

Contents

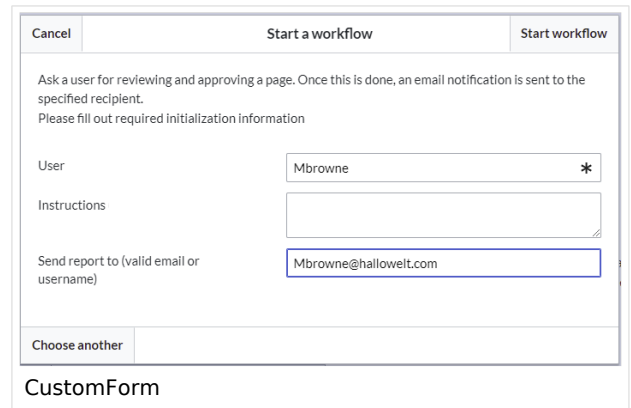
Manual:Extension/Workflows/Activity/CustomForm

Contents

1	Description	3
2	Extension elements	3
2.1	Example	4
3	Properties	4
3.1	Example	4

Description

The *CustomForm* activity is a generic activity, which is based on a custom form. It collects data for the corresponding workflow, which shall be executed. Therefore, the input fields depend on the type of workflow, which is chosen by the user. In general, the data is always the same and differs only in relation to the particular workflow and the number of users required. The *CustomForm* activity mostly include blank values, which can also be inserted by default. This is related to the metadata maintenance associated with the use of wikitext.



Short profile	
Name	CustomForm
Async	No
Input/form	<i>Input depends on the workflow type, but in general:</i> • User picker / Group picker • Textfield for comments • Textfield for mail / username • ...
BPMN Element	bpmn:userTask
BPMN Extension Element "wf: type"	custom_form

Extension elements

Name of extension element	Description	Type
formModule		element
formModule/module	ResourceLoader module to be loaded	string
formModule/class	Form definition class to be shown	string

Example

```
<bpmn:extensionElements>
  <wf:type>custom_form</wf:type>
  <wf:formModule>
    <wf:module>my.custom.form</wf:module>
    <wf:class>MyCustomForm</wf:class>
  </wf:formModule>
</bpmn:extensionElements>
```

This would load the ResourceLoader module called `my.custom.form` and expect it to have a JavaScript class `MyCustomForm` be implemented. Such a RL module may be provided by an extension, or a [custom gadget](#).

Properties

Name of property	Source	Description	Type
<code>due_date</code>	UIActivity	Due date for task completion	date/timestamp

Each field provided by the form specified in `formModule/class` needs to be listed as a property with a default value.

Example

If the form specified in `<wf:class>` looks like this

```
MyCustomForm = function( cfg, activity ) {
    MyCustomForm.parent.call( this, cfg, activity );
};

OO.inheritClass( MyCustomForm, workflows.object.form.Form );

MyCustomForm.prototype.getDefinitionItems = function() {
    return [
        {
            name: 'username',
            label: "Username",
            type: 'user_picker',
            required: true
        },
        {
            name: 'comment',
            label: 'Comment'
            type: 'wikitext',
        }
    ];
};
```

the properties can look like this

```
<bpmn:property name="username" />
<bpmn:property name="comment" default="Some default comment text" />
```

Given the activities `id` specified in the `<bpmn:userTask>` element was `"CollectData"`, connected activities would be able to access the output data in their respective properties like this:

```
<bpmn:task id="SomeActivity">
...
<bpmn:property name="user" default="{{{CollectData.username}}}"></bpmn:property>
<bpmn:property name="intro" default="{{{CollectData.comment}}}"></bpmn:property>
...
```