Plugin Properties Options

Usage

- Plugin Properties Options allow a plugin to gather configuration data from a plugin user.

- Plugin Properties Options should be returned in the abstract method "getPropertyOptions" of each plugin.

Structure

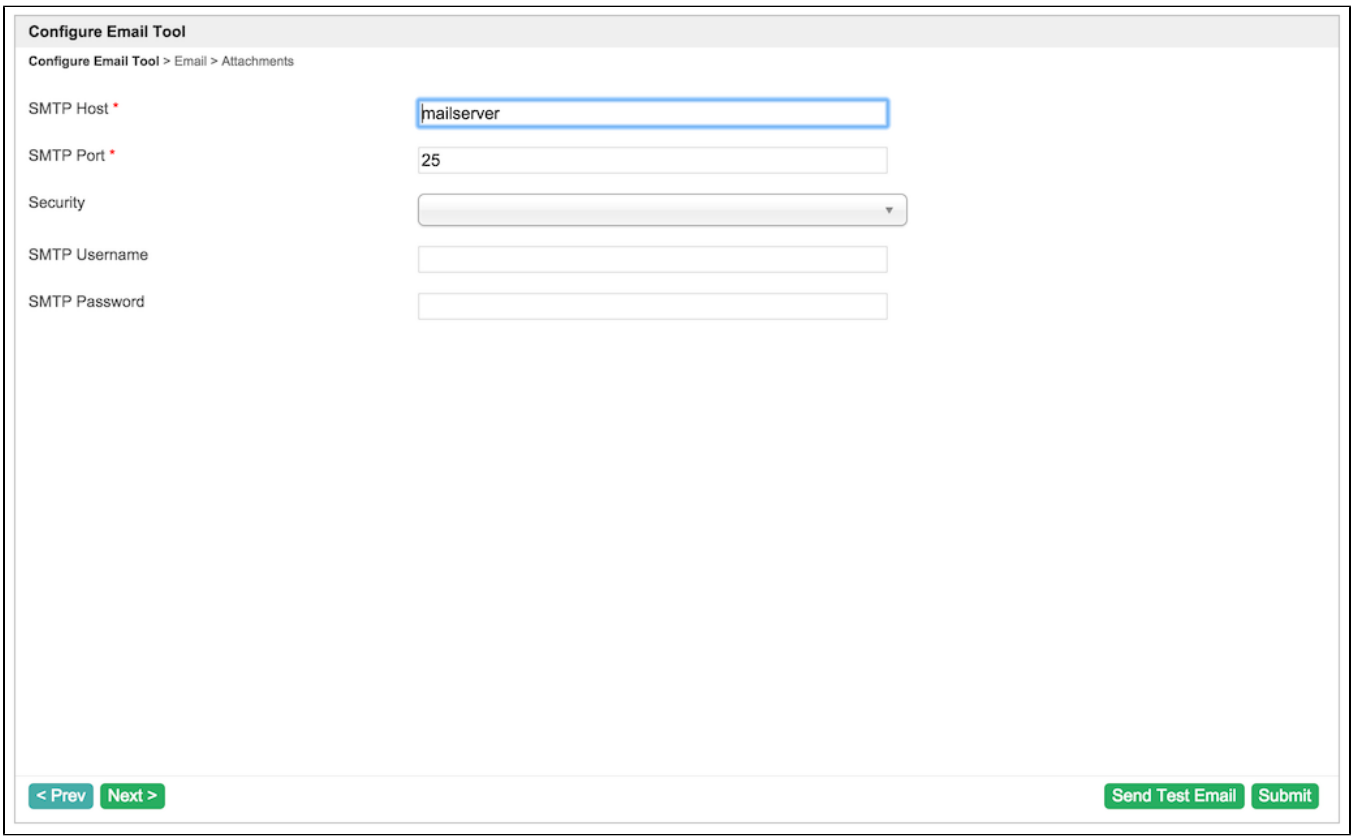
- Plugin Properties Options are in JSON format.

- Plugin Properties Options is an array of Properties Page object

- A Properties Page object has 2 mandatory attributes called "title" and "properties". It also has 2 optional attributes called "validators" and "buttons".

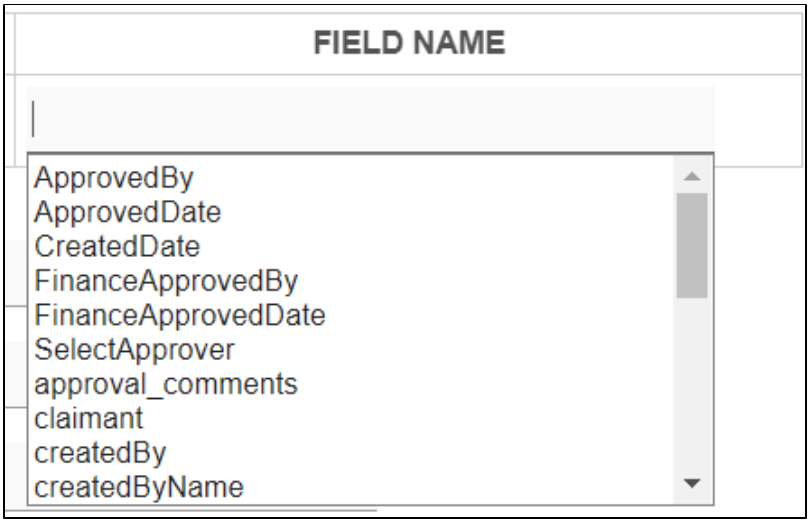
```
[
    {
        title : 'Page Title',
        properties : [
            {
                name : 'Property Name',
                label : 'Property Label',
                description : 'Property Description', //optional, default is NULL
                type : 'Property Type',
                value : 'Property Value', //optional, default is null
                required : 'Mandatory or Not', //optional, 'true' or 'false', default is 'false'
                //… more attributes …
            }, //… more fields …
        ],
        validators : [  //optional
            //… properties custom validators …
        ],
        buttons : [  //optional
            //… custom properties page buttons …
        ]
    }, //… more properties page …
]
```

Sample Look and Feel



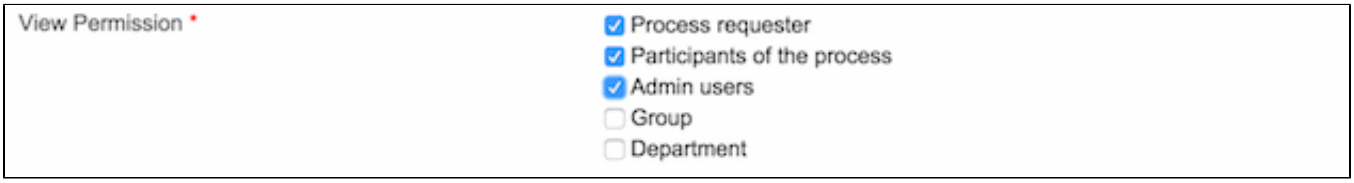
Field Types

Auto Complete



- **type** : 'autocomplete'
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.

Check Box

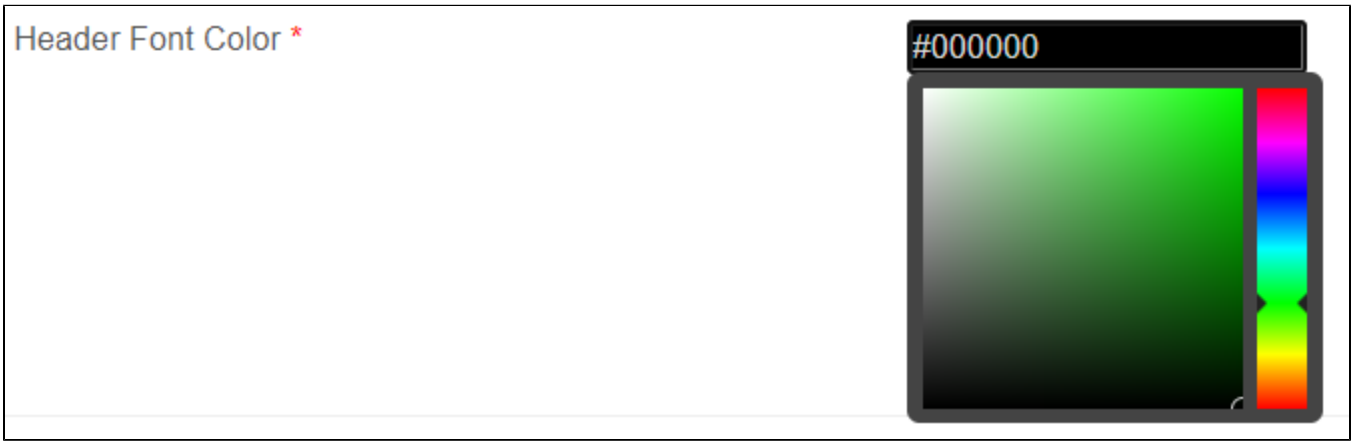


- **type** : 'CheckBox'
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Multi Values Field on how to use the value of this field type in the plugin code.

Code Editor

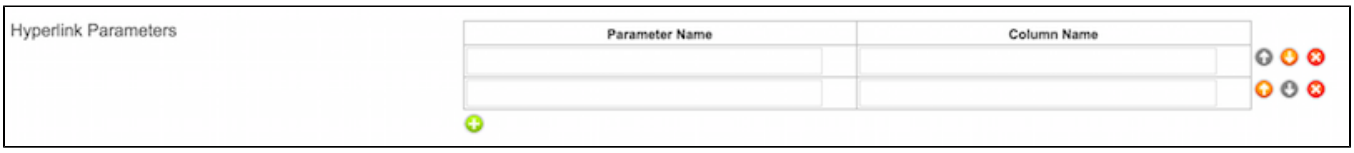
- **type** : 'CodeEditor'
- **mode** : Optional, used for specified highlight mode. Default to 'text', available values are 'text', 'java', 'html', 'javascript', 'css', 'json', 'sql' and 'xml'.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Color



- **type** : 'color'
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.

Combine Grid



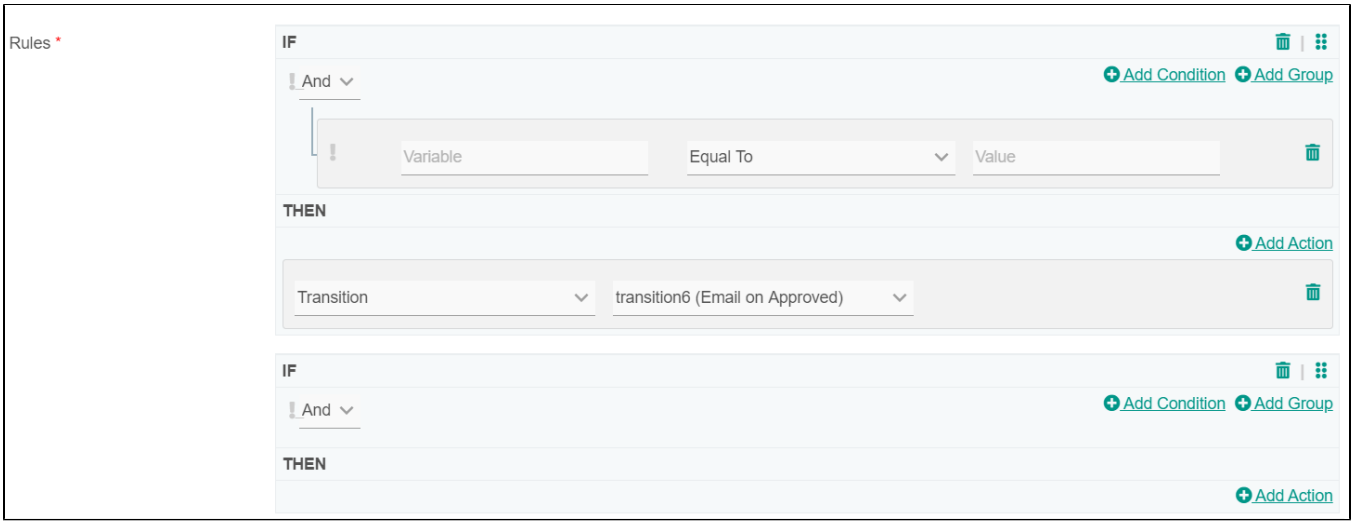
- Combine Grid is used to migrate 2 or more single value property fields from old version plugin (etc Text Field & Select Box) to multi values field in Grid View.
- Combine Grid does not support 'value' attribute.
- **type** : 'GridCombine'
- **columns** : A JSON array of 'column' JSON objects which has 2 mandatory 'key' & 'label' attributes and 2 optional 'required' & 'options' attribute.
    - **key** : Identifier of this column. This value need to be same with the field 'name' that need to migrate from single value field to multiple value field.
    - **label** : Label of the column header
    - **options** : Optional, an array of JSON object with 'value' and 'label' attributes. A column with 'options' attribute will display the input field as select box.
    - **required** : Optional, 'true' or 'false'. A grid cell with the 'required' attribute of 'row' and 'column' set to 'true' value is a mandatory field.

```
columns : [
    {key : 'key', label : 'Columns'},
    {key : 'value', label : 'Value', required: 'true'},
    {key : 'label', label : 'Label', required: 'true'},
    {key : 'width', label : 'Width', options:[
        {value : '10%', label : '10%'},
        {value : '20%', label : '20%'},
        {value : '30%', label : '20%'},
        {value : '40%', label : '20%'}
    ]}
]
```

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Combine Grid Field on how to use the value of this field type in the plugin code.

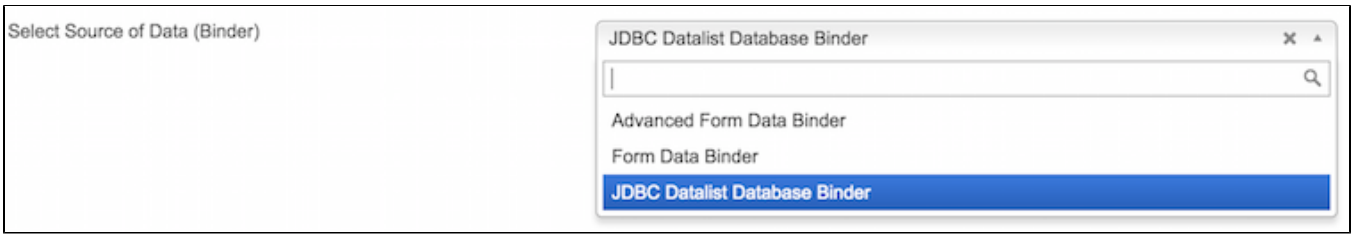
Custom Scripting (New)

New Feature
This is a new feature in Joget DX.



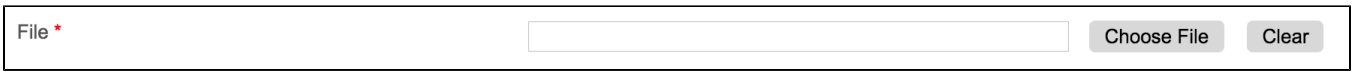
- **type** : 'custom'
- **script_url** : A URL which will return script of the selected element. Built-in URL is "[CONTEXT_PATH]/web/property/json[APP_PATH]/[CLASS_PATH]" which will return the script of a plugin.

Element Select Box



- **type** : 'ElementSelect'
- **url** : A URL which will return Properties Options JSON object of the selected element. Built-in URL is "[CONTEXT_PATH]/web/property/json[APP_PATH]/getPropertyOptions" which will return the Properties Options JSON object of a plugin.
- **keep_value_on_change** : Optional, 'true' or 'false'. Used to decide whether to keep the configuration of the properties options of previous selected element when a new element is selected.
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Usually used for select a plugin and configure the properties of the selected plugin.
- Refer to Retrieve Properties Value in Plugin - ElementSelectBox on how to use the value of this field type in the plugin code.

File



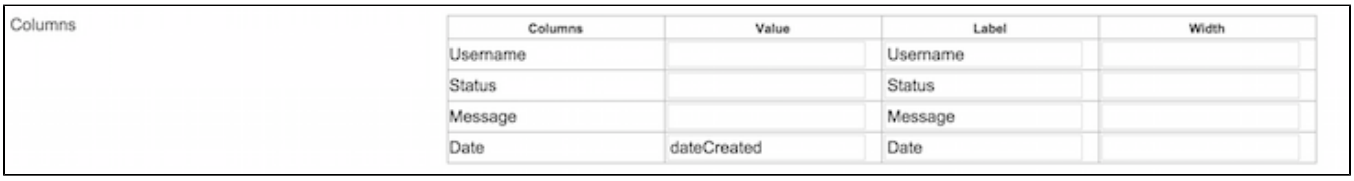
- **type** : 'file',
- **appPath** : '[APP_PATH]',
- **allowInput** : Optional, 'true' or 'false'. Set to 'true' to allow custom link.
- **isPublic** : Optional, 'true' or 'false'. Set to 'true' to auto set the permission to access by anonymous.
- **allowType** : Optional, a string of file extension seperated (;). Example: ".jpeg;.jpg;.gif;.png".

- **maxSize** : Optional, integer value in string format. In kB.
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.

- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Fixed Row Grid



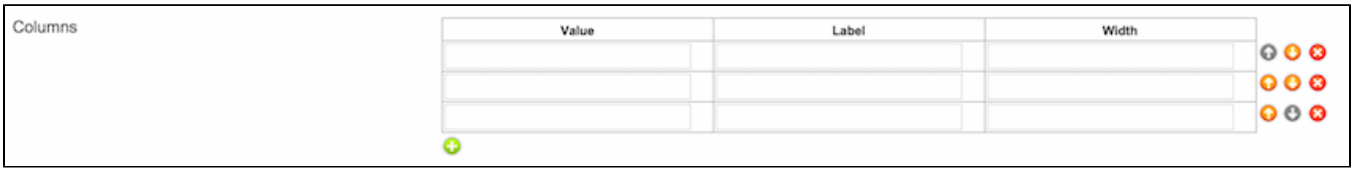
- **type** : 'GridFixedRow'
- **columns** : A JSON array of 'column' JSON objects which has 2 mandatory 'key' & 'label' attributes and 2 optional 'required' & 'options' attribute.
    - **key** : Identifier of this column.
    - **label** : Label of the column header
    - **options** : Optional, an array of JSON object with 'value' and 'label' attributes. A column with 'options' attribute will display the input field as select box.
    - **required** : Optional, 'true' or 'false'. A grid cell with the 'required' attribute of 'row' and 'column' set to 'true' value is a mandatory field.
- **rows** : A JSON array of 'row' JSON Object with 'label' attribute and an optional 'required' attribute. A grid cell with the 'required' attribute of 'row' and 'column' set to 'true' value is a mandatory field.
    - **label** : Label of a row. Used to populate in the first column or every row.
    - **required** : Optional, 'true' or 'false'. A grid cell with the 'required' attribute of 'row' and 'column' set to 'true' value is a mandatory field.
- **value** : A JSON array of grid row values in JSON Object format with all the 'key' attribute of 'column' object used as attribute.

```
columns : [
    {key : 'key', label : 'Columns'}, // first column will used to populate row label
    {key : 'value', label : 'Value', required: 'true'},
    {key : 'label', label : 'Label', required: 'true'},
    {key : 'width', label : 'Width', options:[
        {value : '10%', label : '10%'},
        {value : '20%', label : '20%'},
        {value : '30%', label : '20%'},
        {value : '40%', label : '20%'}
    ]}
],
rows : [
    {label : 'Username', required: 'true'},
    {label : 'Status'},
    {label : 'Message'},
    {label : 'Date'}
],
value : [
    {label : 'Username'},
    {label : 'Status'},
    {label : 'Message', width : '20%'},
    {label : 'Date', value : 'dateCreated'}
]
```

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.

- Refer to Retrieve Properties Value in Plugin - Grid Field on how to use the value of this field type in the plugin code.

Grid



- **type** : 'Grid'
- **columns** : A JSON array of 'column' JSON objects which has 2 mandatory 'key' & 'label' attributes and 2 optional 'required' & 'options' attribute.
    - **key** : Identifier of this column.
    - **label** : Label of the column header
    - **options** : Optional, an array of JSON object with 'value' and 'label' attributes. A column with 'options' attribute will display the input field as select box.
    - **required** : Optional, 'true' or 'false'. A grid cell with the 'required' attribute of 'row' and 'column' set to 'true' value is a mandatory field.
- **value** : A JSON array of grid row values in JSON Object format with all the 'key' attribute of 'column' object used as attribute.

```
columns : [
    {key : 'value', label : 'Value', required: 'true'},
    {key : 'label', label : 'Label', required: 'true'},
    {key : 'width', label : 'Width', options:[
        {value : '10%', label : '10%'},
        {value : '20%', label : '20%'},
        {value : '30%', label : '20%'},
        {value : '40%', label : '20%'}
    ]}
],
value : [
    {label : 'Username', value : 'username'},
    {label : 'Status', value : 'status'},
    {label : 'Message', value : 'message', width : '20%'},
    {label : 'Date', value : 'dateCreated'}
]
```

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Grid Field on how to use the value of this field type in the plugin code.

Header

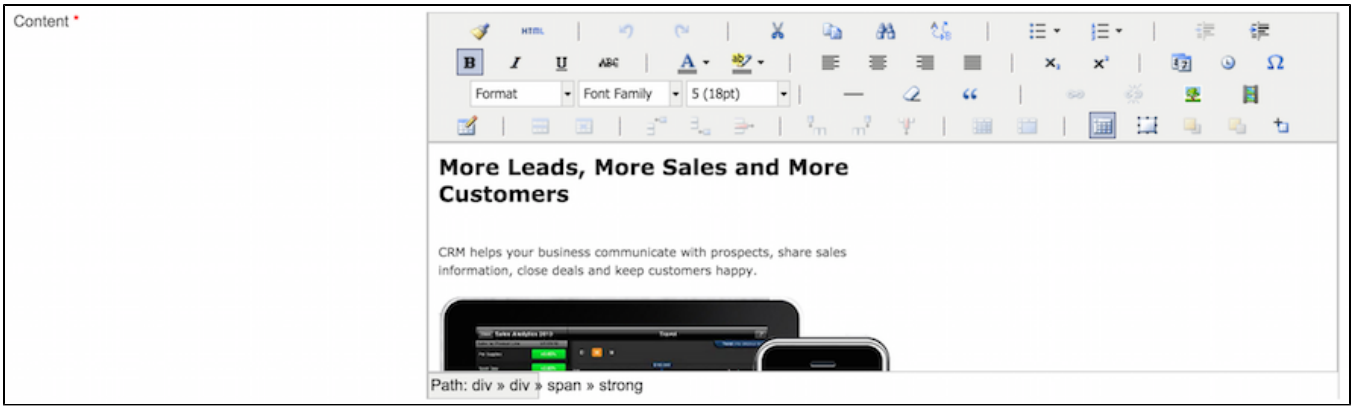
Data

- **type** : 'Header'
- Header does not support 'value' and 'required' attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- This field type is used for separate the fields into different groups. It is not use for capture data.

Hidden Field

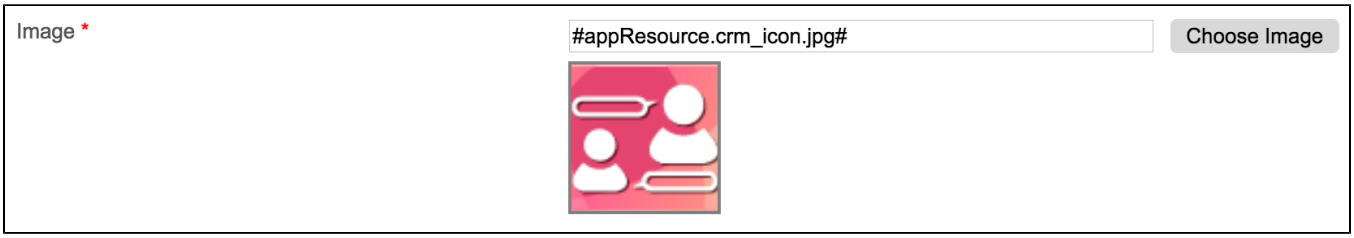
- **type** : 'Hidden'
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

HTML Editor



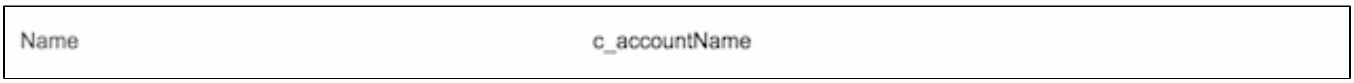
- **type** : 'HtmlEditor'
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Image



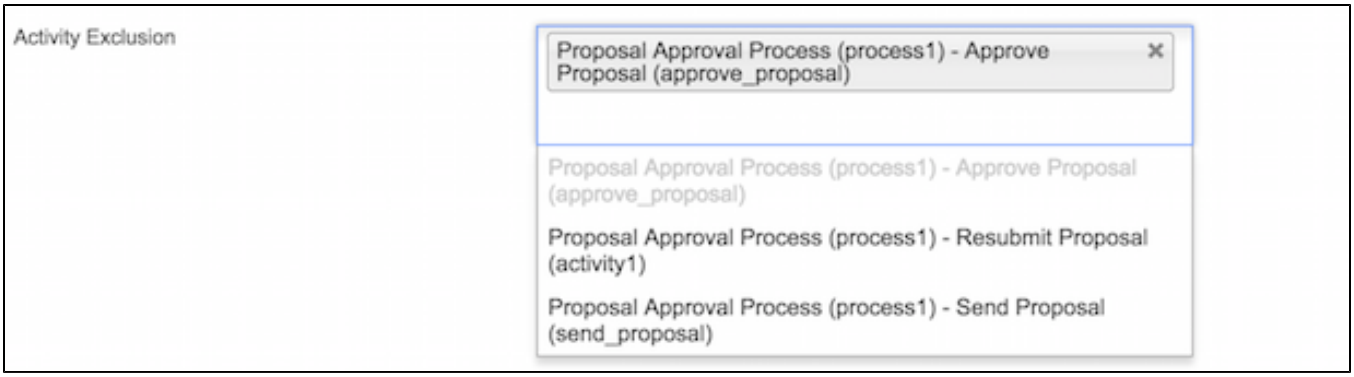
- **type** : 'file',
- **appPath** : '[APP_PATH]',
- **allowInput** : Optional, 'true' or 'false'. Set to 'true' to allow custom link.
- **isPublic** : Optional, 'true' or 'false'. Set to 'true' to auto set the permission to access by anonymous.
- **allowType** : Optional, a string of file extension seperated (;). Example: ".jpeg;.jpg;.gif;.png".
- **maxSize** : Optional, integer value in string format. In kB.
- **imageSize** : Optional, can be integer value in string format or a css expression. Example: '50' or "width:100px;height:70px;".
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Label



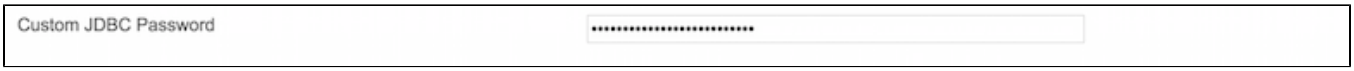
- **type** : 'Label'
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Multi Select Box



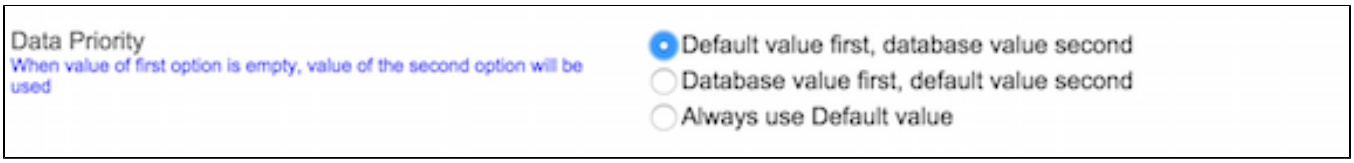
- **type** : 'MultiSelect'
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Multi Values Field on how to use the value of this field type in the plugin code.

Password Field



- **type** : 'Password'
- **size** : Optional, integer value in string format. Default to '50'. Used to control the length of the input field.
- **maxlength** : Optional, integer value in string format. Used to limit the number of characters can be enter in the input field.
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Radio Button



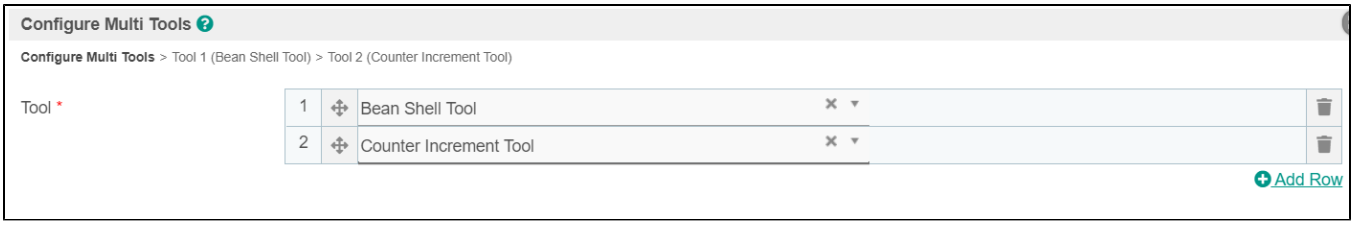
- **type** : 'Radio'
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Readonly Text Field

- **type** : 'Readonly'
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Multiselect in Grid Interface (New)

New Feature
This is a new feature in Joget DX.



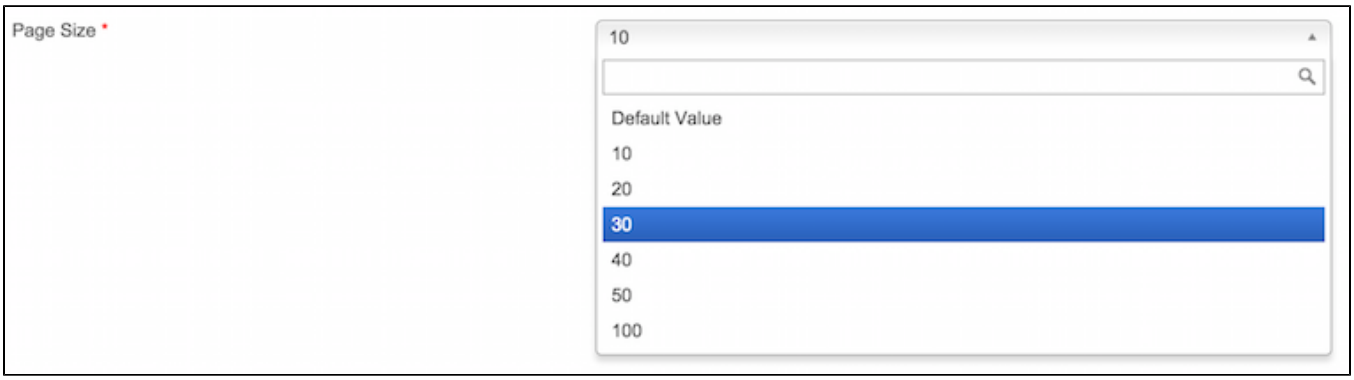
- **type** : 'elementmultiselect'
- Refer to Option Field Attributes for extra attributes.

Number (New)

New Feature
This is a new feature in Joget DX.

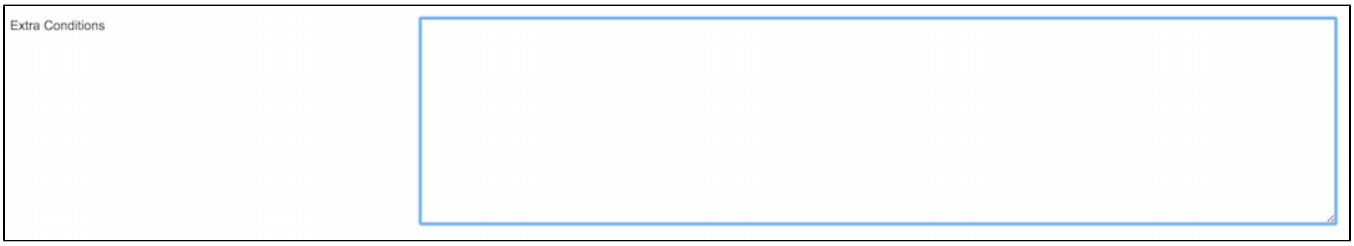
- **type** : 'elementmultiselect'

Select Box



- **type** : 'SelectBox'
- Refer to Option Field Attributes for extra attributes.
- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.
- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Text Area

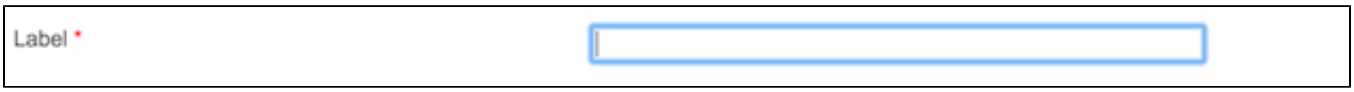


- **type** : 'TextArea'
- rows : Optional, integer value in string format. Default to '5'. Used to control the height of the input field.
- cols : Optional, integer value in string format. Default to '50'. Used to control the length of the input field.
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.

- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Text Field



- **type** : 'TextField'
- **size** : Optional, integer value in string format. Default to '50'. Used to control the length of the input field.
- **maxlength** : Optional, integer value in string format. Used to limit the number of characters can be enter in the input field.
- Refer to Regular Express (Regex) Validation Attributes for extra attributes to do validation using regex.

- Refer to Dependency Field Attributes for extra attributes to do show/hide this field based on other field value.

- Refer to Retrieve Properties Value in Plugin - Single Value Field on how to use the value of this field type in the plugin code.

Regular Express (Regex) Validation Attributes

- The following attributes are designed for Auto Complete, Password Field, Text Area and Text Field.

- **regex_validation** : Optional, regular express in string format.
- **validation_message** : Optional, error message to display when validation failure.

```
{
    regex_validation : '^[a-zA-Z0-9_]+$',
    validation_message : 'Invalid ID!!'
}
```

Dependency Field Attributes

- The following attributes are available for all field types.

- These attributes are used to show/hide a field based on the value of another field.

- The value of a field hidden by these attributes will be ignore during save.

- **control_field** : Optional, 'name' of another field used to control the show/hide of current field.
- **control_value** : Optional, value or regular expression (regex) in string format. This value need to match the value of the controlling field in order to make the field visible.
- **control_use_regex** : Optional, 'true' or 'false'. Default to 'false'. Set to 'true' to use regular expression (regex) in matching the value.

```
{
    control_field: 'chartType',
    control_value: 'bar|xy|area|bubble|line|candlestick|ohlc',
    control_use_regex: 'true',
}
```

Options Field Attributes

- The following attributes are designed for options fields like Auto Complete, Check Box, Element Select Box, Multi Select Box, Radio Button and Select Box.
- You can choose to use one of the following attributes "options", "options_ajax", "options_callback" or "options_script" to populate the options for the field.
- **options :** Optional, an array of JSON object with 'value' and 'label' attributes.

```
options : [
    {value: 'value1', label : 'Value 1'},
    {value: 'value2', label : 'Value 2'},
    {value: 'value3', label : 'Value 3'}
]
```

- **options_ajax :** Optional, a URL which will return an array of JSON object with 'value' and 'label' attributes.

```
options_ajax : '[CONTEXT_PATH]/web/json/console/app[APP_PATH]/datalist/options'
```

- **options_ajax_on_change :** Optional, name of a property field. Used together with 'options_ajax' attribute. The field name and its value will passed as HTTP request parameter to the URL.

```
options_ajax_on_change : 'type'
options_ajax : '[CONTEXT_PATH]/web/json/app[APP_PATH]/plugin/org.joget.plugin.enterprise.SamplePlugin/service?
action=getJson'
```

- **options_callback :** Optional, a javascript function name. All attributes in the field will passed as a single JSON object parameter to this function. The function should return an array of JSON object with 'value' and 'label' attributes.

```
options_callback: 'DatalistBuilder.getColumnOptions'
```

- **options_script :** Optional, a string of javascript which will return an array of JSON object with 'value' and 'label' attributes.

```
options_script: 'var tempArray = [{\'label\':\'\',\'value\':\'\'}];
for(ee in DatalistBuilder.availableColumns){ var temp = {
\'label\' : UI.escapeHTML(DatalistBuilder.availableColumns[ee].label),
\'value\' : DatalistBuilder.availableColumns[ee].id};
tempArray.push(temp);}tempArray;'
```

Built-in JSON API for 'options_ajax'

- **[CONTEXT_PATH]/web/json/console/app[APP_PATH]/forms/options**
  Return all available forms of current app.
- **[CONTEXT_PATH]/web/json/console/app[APP_PATH]/datalist/options**
  Return all available datalists of current app.

- **[CONTEXT_PATH]/web/json/console/app[APP_PATH]/userview/options**
  Return all available userviews of current app.
- **[CONTEXT_PATH]/web/property/json/getElements?classname={plugin interface/abstract class name, optional}**
  Return all available plugins based on the classname filter.
- **[CONTEXT_PATH]/web/json/console/app[APP_PATH]/form/columns/options?formDefId={Form Def Id, optional}&tableName={Form Data tabel name, optional}&tables={Additional form data tables name seperated by (;), optional}&fields={Additional fields to add to result seperated by (;), optional}**
  Return all available fields from form data table

Built-in Javascript Function for 'options_callback'

- **DatalistBuilder.getColumnOptions(properties)**
  Can be used by plugins related to Datalist Builder. It return all available columns based on binder configuration.

Validator Types

- Page validator is execute during change page.
- All fields data in the same page will pass to the validator for validation.

AJAX

- Call to a URL for validation.
- **type** : 'AJAX'
- **url** : An URL return a JSON Object with status (success or fail) & message (JSONArray of String) attribute
- **default_error_message** : Optional. A string of error message.

Page Button

- Page button can be added on the bottom of each page to provide extra feature. Such as send an test email to test the email configuration or make a test connection to database.
- Page button will collect the required fields data from the page and popup dialog and call an AJAX URL.
- **name** : Identifier of this button.
- **label** : Label of the button.
- **ajax_url** : A URL to execute the button action. The URL should return a JSON Object with message (String) attribute.
- **fields** : An array of fields name in the same page that will be used by this button.
- **addition_fields** : An array of Property Field JSON object that will be shown in a popup dialog to collect extra data.
- Example:

```
buttons : [{
    name : 'testmail',
    label : 'Send Test Email',
    ajax_url : '[CONTEXT_PATH]/web/json/app[APP_PATH]/plugin/org.joget.apps.app.lib.EmailTool/service?
action=testmail',
    fields : ['host', 'port', 'security', 'username', 'password'],
    addition_fields : [
        {
            name : 'from',
            label : 'From',
            type : 'textfield',
            required : 'True'
        },
        {
            name : 'toSpecific',
            label : 'To',
            type : 'textfield',
            required : 'True'
        }
```

```
    ]
}]
```

Variable

[CONTEXT_PATH]

- This variable will be replaced by Context Path of current URL.
- Usually used in property attribute value which is URL
- Example : '[CONTEXT_PATH]/web/property/json/getElements?classname=org.joget.apps.form.model.FormValidator'
- Resulted URL : '/jw/web/property/json/getElements?classname=org.joget.apps.form.model.FormValidator'

[APP_PATH]

- This variable will be replaced by Current App Id and App Version of current URL.
- Usually used in property attribute value which is URL
- Example : '[CONTEXT_PATH]/web/json/console/app[APP_PATH]/datalist/options'
- Resulted URL : '/jw/web/json/console/app/crm/3/datalist/options'

Retrieve Properties Value in Plugin

- All the plugin must extends the "org.joget.plugin.base.ExtDefaultPlugin" abstract class. We can use the "Object getProperty(String)" and "String getPropertyString(String)" method to retrieve the configured properties.

Single Value Field

```
String value = getPropertyString("property_name");
```

Multi Values Field

```
String[] values = getPropertyString("property_name").split(";");
```

Combine Grid Field

```
String[] col1_values = getPropertyString("col1_name").split(";");
String[] col2_values = getPropertyString("col2_name").split(";");
```

Grid Field

```
Object columns = getProperty("property_name");
if (columns != null) {
    for (Object colObj : (Object[]) columns) {
        Map col = (Map) colObj;
        String col1_value = (String) opt.get("col1_key");
        String col2_value = (String) opt.get("col2_key");
    }
}
```

Element Select Box

```
import org.joget.plugin.base.PluginManager;
import org.joget.apps.app.service.AppUtil;
import org.joget.plugin.base.ExtDefaultPlugin;

Object element = getProperty("property_name");
if (element != null && element instanceof Map) {
    Map elementMap = (Map) element;
    String className = (String) elementMap.get("className");
    Map<String, Object> properties = (Map<String, Object>) elementMap.get("properties");

    //convert it to plugin
    PluginManager pm = (PluginManager) AppUtil.getApplicationContext().getBean("pluginManager");
    ExtDefaultPlugin plugin = (ExtDefaultPlugin) pm.getPlugin(className);
    if (plugin != null) {
        plugin.setProperties(properties);
    }
}
```