

Sending the change log via email



This tutorial shows you how to get and send out the change log via email. It is a little bit technical, but the concept and the configuration is very straightforward and simple

In ConfiForms, since the introduction of the ["hasChanged" function](#) in version 1.36, you can create a change log of fields that were changed during the last update and send them out to interested parties.

Here is how this can be done.

(Please take a look at this tutorial to get the basics on how to send emails from ConfiForms - [Sending an email with attachments from Confluence with ConfiForms app](#))

We use Velocity templating features available in [IFTTT macro body](#) and we also use the fact that raw object for ConfiForms record is available in the context via name "entry".

This object has a method

```
hasChanged(propertyname)
```

which we will use in this scenario

So, imagine we have a form with 3 fields

- t1 - text field
- t2 - textarea field
- d1 - dropdown field

So, in order to notify ourselves about the change in the value for each field we need to do the following

CONFIForms

Form (Definition) | formName = myform

CONFIForms

Form Field (Definition) | fieldName = t1 | fieldLabel...

CONFIForms

Form Field (Definition) | fieldName = t2 | fieldLabel...

CONFIForms

Form Field (Definition) | fieldName = d1 | fieldLabel...

CONFIForms

IFTTT Integration Rules | event = onModified | action = Send Email | title = Change log for entry [entry.id] | wh...

Changelog

No Format

```
#if(${entry.hasChanged("t1")})
  New value for t1: [entry.t1]
#end
#if(${entry.hasChanged("t2")})
  New value for t2: [entry.t2]
#end
#if(${entry.hasChanged("d1")})
  New value for d1: [entry.d1]
#end
```

We use Velocity's `#if #end` construction and call for `hasChanged` method for each field we are interested in.

This is how you can get the change log for your record via email in ConfiForms.