, showing the status of the content on this page.

As usual, let's start with a form definition.

We will create a ConfiForms form with 5 fields

- · text field to capture the page title
- raw HTML field to store the link to created page (Hidden HTML field type)
- status field with 3 statuses for page content: draft, completed, approved)
- a field to choose which label should be applied on the page create (a multiselect dropdown field)
- and a formula field to calculate last updated time for us

Here is how the form looks like in the admin view

Template engine

Created by admin, last modified 2 minutes ago

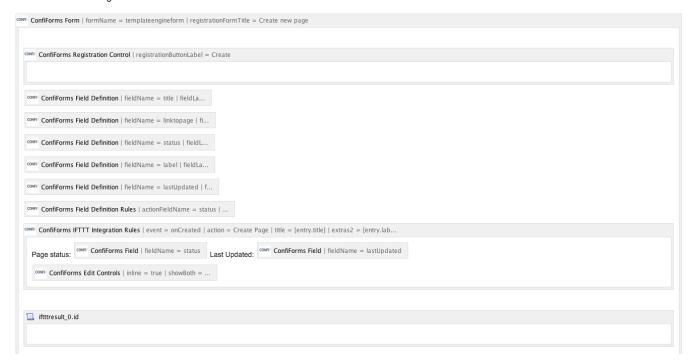


CSV (flatten) | XLS (flatten) | JSON | XML | RAW | Records: 3 Size: 2.6 KB

Two ConfiForms IFTTT macros configured:

- to create a page when new ConfiForms record is created, taking page title from a text field, adding labels to the page
- to update the ConfiForms record when the page is created with link to this page

Here is how the configuration looks like in Confluence editor



(1)

See also, Advanced integrations with IFTTT macros with ConfiForms and Using Confluence templates with ConfiForms and avoid configuration duplication on hints how to mix and use IFTTT with Confluence templates

Here is the complete storage format for the page if you want to import the solution right into your Confluence page (You will need Confluence Source editor, a free plugin from Atlassian to work with raw storage format)

```
<ac:structured-macro ac:macro-id="lb09adaf-ef48-4635-912e-a60b2f7f3e5a" ac:name="confiform" ac:schema-version="</pre>
  <ac:parameter ac:name="formName">templateengineform</ac:parameter>
  <ac:parameter ac:name="registrationFormTitle">Create new page</ac:parameter>
  <ac:rich-text-body>
      <ac:structured-macro ac:macro-id="e80ac6ff-7773-4f53-99e5-4305941266b3" ac:name="confiform-entry-</pre>
register" ac:schema-version="1">
       <ac:parameter ac:name="registrationButtonLabel">Create</ac:parameter>
       <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
       <ac:rich-text-body>

       </ac:rich-text-body>
      </ac:structured-macro>
      <ac:structured-macro ac:macro-id="ec0cef2d-ab06-4947-ab5e-dc8d4898439a" ac:name="confiform-field-
definition" ac:schema-version="1">
       <ac:parameter ac:name="fieldName">title</ac:parameter>
       <ac:parameter ac:name="fieldLabel">Page title</ac:parameter>
       <ac:parameter ac:name="type">text</ac:parameter>
     </ac:structured-macro>
    >
      <ac:structured-macro ac:macro-id="f723479f-fb4c-45d4-99a8-06a143752599" ac:name="confiform-field-</pre>
definition" ac:schema-version="1">
       <ac:parameter ac:name="fieldName">linktopage</ac:parameter>
       <ac:parameter ac:name="fieldLabel">Page</ac:parameter>
       <ac:parameter ac:name="type">hiddenhtml</ac:parameter>
      </ac:structured-macro>
    >
```

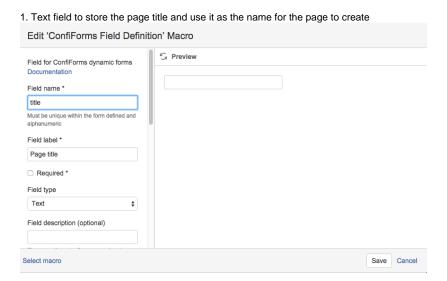
```
<ac:structured-macro ac:macro-id="a96c7f02-cadc-4db3-8fb7-b89b83e022e1" ac:name="confiform-field-
definition" ac:schema-version="1">
       <ac:parameter ac:name="fieldName">status</ac:parameter>
       <ac:parameter ac:name="fieldLabel">Page status</ac:parameter>
       <ac:parameter ac:name="values">false[default=draft|current=completed|success=approved|]</ac:parameter>
       <ac:parameter ac:name="type">status</ac:parameter>
     </ac:structured-macro>
     <ac:structured-macro ac:macro-id="62alaf2e-aede-4135-9c6d-20f7ec6655ca" ac:name="confiform-field-
definition" ac:schema-version="1">
       <ac:parameter ac:name="fieldName">label</ac:parameter>
       <ac:parameter ac:name="fieldLabel">Label</ac:parameter>
       <ac:parameter ac:name="values">false[docs=docs|spec=spec|hr=hr|]</ac:parameter>
       <ac:parameter ac:name="type">multi</ac:parameter>
       <ac:parameter ac:name="separator">,</ac:parameter>
     </ac:structured-macro>
   <q>>
     <ac:structured-macro ac:macro-id="8bc3908a-3246-4863-bf2d-41a5a13c3f05" ac:name="confiform-field-
definition" ac:schema-version="1">
       <ac:parameter ac:name="fieldName">lastUpdated</ac:parameter>
       <ac:parameter ac:name="fieldLabel">Last Updated</ac:parameter>
       <ac:parameter ac:name="extras">FORMATDATETIME(NOW())</ac:parameter>
       <ac:parameter ac:name="type">formula</ac:parameter>
     </ac:structured-macro>
   <ac:structured-macro ac:macro-id="38069eb9-4931-4569-b50c-ed0499f096e0" ac:name="confiform-field-
definition-rules" ac:schema-version="1">
       <ac:parameter ac:name="values">status=default</ac:parameter>
       <ac:parameter ac:name="action">Set value if empty</ac:parameter>
       <ac:parameter ac:name="actionFieldName">status</ac:parameter>
     </ac:structured-macro>
   <ac:structured-macro ac:macro-id="e05996b4-65b3-422e-999e-c25c46a8f25f" ac:name="confiform-ifttt" ac:schema-
version="1">
     <ac:parameter ac:name="action">Create Page</ac:parameter>
     <ac:parameter ac:name="event">onCreated</ac:parameter>
     <ac:parameter ac:name="title">[entry.title]</ac:parameter>
     <ac:parameter ac:name="extras2">[entry.label]</ac:parameter>
     <ac:rich-text-body>
       Page status: <ac:structured-macro ac:macro-id="662a6841-0c6c-48a1-a62a-affafc4a2512" ac:name="</p>
confiform-field" ac:schema-version="1">
           <ac:parameter ac:name="fieldName">status</ac:parameter>
         </ac:structured-macro> Last Updated: <span style="color: rgb(153,153,153);">
           <ac:structured-macro ac:macro-id="e0aebaa9-0d19-42ad-b3a9-22a36e777c89" ac:name="confiform-field"</pre>
ac:schema-version="1">
             <ac:parameter ac:name="fieldName">lastUpdated</ac:parameter>
           </ac:structured-macro>
         </span>
       >
         <ac:structured-macro ac:macro-id="3dfe2568-e61f-4d8b-80c7-2181b06859eb" ac:name="confiform-entry-
edit" ac:schema-version="1">
           <ac:parameter ac:name="inline">true</ac:parameter>
           <ac:parameter ac:name="showBoth">false</ac:parameter>
         </ac:structured-macro>

     </ac:rich-text-body>
   </ac:structured-macro>
   <ac:structured-macro ac:macro-id="e58f83c4-99eb-4642-ab80-90176ba8104a" ac:name="confiform-ifttt" ac:schema-
version="1">
     <ac:parameter ac:name="action">Create ConfiForms Entry</ac:parameter>
     <ac:parameter ac:name="event">onCreated</ac:parameter>
      <ac:parameter ac:name="title">entryId=[entry.id]&amp;linktopage=&lt;a href="http://localhost:1990
/confluence/pages/viewpage.action?pageId=${iftttResult_0.id}">[entry.title]</a&gt;</ac:parameter>
     <ac:parameter ac:name="who">templateengineform:2588674</ac:parameter>
     <ac:rich-text-body>
```

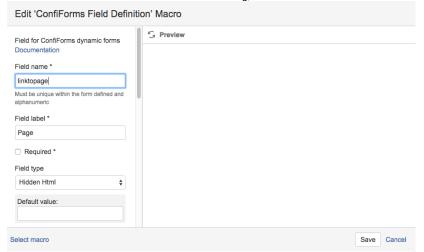
```
</ac:rich-text-body>
  </ac:structured-macro>
  </ac:rich-text-body>
</ac:structured-macro>
```

So, let's walk through each macro on the page

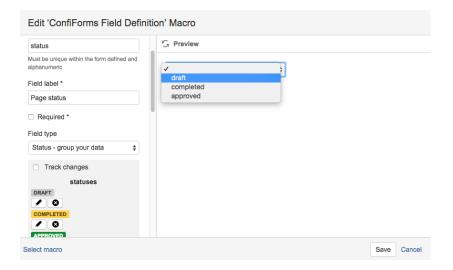
- The ConfiForms Form is used as a container as alway, and in our case here uses standard settings. We have only set the form title to be "Create new page" and named our form as "templateengineform"
- Next is ConfiForms Registration Control, to enable "Register/Add New" button. We place it inside the form container and have it with default settings, but as usual you can set permissions who can use this button, restrict it to be visible for certain groups or users
- Next 5 macros are the field definitions



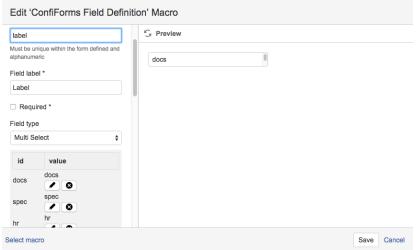
2. Hidden HTML field which is not shown on the dialog, but needed for us to store the link to created page



3. Field to set status on the page - so users know exactly what content they are reading: is it draft, approved and final or just a completed, but not yet approved document

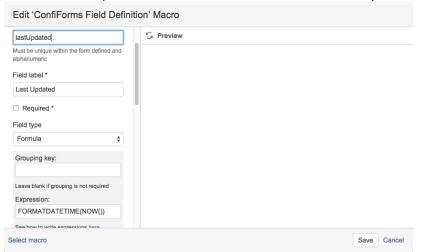


4. Labels which will be applied on the created page

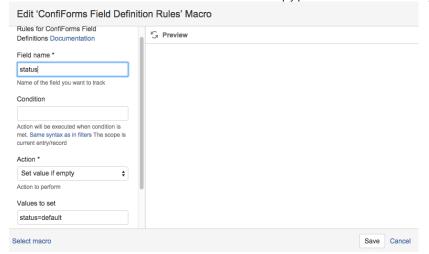


please note that we also set **separator** parameter to be ",". This is needed to enable correct handling of labels - otherwise when multiple labels are selected they are set as one Confluence label, not as multiple

5. Formula field to keep the information about when the record has been last time updated

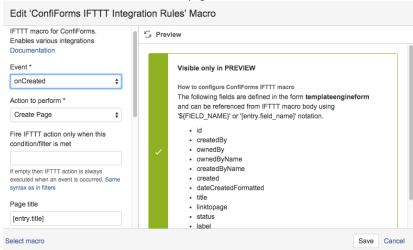


. There is a ConfiForms Field Definition Rules macro which simply preselects a "draft" status fro you when a dialog to create new page is shown

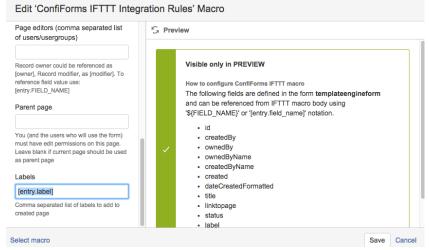


a small trick here is that it sets a status value to "default", as this is the colour selected to show a draft status. Check the Documentation for Status field type to see the colours and corresponding values

Then there is an IFTTT macro to create a page

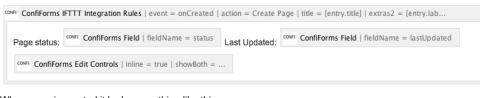


we set page title to be dynamic and based on the value entered on the form, in the field "title"



and we also set labels to be dynamic as well (Label field definition has a value separator set to ",", so the labels param get's given as a comma separated list)

The layout we have for the new pages is very simple (for demonstration)



When page is created it looks something like this:

Employee benefits

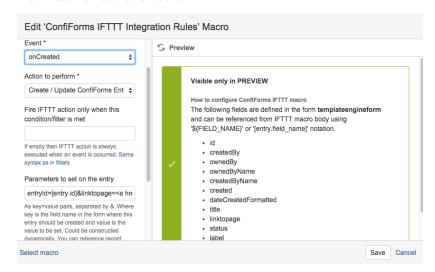
Created by admin about an nour ago
Page status: APPROVED Last Updated: Sep 17, 2015 09:58
Edit

 OK, and the last bit is the ConfiForms IFTTT macro as well, which looks in the Confluence editor like this, because of the \${}} construction used in macro parameters



Actually it is the same IFTTT macro as the fisrt one, but does a little bit different thing: it updates the original ConfiForms record and uses the result of a previous IFTTT macro execution (which is a created page, and the page object which is stored in the variable \${iftttResult_0}, and we use it's property "id", \${iftttResult_0.id}, which is actually a page id of the page which was created)

In edit mode the macro looks like this:



Important bits are: the action it does and the parameters

```
entryId=[entry.id]&linktopage=<a href="http://your_server/confluence/pages/viewpage.action?
pageId=${iftttResult_0.id}">[entry.title]</a>
```

As you can see, it updates the entry based on the entry id and updates one of the entry fields with this HTML link

Where

- [entry.id] is replaced with record id of the record created
- \${iftttResult_0.id} uses a result of the previous IFTTT macro execution, and it is a pageId it asks for
- [entry.title] is replace with a page title (which is entered on the form)

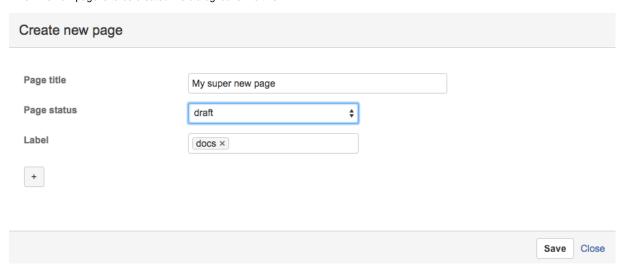
Let's say we have a field of this type called "mypage". In this case to update it with newly created page id you will do something like

entryId=[entry.id]&mypage=[iftttResult_0.id]

①

Important to know that IFTTT macro body is also evaluated as Velocity template (unless this feature is globally switched off by your administrators) - Configuring ConfiForms IFTTT actions and rules#CreatePage

When the new page is to be created the dialog looks like this



Why only 3 fields are visible? The reason is simple: hidden HTML and formula fields are not shown on the UI, but defined as field records and could be referenced to view their values

So, what happens when a user clicks "Save"

- A new page with a title given is created, the layout for the page created is defined as a body of IFTTT macro to create a page
- Label or labels are assigned/added to the created page
- When page is created another IFTTT macro kicks in to update original ConfiForms record with a link to created page (the concept in more
 details is described here: Creating links in original ConfiForms entry to just created page)



This is only a demonstration and the layout for the page created is very simple, you can of course add permissions to the Edit Controls we place on the page, so only certain user groups or users will be able to change page status. You can extend page layout as you wish, as long as the Confluence editor is happy about it or you can go "raw" and use Confluence storage format editor to edit it the way you like.

If you have any questions do not hesitate to contact us at vertuna(at)vertuna.com



Quick FAQ on creating pages

Why ConfiForms creates a broken page from Confluence template

When you setup ConfiForms to use Confluence templates to create the page it is very important to check how your ConfiForms IFTTT macro is configured

It should not contain any extras in it's macro body and only the reference to a template, as you can see in the example below

```
[template.template_with_columns]
```

as everything you have in your ConfiForms IFTTT macro body will go as a template contents and the page will be created from it. (No worries about tags around the template - we can recognise them and safely remove them)



This is extremely important when your Confluence template uses layouts! As Confluence is very fragile when the page is having a layout but does not start with a layout tags!

More on Confluence templates in ConfiForms Using Confluence templates with ConfiForms and avoid configuration duplication

Why ConfiForms "auto-wraps" everything with ConfiForms ListView on a created page

Before ConfiForms IFTTT creates a page it analyzes it's contents to see if you have some macros that need a "view" to work. This macros include: ConfiForms Field, ConfiForms Edit Controls, ConfiForms EntryViewer. But most important is ConfiForms Field macros that are placed alone, not inside the ConfiForms *View macros

When ConfiForms finds this it attempts to "fix" your template (and page) and adds the ConfiForms ListView macro around everything (filtered by current record). So your "viewless" ConfiForms Field macro will work

ConfiForms never auto-wraps the contents if each and every ConfiForms Field macro has been carefully placed inside "views" macros

Make sure you set the filter on the views to

id:[entry.id]

Filter to apply on entries dataset

id:[entry.id]

See examples in macro preview window

to dynamically show current record's data (this will be processed to have an actual ID, to limit the view to render the data for the current record's data that was used to create the page)

https://wiki.vertuna.com/display/CONFIFORMS/Advanced+integrations+with+IFTTT+macros+with+ConfiForms