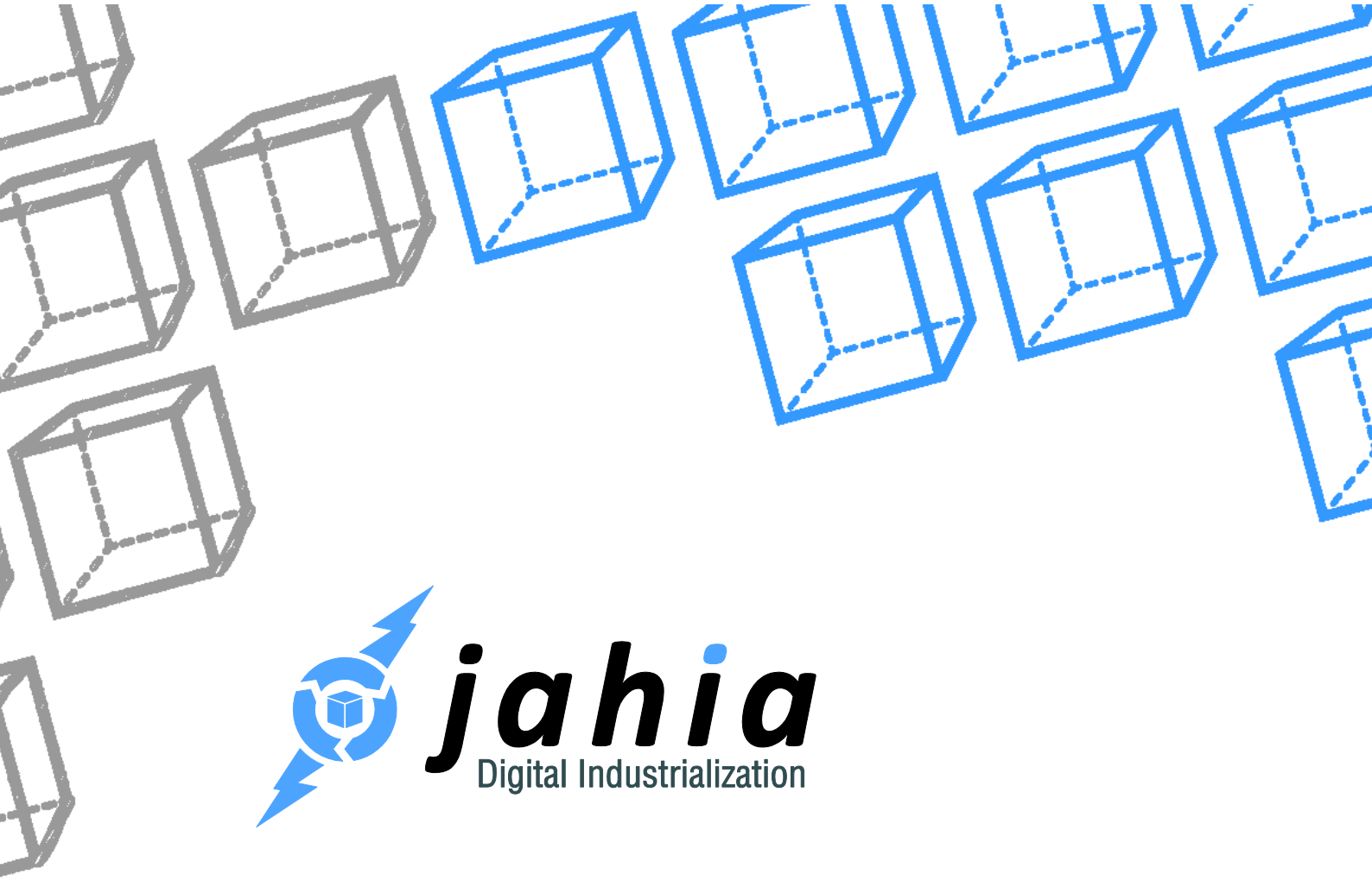


jahia
Digital Industrialization

Form Factory
Extending Input Fields, Validators and Results

Rooted in Open Source CMS, Jahia's Digital Industrialization paradigm is about streamlining Enterprise digital projects across channels to truly control time-to-market and TCO, project after project.

Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

http://www.jahia.com

Summary

1 Introduction

Jahia Form Factory is a module that makes it possible to build forms from scratch, letting users choose among input fields, validation rules, executable actions, labels etc.
It is possible to extend Form Factory with your own input blocks and you can define your own validations and actions (actions are explained in a different document).

This document explains how to add a new input block, its validations and its rendering in result views.

There are currently four object types in the Form Builder:

- *definition*
  These are the various types of input fields offered by the Form Builder (e.g., text, multiple choices, checkboxes…). These components can be used in simple or block form (consisting of several sub-components).
- *validation*
  Building components that make it possible to define new validations for (existing or new) definitions.
- *action*
  Actions define what happens at the end of the form, when users submit their data.
- *renderer*
  This is a simple rendering component that makes it possible to define how an input item will be rendered in views exploiting the submitted data.

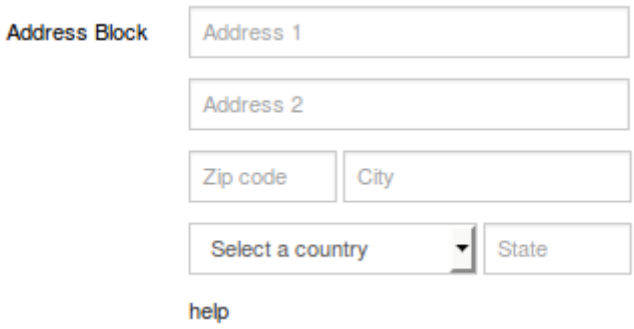
It should be noted that, although their structure is similar to *definition* and *action*, *renderer* and *validation* are actually sub-components of *definition*.

In all four cases, the creation process is similar and follows these steps:

- Declaration in the CND file
- Declaration of the snippet in *repository.xml*
- Creation of the associated view

2 Example Specification

In our example, we want to create an address block snippet, which will look like this:



The form developer, who wants to add address fields for users to fill in, will not need to add all of the fields one by one. Instead, they can simply add a single snippet to their form—a snippet that can have its own characteristics. It is particularly relevant in the case of an address block, considering that address formats can vary greatly from one country to another, and the external address validation systems, which are often associated with these types of forms, are also different for each country.

Since this snippet is specific, we will need to implement its own *validation* and its own *renderer*.

3 Step 1: Declaration in the CND File

Declaration in the CND file is simple: simply add the mixin that corresponds to the type of snippet.

3.1 Definition

In the case of our example, the snippet is of the *definition* type. This means that the declaration will have to contain the *fcmix:definition* mixin:

```
[fcnt:addressDefinition] > jnt:content, fcmix:definition, fcmix:blockInput,
mix:title, jmix:droppableContent, jmix:hiddenType
```

In our example, we want to create a block snippet: we have added the *fcmix:blockInput* mixin, which lets the module know about the type of snippet: simple (e.g., text input) or block.

3.2 Validation

In the case of the *validation* snippet of our *definition* snippet, the declaration will have to contain the *fcmix:validation* mixin:

```
[fcnt:addressValidation] > jnt:content, fcmix:validation, mix:title,
jmix:droppableContent, jmix:hiddenType
```

3.3 Renderer

In the case of the *renderer* snippet of our *definition* snippet, the declaration will have to contain the *fcmix:renderer* mixin:

```
[fcnt:addressRenderer] > jnt:content, jmix:droppableContent,
jmix:hiddenType, fcmix:renderer
```

3.4 Action

In the case where we would want to implement an *action* snippet, the declaration would have to contain the *fcmix:action* mixin:

```
[fcnt:sendDataToUrlAction] > jnt:content, fcmix:action, mix:title, jmix:droppableContent,
jmix:hiddenType
```

4 Step 2: Declaration in *repository.xml*

Declaration in the file *repository.xml* is a little more complex. Different configurations are possible, depending on the type of snippet.

4.1 Definition

Let's start with the *definition* snippet.

Declaration of the *definition* snippet in *repository.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<content xmlns:j="http://www.jahia.org/jahia/1.0"
xmlns:jcr="http://www.jcp.org/jcr/1.0">
    <modules jcr:primaryType="jnt:modules">
        <df-form-center-snippets-extension j:dependencies="default df-form-center
siteSettings" j:modulePriority="0" j:moduleType="module" j:title="DF Form Center
Snippets Extension" jcr:primaryType="jnt:module">
            <portlets jcr:primaryType="jnt:portletFolder"/>
            <files jcr:primaryType="jnt:folder"/>
            <contents jcr:primaryType="jnt:contentFolder">
                <form-center-definitions jcr:primaryType="jnt:contentFolder">
                    <addressblock jcr:primaryType="fcnt:addressDefinition"
tab="input"                          supportedValidationTypes="address">
                        <j:translation_en jcr:language="en"
                                  jcr:mixinTypes="mix:title"
                                  jcr:primaryType="jnt:translation"
                                  jcr:title="Address Block"
                                  label="ID / Name"/>
                        ....
                    </addressblock>
                </form-center-definitions>
             </contents>
             <templates j:rootTemplatePath="/base"
jcr:primaryType="jnt:templatesFolder">
                <files jcr:primaryType="jnt:folder"/>
                <contents jcr:primaryType="jnt:contentFolder"/>
             </templates>
        </df-form-center-snippets-extension>
    </modules>
</content>
```

As we can see above, we have added a ***<form-center-definitions>*** tag inside the ***<contents>*** tag. This tag is the one containing the tag that defines our snippet.

The tag also has a *tab* property, which defines in which tab of the form builder our snippet will be found, and a *supportedValidationTypes* property, which defines the validation categories supported by the snippet (the snippet could very well have several validation categories).

A snippet contains a group of subnodes that can be of two different types: *fcnt:definitionOptions* or *fcnt:definitionOptionsTranslatable* . These elements are the ones that allow form developers to customize snippets in the form builder.

Each subnode will be represented in the snippet's *popover* as defined by its type (the list of existing types is: *input*, *checkbox*, *select*, *textarea*, *textarea-split*, *inputdisabled*, *hidden*, *pagefinder*, *key-value*).
It should be noted that the *jsonValue* property of the *select* and *key-value* types contains a JSON object that must be written in *repository.xml* in the following way:

- *select*

```
<inputsize jcr:primaryType="fcnt:definitionOptions" type="select"
           jsonValue="[{&#34;label&#34;: &#34;Mini&#34;,&#34;value&#34;: &#34;input-
mini&#34;,&#34;selected&#34;: false},{&#34;label&#34;:
&#34;Small&#34;,&#34;value&#34;: &#34;input-small&#34;,&#34;selected&#34;:
false},{&#34;label&#34;: &#34;Medium&#34;,&#34;value&#34;: &#34;input-
medium&#34;,&#34;selected&#34;: false},{&#34;label&#34;:
&#34;Large&#34;,&#34;value&#34;: &#34;input-large&#34;,&#34;selected&#34;:
true},{&#34;label&#34;: &#34;Xlarge&#34;,&#34;value&#34;: &#34;input-
xlarge&#34;,&#34;selected&#34;: false},{&#34;label&#34;:
&#34;Xxlarge&#34;,&#34;value&#34;: &#34;input-xxlarge&#34;,&#34;selected&#34;:
false}]">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Input Size"/>
</inputsize>
```

- *key-value*

```
<options jcr:primaryType="fcnt:definitionOptionsTranslatable" type="key-value">
```

```
    <j:translation_en jcr:language="en" jcr:primaryType="jnt:translation"

jsonValue="[{&#34;key&#34;:&#34;optionone&#34;,&#34;value&#34;:&#34;Option
one&#34;},{&#34;key&#34;:&#34;optiontwo&#34;,&#34;value&#34;:&#34;Option two&#34;}]"
                        label="Options"/>
</options>
```

In our example, we need to define a general label, a general help text, a rendering for results and finally an ID and a placeholder for each field in our block.

In most cases, the names given to subnodes are of little importance, as long as the same names are used in the view.

There are, however, two exceptions:

- the subnode defining the id of a field must end with *_id*

```
<address1_id jcr:primaryType="fcnt:definitionOptions" type="hidden"
jsonValue="address1_id">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Address 1 Id"/>
</address1_id>
```

- the subnode defining the *renderer* must end with *_id*

```
<rendererName_id jcr:primaryType="fcnt:definitionOptions" type="hidden"
jsonValue="rendererName">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Renderer name"/>
</rendererName_id>
```

In the case of a subnode of the type *fcnt:definitionOptionsTranslatable*, the *jsonValue* property is *i18n*. This makes it possible to translate this element later on in the form translator.
Below is an example with the label:

```
<label jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      jsonValue="Address Block"
                      label="Address Block label"/>
</label>
```

To sum up, our *definition* snippet has the following structure:

```
[fcmix:definition] > mix:title mixin
 - tab (string, choicelist) = "input" mandatory < "input", "select", "radio",
"button", "hiddenbutton"
 - label (string) i18n mandatory
 - choiceField (string)
 - supportedValidationTypes (string) multiple
 + * (fcnt:definitionOptions) = fcnt:definitionOptions
 + validations (fcnt:validationRules)
```

The *choiceField* property is used in the case of a select, for instance, and will take the options value, the radios value in the case of radio buttons etc.

The structure of our subnodes:

```
[fcnt:definitionOptions] > jnt:content
 - type (string) mandatory
 - label (string) i18n
 - jsonValue (string)
 - choices (string) i18n
 - fieldId (string)

[fcnt:definitionOptionsTranslatable] > fcnt:definitionOptions, fcmix:translatable
 - jsonValue (string) i18n
```

4.2 Validation

Now let's add the *validation* snippet of our *definition* snippet.

Declaration of the *validation* snippet in *repository.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<content xmlns:j="http://www.jahia.org/jahia/1.0"
xmlns:jcr="http://www.jcp.org/jcr/1.0">
    <modules jcr:primaryType="jnt:modules">
        <df-form-center-snippets-extension j:dependencies="default df-form-center
siteSettings" j:modulePriority="0" j:moduleType="module" j:title="DF Form Center
Snippets Extension" jcr:primaryType="jnt:module">
            <portlets jcr:primaryType="jnt:portletFolder"/>
            <files jcr:primaryType="jnt:folder"/>
            <contents jcr:primaryType="jnt:contentFolder">
                <form-center-definitions jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-definitions>
                <form-center-validations jcr:primaryType="jnt:contentFolder">
                    <address jcr:primaryType="fcnt:addressValidation"
types="address">
                        <j:translation_en jcr:language="en"
                                          jcr:mixinTypes="mix:title"
                                          jcr:primaryType="jnt:translation"
                                          jcr:title="Address Required Validation"/>
                    ....
                </form-center-validations>
             </contents>
             <templates j:rootTemplatePath="/base"
jcr:primaryType="jnt:templatesFolder">
                <files jcr:primaryType="jnt:folder"/>
                <contents jcr:primaryType="jnt:contentFolder"/>
            </templates>
        </df-form-center-snippets-extension>
    </modules>
</content>
```

As we can see above, we have added a ***<form-center-validations>*** tag inside the ***<contents>*** tag. This tag is the one containing the tag that defines our *validation* snippet.

Here, the tag doesn't have a *tab* property, but a *types* property that defines the categories our *validation* snippet belongs to, and therefore on which snippet it will be found. (Just like the

supportedValidationTypes property, the *types* property could very well have several validation categories.)

As seen previously, a snippet contains a group of subnodes of the type *fcnt:definitionOptions* .

However, a *validation* snippet must always have the following two subnodes:

- the subnode that will activate the validation. The *jsonValue* property must be set to 'false'.

```
<activator jcr:primaryType="fcnt:definitionOptions" type="checkbox"
jsonValue="false">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Address Validation"/>
</activator>
```

- the subnode that will order the validations of a snippet. The *jsonValue* property must be set to '999' by default (this value will change when the user activates the validation or orders active validations).

```
<ordernumber jcr:primaryType="fcnt:definitionOptions" type="hidden" jsonValue="999">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Order"/>
</ordernumber>
```

In our example, we have also added the following four subnodes of the type *fcnt:definitionOptionsTranslatable* :

```
<messageaddress jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      jsonValue="Address 1 Field is mandatory"
                      label="Address 1 error text"/>
</messageaddress>
<messagecity jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
    <j:translation_en jcr:language="en"
```

```
                    jcr:primaryType="jnt:translation"
                    jsonValue="City is mandatory"
                    label="City error text"/>
</messagecity>
<messagecountry jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
    <j:translation_en jcr:language="en"
                    jcr:primaryType="jnt:translation"
                    jsonValue="Country is mandatory"
                    label="Country error text"/>
<messagecountry>
<messagezipcode jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
    <j:translation_en jcr:language="en"
                    jcr:primaryType="jnt:translation"
                    jsonValue="Zip code is mandatory"
                    label="Zip code error text"/>
</messagezipcode>
```

These are error messages that will be shown to the user if the validation was activated and an error occurs.

4.3 Renderer

We must now add the *renderer* snippet of our *validation* snippet.

Declaration of the *renderer* snippet in *repository.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<content xmlns:j="http://www.jahia.org/jahia/1.0"
xmlns:jcr="http://www.jcp.org/jcr/1.0">
    <modules jcr:primaryType="jnt:modules">
        <df-form-center-snippets-extension j:dependencies="default df-form-center
siteSettings" j:modulePriority="0" j:moduleType="module" j:title="DF Form Center
Snippets Extension" jcr:primaryType="jnt:module">
            <portlets jcr:primaryType="jnt:portletFolder"/>
            <files jcr:primaryType="jnt:folder"/>
            <contents jcr:primaryType="jnt:contentFolder">
                <form-center-definitions jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-definitions>
                <form-center-validations jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-validations>
                <form-center-renderers jcr:primaryType="jnt:contentFolder">
                    <address-renderer jcr:primaryType="fcnt:addressRenderer"
                                      rendererName="address">
                        <j:translation_en jcr:language="en"
                                          jcr:mixinTypes="mix:title"
                                          jcr:primaryType="jnt:translation"
                                          jcr:title="Address renderer"/>
                    </address-renderer>
                </form-center-renderers>
             </contents>
             <templates j:rootTemplatePath="/base"
jcr:primaryType="jnt:templatesFolder">
                <files jcr:primaryType="jnt:folder"/>
                <contents jcr:primaryType="jnt:contentFolder"/>
             </templates>
        </df-form-center-snippets-extension>
    </modules>
</content>
```

As we can see above, we have added a ***<form-center-renderers>*** tag inside the ***<contents>*** tag. This tag is the one containing the tag that defines our *renderer* snippet.

4.4 Action

If we wanted to add an *action* snippet, our file *repository.xml* would look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<content xmlns:j="http://www.jahia.org/jahia/1.0"
xmlns:jcr="http://www.jcp.org/jcr/1.0">
    <modules jcr:primaryType="jnt:modules">
        <df-form-center-snippets-extension j:dependencies="default df-form-center
siteSettings" j:modulePriority="0" j:moduleType="module" j:title="DF Form Center
Snippets Extension" jcr:primaryType="jnt:module">
            <portlets jcr:primaryType="jnt:portletFolder"/>
            <files jcr:primaryType="jnt:folder"/>
            <contents jcr:primaryType="jnt:contentFolder">
                <form-center-definitions jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-definitions>
                <form-center-validations jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-validations>
                <form-center-renderers jcr:primaryType="jnt:contentFolder">
                    ....
                </form-center-renderers>
                <form-center-actions jcr:primaryType="jnt:contentFolder">
                    <senddatatourl jcr:primaryType="fcnt:sendDataToUrlAction"
tab="action">
                        <j:translation_en jcr:primaryType="jnt:translation"
                                          jcr:language="en"
                                          jcr:title="Send data to URL"
                                          label="ID / Name"/>
                        ....
                    </senddatatourl>
                </form-center-actions>
            </contents>
            <templates j:rootTemplatePath="/base"
jcr:primaryType="jnt:templatesFolder">
                <files jcr:primaryType="jnt:folder"/>
                <contents jcr:primaryType="jnt:contentFolder"/>
            </templates>
        </df-form-center-snippets-extension>
    </modules>
</content>
```

As we can see above, we have added a ***<form-center-actions>*** tag inside the ***<contents>*** tag. This tag is the one containing the tag that defines our *action* snippet.

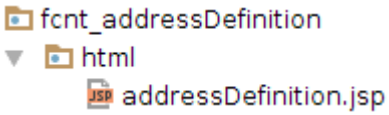
The action snippet must always contain the following subnode:

```
<actionname jcr:primaryType="fcnt:definitionOptions" type="inputdisabled"
            jsonValue="senddatatourl">
    <j:translation_en jcr:language="en"
                      jcr:primaryType="jnt:translation"
                      label="Action Name"/>
</actionname>
```

Here, the *jsonValue* property contains the name of the java action (Class) that will be executed when the form is submitted.

5 Step 3: Creation of the Associated View

Creating the view is quite simple. In the cases of the *definition*, *renderer* and *action* snippets, we use a standard Jahia jsp:



This jsp file contains html and JS code (the form builder uses underscore.js):

```
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="query" uri="http://www.jahia.org/tags/queryLib" %>
<%@ taglib prefix="jcr" uri="http://www.jahia.org/tags/jcr" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

<%--@elvariable id="currentNode" type="org.jahia.services.content.JCRNodeWrapper"--
%>
<%--@elvariable id="currentResource" type="org.jahia.services.render.Resource"--%>
<%--@elvariable id="flowRequestContext"
type="org.springframework.webflow.execution.RequestContext"--%>
<%--@elvariable id="out" type="java.io.PrintWriter"--%>
<%--@elvariable id="renderContext" type="org.jahia.services.render.RenderContext"--
%>
<%--@elvariable id="script" type="org.jahia.services.render.scripting.Script"--%>
<%--@elvariable id="scriptInfo" type="java.lang.String"--%>
<%--@elvariable id="url" type="org.jahia.services.render.URLGenerator"--%>
<%--@elvariable id="workspace" type="java.lang.String"--%>
<jcr:propertyInitializers var="countries" nodeType="fcnt:countryListDefinition"
name="country"/>

<div id="<@= id @>" class="control-group">
    <label class="control-label" for="<@= id @>_block_<@= address1_id @>">
        <@= label @>
    </label>
    <div class="controls">
        <input type="hidden" value="address" name="<@= id @>_block_<@=
rendererName_id @>">
        <input id="<@= address1_id @>" name="<@= id @>_block_<@= address1_id @>"
type="text" placeholder="<@= address1placeholder @>"
        <@ if(validations != undefined && validations.address) { @>
required="required" <@ } @> >
        <span id="<@= id @>_block_<@= address1_id @>" class="hide help-
inline"></span>
    </div>
    <div class="controls" style="margin-top: 10px;">
        <input id="<@= address2_id @>" name="<@= id @>_block_<@= address2_id @>"
```

```
type="text" placeholder="<@= address2placeholder @>">
    </div>
    <div class="controls controls-row" style="margin-top: 10px;">
        <input id="<@= zipcode_id @>" name="<@= id @>_block_<@= zipcode_id @>"
type="text" placeholder="<@= zipcodeplaceholder @>" class="input-mini"
        <@ if(validations != undefined && validations.address) { @>
required="required" <@ } @> >
        <input id="<@= city_id @>" name="<@= id @>_block_<@= city_id @>" type="text"
placeholder="<@= cityplaceholder @>" style="width: 130px;"
        <@ if(validations != undefined && validations.address) { @>
required="required" <@ } @> >
        <span id="<@= id @>_block_<@= zipcode_id @>" class="hide help-
inline"></span>
        <span id="<@= id @>_block_<@= city_id @>" class="hide help-inline"></span>
    </div>
    <div class="controls controls-row" style="margin-top: 10px;">
        <select id="<@= country_id @>" name="<@= id @>_block_<@= country_id @>"
style="width: 144px;"
        <@ if(validations != undefined && validations.address) { @>
required="required" <@ } @> >
        <option value=""><@= countryplaceholder @></option>
        <c:forEach items="${countries}" var="country">
            <option
value='{"country":{"key":"${country.value.string}","name":"${country.displayName}"}}
'>${country.displayName}</option>
        </c:forEach>
        </select>
        <input id="<@= state_id @>" name="<@= id @>_block_<@= state_id @>"
type="text" placeholder="<@= stateplaceholder @>" class="input-mini">
        <@ if(validations != undefined) { @>
        <span id="<@= id @>_block_<@= country_id @>" class="hide help-
inline"></span>
        <@ } @>
    </div>
    <div class="controls">
        <@ if (helptext.length > 0) { @>
        <p class="help-block"><@= helptext @></p>
        <@ } @>
    </div>
</div>
```

The JSP view is executed server-side, only one time per site and per language, and then the result is stored in a cache. This JSP view is used to build a JAVASCRIPT template that will be called for each occurrence of this block in the form.

In a block, the form's field names must be written as follows:

```
<@= id @>_block_<@= country_id @>
```

The value saved from a field can be of the JSON type, which makes it possible to save more complex objects than just text.

This is an example taken from the *countryListDefinition* view:

```
<jcr:propertyInitializers var="countries" nodeType="fcnt:countryListDefinition"
name="country"/>

<div class="control-group">
    <label class="control-label" for="<@= id @>">
        <@= label @>
    </label>
    <div class="controls">
        <select id="<@= id @>" name="<@= id @>" class="<@= inputsize @>" <@
if(validations != undefined && validations.required) { @> required="required" <@ }
@> >
            <option value=""><@= placeholder @></option>
            <c:forEach items="${countries}" var="country">
                <option
value='{"country":{"key":"${country.value.string}","name":"${country.displayName}"},
"rendererName":"country"}'>${country.displayName}</option>
            </c:forEach>
        </select>
        <@ if(validations != undefined) { @>
            <span class="hide help-inline"></span>
        <@ } @>
        <@ if (helptext.length > 0) { @>
            <p class="help-block"><@= helptext @></p>
        <@ } @>
    </div>
</div>
```

The value will be saved as follows:

```
value='{"country":{"key":"${country.value.string}","name":"${country.displayName}"},
"rendererName":"country"}'
```

This will give a *JSON* object containing a *country* sub-object with the *key* and *name* properties and a *rendererName* property that tells the system how to render this object.

5.1 Writing the Validation

In the case of the *validation* snippet, on the other hand, the content is different: in order to validate a form we call Backbone.Validation. The Backbone plugin is very simple to use and to extend. You may refer to our examples below and to the plugin's documentation for further information.

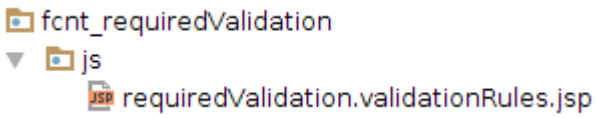
A validation view must provide Javascript. This is a simple example with the 'required' validation:

```
<jcr:node path="${currentNode.path}/message" var="nodeRequiredMessage"/>

<c:set var="requiredMessage"
value="${nodeRequiredMessage.properties['jsonValue'].string}"/>

{
    required: true
    <c:if test="${not empty requiredMessage}">
        ,msg: '${functions:escapeJavaScript(requiredMessage)}'
    </c:if>
}
```

Validation views are located in a *js* folder (not an *html* folder) and the view must be named *validationRules* .




```
<jcr:node path="${currentNode.path}/messageaddress"
var="nodeRequiredMessageAddress"/>
<jcr:node path="${currentNode.path}/messagecity" var="nodeRequiredMessageCity"/>
<jcr:node path="${currentNode.path}/messagecountry"
var="nodeRequiredMessageCountry"/>
<jcr:node path="${currentNode.path}/messagezipcode"
var="nodeRequiredMessageZipCode"/>

<c:set var="requiredMessageAddress"
value="${nodeRequiredMessageAddress.properties['jsonValue'].string}"/>
<c:set var="requiredMessageCity"
value="${nodeRequiredMessageCity.properties['jsonValue'].string}"/>
<c:set var="requiredMessageCountry"
value="${nodeRequiredMessageCountry.properties['jsonValue'].string}"/>
<c:set var="requiredMessageZipCode"
value="${nodeRequiredMessageZipCode.properties['jsonValue'].string}"/>
{
    fn: function(value, blockname, model) {
```

```
        //Address mandatory fields list
        var mandatoryFields = {
            <c:choose>
                <c:when test="${not empty requiredMessageAddress}">
                    address1:
'${functions:escapeJavaScript(requiredMessageAddress)}',
                </c:when>
                <c:otherwise>
                    address1: "<fmt:message key='address.error.address1'/>",
                </c:otherwise>
            </c:choose>
            <c:choose>
                <c:when test="${not empty requiredMessageCity}">
                    city: '${functions:escapeJavaScript(requiredMessageCity)}',
                </c:when>
                <c:otherwise>
                    city: "<fmt:message key='address.error.city'/>",
                </c:otherwise>
            </c:choose>
            <c:choose>
                <c:when test="${not empty requiredMessageCountry}">
                    country:
'${functions:escapeJavaScript(requiredMessageCountry)}',
                </c:when>
                <c:otherwise>
                    country: "<fmt:message key='address.error.country'/>",
                </c:otherwise>
            </c:choose>
            <c:choose>
                <c:when test="${not empty requiredMessageZipCode}">
                    zipcode: '${functions:escapeJavaScript(requiredMessageZipCode)}'
                </c:when>
                <c:otherwise>
                    zipcode: "<fmt:message key='address.error.zipcode'/>"
                </c:otherwise>
            </c:choose>
        }

        //Init error Object
        var fields = [];
        _.each(mandatoryFields, function(errorMessage, field){
            var fieldName = blockname + "_block_" + field + "_id";
            fields.push({name: fieldName, message: errorMessage, error: false});
        });

        //Check fields value
        _.each(mandatoryFields, function(errorMessage,field){
            var fieldName = blockname + "_block_" + field + "_id";
            //Get field error Object
            var errorObject = _.find(fields, function(field){
                return field.name == fieldName;
            });
            //set field error value
            errorObject.error = !model[blockname + "_block_" + field + "_id"];
```

```
        });

        //Get error State from fields
        var errorState = _.reduce(fields, function(errorBoolean, field){
            return errorBoolean || field.error;
        }, false);

        //return false if no error or field in the other case;
        return errorState?fields:errorState;
    }
}
```

6 Appendix

6.1 repository.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<content xmlns:j="http://www.jahia.org/jahia/1.0"
xmlns:jcr="http://www.jcp.org/jcr/1.0">
    <modules jcr:primaryType="jnt:modules">
        <df-form-center-snippets-extension j:dependencies="default df-form-center
siteSettings"
                                j:modulePriority="0"
                                j:moduleType="module"
                                j:title="DF Form Center Snippets Extension"
                                jcr:primaryType="jnt:module">

            <portlets jcr:primaryType="jnt:portletFolder"/>
            <files jcr:primaryType="jnt:folder"/>
            <contents jcr:primaryType="jnt:contentFolder">
                <form-center-definitions jcr:primaryType="jnt:contentFolder">
                    <addressblock jcr:primaryType="fcnt:addressDefinition"
tab="input" supportedValidationTypes="address">
                                <j:translation_en jcr:language="en"
                                        jcr:mixinTypes="mix:title"
                                        jcr:primaryType="jnt:translation"
                                        jcr:title="Address Block"
                                        label="ID / Name"/>
                        <label jcr:primaryType="fcnt:definitionOptionsTranslatable"
type="input" jcr:mixinTypes="fcmix:translatable">
                                <j:translation_en jcr:language="en"
                                                jcr:primaryType="jnt:translation"
                                                jsonValue="Address Block"
                                                label="Address Block label"/>
                        </label>
                        <helptext
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
```

```
                            <j:translation_en jcr:primaryType="jnt:translation"
                                              jcr:language="en"
                                              label="Help Text"
                                              jsonValue="help"/>
                        </helptext>
                        <rendererName_id jcr:primaryType="fcnt:definitionOptions"
type="hidden"
                                         jsonValue="rendererName">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Renderer name"/>
                        </rendererName_id>
                        <address1_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="address1_id">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Address 1 Id"/>
                        </address1_id>
                        <address1placeholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                                    <j:translation_en jcr:language="en"
                                           jcr:primaryType="jnt:translation"
                                           jsonValue="Address 1"
                                           label="Address 1 Placeholder"/>
                        </address1placeholder>

                        <address2_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="address2_id">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Address 2 Id"/>
                        </address2_id>
                        <address2placeholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="Address 2"
                                              label="Address 2 Placeholder"/>
                        </address2placeholder>

                        <zipcode_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="zipcode_id">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Zip code Id"/>
                        </zipcode_id>
                        <zipcodeplaceholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="Zip code"
```

```
                                        label="Zip code Placeholder"/>
                    </zipcodeplaceholder>

                    <city_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="city_id">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        label="City Id"/>
                    </city_id>
                    <cityplaceholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        jsonValue="City"
                                        label="City Placeholder"/>
                    </cityplaceholder>

                    <country_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="country_id">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        label="Country Id"/>
                    </country_id>
                    <countryplaceholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        jsonValue="Select a country"
                                        label="Country first option"/>
                    </countryplaceholder>

                    <state_id jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="state_id">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        label="State Id"/>
                    </state_id>
                    <stateplaceholder
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input"
jcr:mixinTypes="fcmix:translatable">
                        <j:translation_en jcr:language="en"
                                        jcr:primaryType="jnt:translation"
                                        jsonValue="State"
                                        label="State Placeholder"/>
                    </stateplaceholder>
                </addressblock>
            </form-center-definitions>
            <form-center-actions jcr:primaryType="jnt:contentFolder">
                <senddatatourl jcr:primaryType="fcnt:sendDataToUrlAction"
tab="action">
                    <j:translation_en jcr:primaryType="jnt:translation"
jcr:language="en" jcr:title="Send data to URL" label="ID / Name"/>
```

```
                        <label jcr:primaryType="fcnt:definitionOptions"
type="inputdisabled" jsonValue="Send data to URL">
                            <j:translation_en jcr:language="en"
jcr:primaryType="jnt:translation" label="Label Action"/>
                        </label>
                        <actionname jcr:primaryType="fcnt:definitionOptions"
type="inputdisabled" jsonValue="senddatatourl">
                            <j:translation_en jcr:language="en"
jcr:primaryType="jnt:translation"  label="Action Name"/>
                        </actionname>
                        <sendto jcr:primaryType="fcnt:definitionOptions"
type="input" jsonValue="">
                            <j:translation_en jcr:language="en"
jcr:primaryType="jnt:translation" label="Send to" />
                        </sendto>
                        <methodtouse jcr:primaryType="fcnt:definitionOptions"
type="inputdisabled" jsonValue="POST">
                            <j:translation_en jcr:language="en"
jcr:primaryType="jnt:translation" label="Method to use" />
                        </methodtouse>
                        <parametername jcr:primaryType="fcnt:definitionOptions"
type="input" jsonValue="">
                            <j:translation_en jcr:language="en"
jcr:primaryType="jnt:translation" label="Parameter name" />
                        </parametername>
                    </senddatatourl>
                </form-center-actions>
                <form-center-validations jcr:primaryType="jnt:contentFolder">
                    <address jcr:primaryType="fcnt:addressValidation"
types="address">
                        <j:translation_en jcr:language="en"
                                          jcr:mixinTypes="mix:title"
                                          jcr:primaryType="jnt:translation"
                                          jcr:title="Address Required Validation"/>
                        <activator jcr:primaryType="fcnt:definitionOptions"
type="checkbox" jsonValue="false">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Address Validation"/>
                        </activator>
                        <messageaddress
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="Address 1 Field is
mandatory"
                                              label="Address 1 error text"/>
                        </messageaddress>
                        <messagecity
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="City is mandatory"
                                              label="City error text"/>
```

```
                        </messagecity>
                        <messagecountry
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="Country is mandatory"
                                              label="Country error text"/>
                        </messagecountry>
                        <messagezipcode
jcr:primaryType="fcnt:definitionOptionsTranslatable" type="input">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              jsonValue="Zip code is mandatory"
                                              label="Zip code error text"/>
                        </messagezipcode>
                        <ordernumber jcr:primaryType="fcnt:definitionOptions"
type="hidden" jsonValue="999">
                            <j:translation_en jcr:language="en"
                                              jcr:primaryType="jnt:translation"
                                              label="Order"/>
                        </ordernumber>
                    </address>
                </form-center-validations>
                <form-center-renderers jcr:primaryType="jnt:contentFolder">
                    <address-renderer jcr:primaryType="fcnt:addressRenderer"
rendererName="address">
                        <j:translation_en jcr:language="en"
                                          jcr:mixinTypes="mix:title"
                                          jcr:primaryType="jnt:translation"
                                          jcr:title="Address renderer"/>
                    </address-renderer>
                </form-center-renderers>
            </contents>
                <templates j:rootTemplatePath="/base"
jcr:primaryType="jnt:templatesFolder">
                    <files jcr:primaryType="jnt:folder"/>
                    <contents jcr:primaryType="jnt:contentFolder"/>
                </templates>
        </df-form-center-snippets-extension>
    </modules>
</content>
```

Jahia Solutions Group SA

9 route des Jeunes,
CH-1227 Les acacias
Geneva, Switzerland

http://www.jahia.com