Email Template Guide

This document describes how to create your own email template. Templates can be used by Form Factory actions such as ‘Send Email Action’.

In order for Form Factory to discover custom email templates, we need to modify our pom.xml and add Form Factory as a dependency of our module.

Dependency Requirement

In **pom.xml**:

```
<dependencies>
  <dependency>
    <groupId>org.jahia.modules</groupId>
    <artifactId>form-factory-core</artifactId>
    <version>2.1.2</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

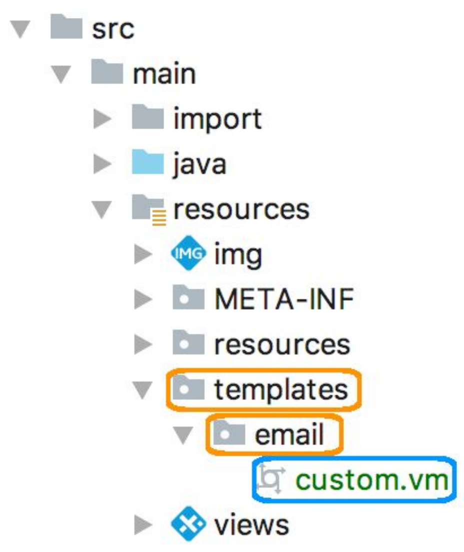
```
<plugin>
  <groupId>org.apache.felix</groupId>
  <artifactId>maven-bundle-plugin</artifactId>
  <extensions>true</extensions>
  <configuration>
    <instructions>
      <Jahia-Depends>default,siteSettings,form-factory-core</Jahia-Depends>
      <!--Higher priority than form-factory-core (2)-->
      <Jahia-Module-Priority>3</Jahia-Module-Priority>
    </instructions>
  </configuration>
</plugin>
```

Velocity Template Language

Templates are written in *Velocity Template Language* *(VTL).* These files can be identified by the extension ***.vm*** and allow us to access java code. Within your action(ex: *SendEmailAction)* you can bind variables to the template, which can then be accessed and expressed.

Folder Structure

Templates should be created under the src/main/resources/templates/***email*** directory.
By default **templates** folder does not exist in a newly created module. If that is the case, it must be created and the subdirectory ***'email'*** under it.



custom.vm will be the name of our custom template.

Template Parameter Binding

In our action(example taken from *Send Email Action*) we can pass parameters to our template by creating a ***Map<String, Object>*** Object; Store our parameters within it and pass the map as a parameter to ***sendMessageWithTemplate*** method of ***org.jahia.services.mail.MailService*** class.

For a more convenient approach, we can create an instance of ***org.jahia.modules.formfactory.actions.models.Email*** class and use the ***addBinding*** method, to store all the parameters that we want available within our template(see Send Email Action for an example).

Template

Within the template we can declare the variable by:

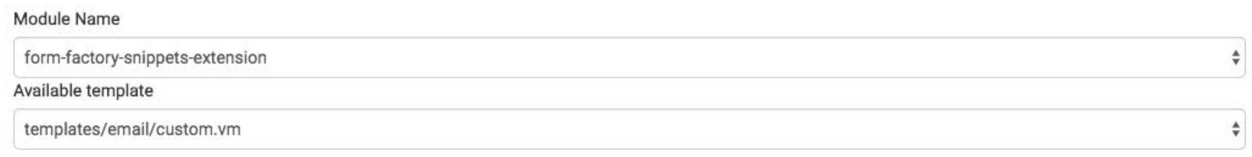
```
#* @vtlvariable name="formName" type="java.lang.String" *#
#* @vtlvariable name="formDatas" type="java.util.Map" *#
#* @vtlvariable name="bundle" type="java.util.ResourceBundle" *#
#* @vtlvariable name="emailBodyMessage" type="java.lang.String" *#
```

And make use of the binding(s) as follows:

```
...
<body>
...
<table cellpadding="0" cellspacing="0" border="1" class="table table-striped table-bordered"
id="result_table">
   ...
   <tbody id="tableBody">
       #foreach( $it in $formDatas.keySet())
           <tr>
               <td>&nbsp;<label style="font:bold"> $it</label>&nbsp;</td>
               <td>&nbsp;$formDatas.get($it)&nbsp;</td>
           </tr>
       #end
   </tbody>
</table>
</body>
```

Template Selection

Using the *Send Email Action* as an example, we can configure the action to use our custom template by selecting it from the dropdown menu(located in the action configuration panel):


Module Name
form-factory-snippets-extension
Available template
templates/email/custom.vm

The emails sent by *Send Email Action* will now use our custom.vm template!