

Form Field Element Plugin

- Usages
- Abstract Class
 - org.joget.apps.form.model.Element
 - Method Detail
 - Abstract Methods
 - renderTemplate
 - Overridable Methods
 - formatData
 - formatDataForValidation
 - selfValidate
 - render
 - renderErrorTemplate
 - renderReadOnlyTemplate
 - continueValidation
 - getDynamicFieldNames
 - hasError
 - isAuthorize
 - Utility Methods
 - getChildren
 - getChildren
 - setChildren
 - getParent
 - setParent
 - getPrimaryKeyValue
 - getLoadBinder
 - setLoadBinder
 - getOptionsBinder
 - setOptionsBinder
 - getStoreBinder
 - setStoreBinder
 - getValidator
 - setValidator
 - getCustomParameterName
 - setCustomParameterName
 - getDefaultPropertyValues
 - org.joget.apps.form.model.AbstractSubForm
 - Field Detail
 - Constants
 - PROPERTY_PARENT_SUBFORM_ID
 - PROPERTY_SUBFORM_PARENT_ID
 - Method Detail
 - Overridable Methods
 - loadSubForm
 - updateElementParameterNames
 - populateParentWithSubFormKey
 - populateSubFormWithParentKey
 - checkForRecursiveForm
 - getSubForm
- Interface
 - org.joget.apps.form.model FormBuilderPaletteElement
 - Method Detail
 - Interface Methods
 - getFormBuilderCategory
 - getFormBuilderPosition
 - getFormBuilderIcon
 - getDefaultPropertyValues
 - getFormBuilderTemplate
 - org.joget.apps.form.model FormBuilderContainer
 - org.joget.apps.form.model FormBuilderOptionsElement
 - org.joget.apps.form.model FormBuilderAjaxOptionsElement
 - Method Detail
 - Interface Methods
 - getControlElement
 - org.joget.apps.form.model FormBuilderReferenceDataRetriever
 - Method Detail
 - Interface Methods
 - loadFormRows
- Plugin Properties Options
 - Mandatory Property

- id
 - Predefined Property
 - loadBinder
 - optionsBinder
 - permission
 - storeBinder
 - readonly
 - readonlyLabel
 - workflowVariable
 - value
 - validator
- Form Field Element Template Sample
 - Explanations
- Tutorials
- Related Community Plugins

Usages

- Extend types of fields available in Form Builder.
- A useable Form Field Element plugin must extends [org.joget.apps.form.model.Element](#) abstract class and implements [org.joget.apps.form.model FormBuilderPaletteElement](#) interface.

Abstract Class

org.joget.apps.form.model.Element

- Under wflow-core module
- Extended [org.joget.plugin.base.ExtDefaultPlugin](#). Please refer to [Plugin Base Abstract Class and Interface](#).
- Implemented [org.joget.plugin.property.model.PropertyEditable](#). Please refer to [Plugin Base Abstract Class and Interface](#).
- A base abstract class to develop a Form Field Element plugin.
- All forms, containers and form fields must extend this class.

Method Detail

Abstract Methods

renderTemplate

```
public abstract java.lang.String renderTemplate(org.joget.apps.form.model.FormData formData, java.util.Map dataModel)
```

HTML template for front-end UI.

Parameters:

- dataModel* - Model containing values to be displayed in the template.

Overridable Methods

formatData

```
public org.joget.apps.form.model.FormRowSet formatData(org.joget.apps.form.model.FormData formData)
```

Method that retrieves loaded or submitted form data, and formats it for a store binder. The formatted data is to be stored and returned in a FormRowSet.

formatDataForValidation

```
public org.joget.apps.form.model.FormData formatDataForValidation(org.joget.apps.form.model.FormData formData)
```

Method for override to perform format data in request parameter before execute validation

selfValidate

```
public java.lang.Boolean selfValidate(org.joget.apps.form.model.FormData formData)
```

Method for override to perform specify validation for this field.

Error message can display with following code:

```
String id = FormUtil.getElementParameterName(this);

formData.addFormError(id, "Error!!");
```

render

```
public java.lang.String render(org.joget.apps.form.model.FormData formData, java.lang.Boolean includeMetaData)
```

Render HTML template for UI, with option for form builder design mode. This method will call abstract method **renderTemplate** for rendering.

Parameters:

- includeMetaData - set true to render additional meta required for the Form Builder.

renderErrorTemplate

```
public java.lang.String renderErrorTemplate(org.joget.apps.form.model.FormData formData, java.util.Map dataModel)
```

HTML template with errors for front-end UI. This method will call abstract method **renderTemplate** for rendering.

Parameters:

- dataModel - Model containing values to be displayed in the template.

renderReadOnlyTemplate

```
public java.lang.String renderReadOnlyTemplate(org.joget.apps.form.model.FormData formData, java.util.Map dataModel)
```

Read-only HTML template for front-end UI (Not used at the moment)

Parameters:

- dataModel - Model containing values to be displayed in the template.

continueValidation

```
public boolean continueValidation(org.joget.apps.form.model.FormData formData)
```

Flag to indicate whether or not continue validating descendent elements.

getDynamicFieldNames

```
public Collection<java.lang.String> getDynamicFieldNames()
```

Used to create multiple form data column in database by returning extra column names.

hasError

```
public java.lang.Boolean hasError(org.joget.apps.form.model.FormData formData)
```

Flag to indicate whether or not this field has fail the validation process

isAuthorize

```
public java.lang.Boolean isAuthorize(org.joget.apps.form.model.FormData formData)
```

Flag to indicate whether or not the current logged in user is authorized to view this field in the form.

It used property key "permission" to retrieve Form Permission plugin.

Utility Methods

getChildren

```
public java.util.Collection<org.joget.apps.form.model.Element> getChildren()
```

Retrieves all children form field element under this field

getChildren

```
public void setChildren(java.util.Collection<org.joget.apps.form.model.Element> children)
```

Retrieves all children form field element under this field

setChildren

```
public void setChildren(java.util.Collection<org.joget.apps.form.model.Element> children)
```

Sets form fields as children of this field

getParent

```
public org.joget.apps.form.model.Element getParent()
```

Returns the immediate parent for this element.

setParent

```
public void setParent(org.joget.apps.form.model.Element parent)
```

Sets the immediate parent for this element.

getPrimaryKeyValue

```
public java.lang.String getPrimaryKeyValue(org.joget.apps.form.model.FormData formData)
```

Returns the primary key value for the current element. Defaults to the primary key value of the form.

getLoadBinder

```
public org.joget.apps.form.model.FormLoadBinder getLoadBinder()
```

Get load binder

setLoadBinder

```
public void setLoadBinder(org.joget.apps.form.model.FormLoadBinder loadBinder)
```

Set load binder

getOptionsBinder

```
public org.joget.apps.form.model.FormLoadBinder getOptionsBinder()
```

Gets an Options Binder

setOptionsBinder

```
public void setOptionsBinder(org.joget.apps.form.model.FormLoadBinder optionsBinder)
```

Sets an Options Binder

getStoreBinder

```
public org.joget.apps.form.model.FormStoreBinder getStoreBinder()
```

Gets a Store Binder

setStoreBinder

```
public void setStoreBinder(org.joget.apps.form.model.FormStoreBinder storeBinder)
```

Sets a Store Binder

getValidator

```
public org.joget.apps.form.model.Validator getValidator()
```

Gets a validator

setValidator

```
public void setValidator(org.joget.apps.form.model.Validator validator)
```

Sets a validator

getCustomParameterName

```
public java.lang.String getCustomParameterName()
```

If non-null, this is to be used as the HTML input name for the element.

setCustomParameterName

```
public void setCustomParameterName(java.lang.String customParameterName)
```

Sets a custom parameter name for the HTML input name of the element.

getDefaultPropertyValues

```
public java.lang.String getDefaultPropertyValues()
```

Set default Plugin Properties Options value to a new added Field in Form Builder.

org.joget.apps.form.model.AbstractSubForm

- Under wflow-core module
- Extended [org.joget.apps.form.model.Element](#).
- Implemented [org.joget.apps.form.model.FormContainer](#).
- Use to develop a Form Field element which embed a Form as its child.

Field Detail

Constants

PROPERTY_PARENT_SUBFORM_ID

```
public static final String PROPERTY_PARENT_SUBFORM_ID = "parentSubFormId";
```

Property key that use to retrieve the field id in parent form used to store subform primary key as reference key.

PROPERTY_SUBFORM_PARENT_ID

```
public static final String PROPERTY_SUBFORM_PARENT_ID = "subFormParentId";
```

Property key that use to retrieve the field id in subform used to store parent form primary key as foreign key.

Method Detail

Overridable Methods

loadSubForm

```
protected org.joget.apps.form.model.Form loadSubForm(org.joget.apps.form.model.FormData parentFormData)
throws org.springframework.beans.BeansException
```

Retrieve a Form object as subform. This method will use either value from property key "formDefId" or "json" to construct the Form object.

updateElementParameterNames

```
protected void updateElementParameterNames(org.joget.apps.form.model.Element element, java.lang.String
prefix)
```

Update all the parameter name of field elements in subform with a prefix

populateParentWithSubFormKey

```
protected void populateParentWithSubFormKey(org.joget.apps.form.model.FormData formData)
```

Update parent form field value with primary key of subform based on property key of this constant PROPERTY_PARENT_SUBFORM_ID.

populateSubFormWithParentKey

```
protected void populateSubFormWithParentKey(org.joget.apps.form.model.FormData formData)
```

Update subform field value with primary key of parent form based on property key of this constant PROPERTY_SUBFORM_PARENT_ID.

checkForRecursiveForm

```
protected boolean checkForRecursiveForm(org.joget.apps.form.model.Element e, java.lang.String id)
```

Check the subform is not exist in the parent elements tree.

getSubForm

```
protected org.joget.apps.form.model.Form getSubForm(org.joget.apps.form.model.FormData formData)
```

Get From object from its children.

Interface

org.joget.apps.form.model FormBuilderPaletteElement

- Under wflow-core module
- Extends interface [org.joget.apps.form.model FormBuilderEditable](#).
- Interface that describes meta information used for adding an element into the Form Builder palette.

Method Detail

Interface Methods

[getFormBuilderCategory](#)

```
public java.lang.String getFormBuilderCategory()
```

Category for the element in the Form Builder palette

[getFormBuilderPosition](#)

```
public int getFormBuilderPosition()
```

Ordering position. Palette to display based on the position value in ascending order for a category.

[getFormBuilderIcon](#)

```
public java.lang.String getFormBuilderIcon()
```

Path to icon for the element in the Form Builder palette. This path is relative to the context path. Return NULL to use the default icon.

[getDefaultPropertyValues](#)

```
public java.lang.String getDefaultPropertyValues()
```

Set default Plugin Properties Options value to a new added Field in Form Builder. This method is implemented in [org.joget.apps.form.model.Element](#).

[getFormBuilderTemplate](#)

```
public java.lang.String getFormBuilderTemplate()
```

HTML template used for display a new added field in the Form Builder

org.joget.apps.form.model.FormContainer

- Under wflow-core module
- This interface indicate that a Form Field Element is a container and should not create a form data column in database.
- No interface method is available in this interface

org.joget.apps.form.model.FormOptionsElement

- Under wflow-core module
- This interface indicate that a Form Field Element is a multi options field such as Select Box, Check Box & Radio Button. It can use [Form Options Binder](#) to populate its options.
- No interface method is available in this interface

org.joget.apps.form.model.FormAjaxOptionsElement

- Under wflow-core module
- Extends interface [org.joget.apps.form.model.FormOptionsElement](#).
- This interface indicate that a Form Field Element is a multi options field such as Select Box, Check Box & Radio Button. It can use [Form Options Binder](#) which implemented [org.joget.apps.form.model.FormAjaxOptionsBinder](#) to populate its options using AJAX.

Method Detail

Interface Methods

[getControlElement](#)

```
public org.joget.apps.form.model.Element getControlElement(org.joget.apps.form.model.FormData formData)
```

Get dependent field element which use to control the options of this field

org.joget.apps.form.model.FormReferenceDataRetriever

- Under wflow-core module
- Indicate that this field hold the primary keys for other form data records. The reference records can be retrieve using its interface method.
- This interface is used by [org.joget.apps.form.service.FormUtil.loadFormData\(\)](#) and [org.joget.apps.form.service.FormUtil.loadFormDataJson\(\)](#) method to fetch submitted form data values including data from subforms, and reference fields.

Method Detail

Interface Methods

loadFormRows

```
public org.joget.apps.form.model.FormRowSet loadFormRows(String[] primaryKeyValues, org.joget.apps.form.model.FormData formData)
```

Retrieve form data rows for an array of specified primary key values.

Plugin Properties Options

- The following are some mandatory properties and system predefined properties can be used in your Plugin Properties Options JSON file. Please refer to [Plugin Properties Options](#) for more information.

Mandatory Property

id

- All Form Element Field must has this property appear in the Properties Options JSON.
- This is used as unique identifier of a field in a form.

```
{
  name : 'id',
  label : '@@form.textfield.id@@',
  type : 'textfield',
  required : 'True',
  regex_validation : '^[a-zA-Z0-9_]+$',
  validation_message : '@@form.textfield.invalidId@@'
}
```

Predefined Property

loadBinder

- If this property appear in the Properties Options JSON, admin user is able to choose a Form Load Binder.
- The value of this property will auto convert into Form Load Binder Plugin and the plugin can be retrieved by [getLoadBinder](#) method.

```
{
  name:'loadBinder',
  label:'@@form.subform.loadBinder@@',
  type:'elementselect',
  options_ajax:'[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.
FormLoadElementBinder',
  url:'[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions',
  value:'org.joget.apps.form.lib.WorkflowFormBinder',
  required:'True'
}
```

optionsBinder

- If this property appear in the Properties Options JSON, admin user is able to choose a Form Options Binder.
- The value of this property will auto convert into Form Options Binder Plugin and the plugin can be retrieved by [getOptionsBinder](#) method.

```
{
  name : 'optionsBinder',
  label : '@@form.checkbox.chooseOptionsBinder@@',
  type : 'elementselect',
  options_ajax : '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.
FormLoadOptionsBinder',
  url : '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
}
```

permission

- If this property appear in the Properties Options JSON, admin user is able to choose a Form Permission.
- The value of this property will auto convert into Form Permission Plugin and the plugin is used by [isAuthorize](#) method.

```
{
  name: 'permission',
  label: '@@form.form.permission@@',
  type: 'elementselect',
  options_ajax: '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.
FormPermission',
  url: '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
}
```

storeBinder

- If this property appear in the Properties Options JSON, admin user is able to choose a Form Store Binder.
- The value of this property will auto convert into Form Store Binder Plugin and the plugin can be retrieved by [getStoreBinder](#) method.

```
{
  name: 'storeBinder',
  label: '@@form.form.storeBinder@@',
  type: 'elementselect',
  options_ajax: '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.
FormStoreElementBinder',
  url: '[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions'
}
```

readonly

- System use this property to flag a field a readonly.
- If this property appear in the Properties Options JSON, admin user can force this field to display as readonly.
- This value is ignored if the parent form is set to readonly

```
{
  name : 'readonly',
  label : '@@form.checkbox.readonly@@',
  type : 'checkbox',
  value : 'false',
  options : [{
    value : 'true',
    label : ''
  }]
}
```

readonlyLabel

- System use this property to flag a field a readonly field should display as label or not.
- If this property appear in the Properties Options JSON, admin user can force this field to display as label when it is readonly.
- This value is ignored if the parent form is set to display as label when it is readonly

```
{
  name : 'readonlyLabel',
  label : '@@form.checkbox.readonlyLabel@@',
  type : 'checkbox',
  value : 'false',
  options : [{
    value : 'true',
    label : ''
  }]
}
```

workflowVariable

- If this property appear in the Properties Options JSON, admin user is set to store this field value to a workflow variable.
- The value of this field will auto set to the mentioned workflow variable if the Form Store Binder is set to [Workflow Form Binder](#).

```
{
  name : 'workflowVariable',
  label : '@@form.checkbox.workflowVariable@@',
  type : 'textfield'
}
```

value

- If this property appear in the Properties Options JSON, admin user is able to set a default value for this field.

```
{
  name : 'value',
  description : '@@form.checkbox.value.desc@@',
  label : 'Default @@form.checkbox.value@@',
  type : 'textfield'
}
```

validator

- If this property appear in the Properties Options JSON, admin user is able to choose a Form Validator.
- The value of this property will auto convert into Form Validator Plugin and the plugin can be retrieved by [getValidator](#) method.

Form Field Element Template Sample

Following is a sample Form Field Template of a [Text Field](#) element. It is constructed using [FreeMaker](#) syntax. This template will be used and returned by [renderTemplate](#) method.

```
@Override
public String renderTemplate(FormData formData, Map dataModel) {
    String templateName = "textField.ftl";

    // set value
    String value = FormUtil.getElementPropertyValue(this, formData);
    dataModel.put("value", value);

    String html = FormUtil.generateElementHtml(this, formData, templateName, dataModel);
    return html;
}
```

```
<div class="form-cell" ${elementMetaData!}>
  <label class="label">
    ${element.properties.label} <span class="form-cell-validator">${decoration}</span>
    <#if error??>
      <span class="form-error-message">${error}</span>
    </#if>
  </label>
  <#if (element.properties.readonly! == 'true' && element.properties.readonlyLabel! == 'true') >
    <div class="form-cell-value"><span>${value!?html}</span></div>
    <input id="${elementParamName!}" name="${elementParamName!}" type="hidden" value="${value!?html}" />
  <#else>
    <input id="${elementParamName!}" name="${elementParamName!}" type="text" size="${element.properties.size!}" value="${value!?html}" maxlength="${element.properties.maxlength!}" <#if error??>class="form-error-cell"
  </#if> <#if element.properties.readonly! == 'true'>readonly</#if> />
  </#if>
</div>
```

Explanations

- A form field element template should be wrapped under a div with "form-cell" class.

- \$(elementMetadata) need to put in the ".form-cell" div. It is used by Form Builder.
- A useable field element template must handle the following states of a field.
 - Has errors
 - Readonly
 - Readonly and needed to display as label
 - Normal display

Tutorials

- [Guideline for developing a plugin](#)
- [File Upload Form Element Integrated with Amazon S3](#)
- [How to develop a Mood Rating Form Field](#)

Related Community Plugins

- **Check Box**
org.joget.apps.form.lib.CheckBox (wflow-core)
- **Custom HTML**
org.joget.apps.form.lib.CustomHTML (wflow-core)
- **Date Picker**
org.joget.apps.form.lib.DatePicker (wflow-core)
- **File Upload**
org.joget.apps.form.lib.FileUpload (wflow-core)
- **Grid**
org.joget.apps.form.lib.Grid (wflow-core)
- **Hidden Field**
org.joget.apps.form.lib.HiddenField (wflow-core)
- **Id Generator Field**
org.joget.apps.form.lib.IdGeneratorField (wflow-core)
- **Password Field**
org.joget.apps.form.lib.PasswordField (wflow-core)
- **Radio Button**
org.joget.apps.form.lib.Radio (wflow-core)
- **Select Box**
org.joget.apps.form.lib.SelectBox (wflow-core)
- **Subform**
org.joget.apps.form.lib.SubForm (wflow-core)
- **Text Area**
org.joget.apps.form.lib.TextArea (wflow-core)
- **Text Field**
org.joget.apps.form.lib.TextField (wflow-core)