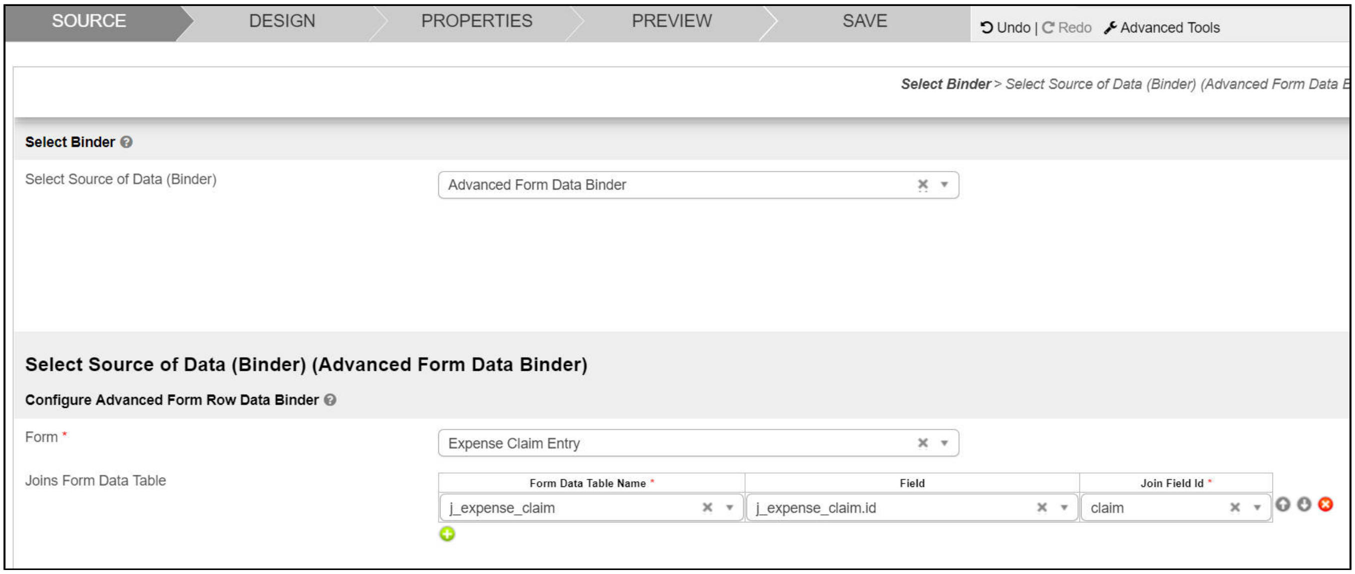
Advanced Form Data Binder

Definition

Advanced Form Row Data Binder is an extended version of the default Form Data Binder. It allows you to add in **Filter Conditions** in a guided and friendly manner.

New Feature

This feature has been enhanced in Joget Workflow v6 to support join, group and aggregate function.



Select Binder *> Select Source of Data (Binder) (Advanced Form Data B*

Select Binder

Select Source of Data (Binder): Advanced Form Data Binder

Select Source of Data (Binder) (Advanced Form Data Binder)

Configure Advanced Form Row Data Binder

Form *: Expense Claim Entry

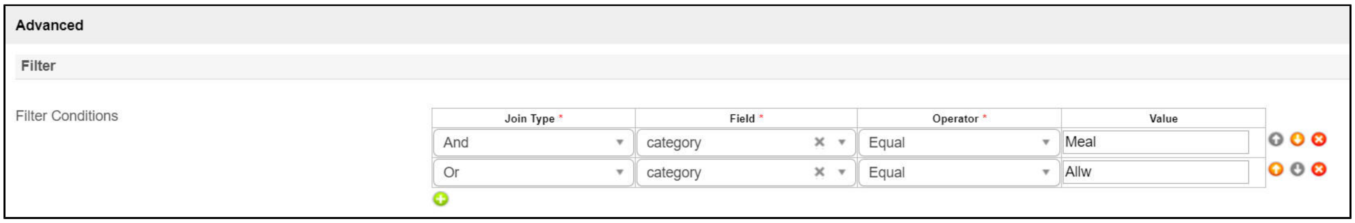
Joins Form Data Table

Form Data Table Name *	Field	Join Field Id *
j_expense_claim	j_expense_claim.id	claim

Figure 1: Advanced Form Data Binder Properties

Name	Description
Form	Source form to retrieve data from.

<table>
<tr>
<td>Joins Form Data Table</td>
<td>
New Feature
This is a new feature in Joget Workflow v6.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Form Data Table Name</td><td>Target table to join with</td></tr>
<tr><td>Field</td><td>Target table field to join with</td></tr>
<tr><td>Join Field Id</td><td>Parent field Id to join with</td></tr>
</table>
Sample
In the screenshot example in Figure 1, such configurations can be presented with the following SQL.
Sample SQL
SELECT * FROM "Claim Entry" entry JOIN hr_expense_claim claim ON claim.id = entry.claim
</td>
</tr>
</table>



Advanced

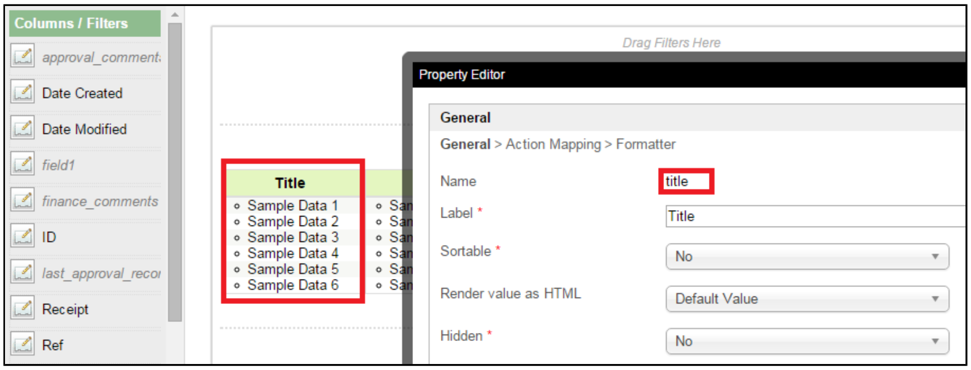
Filter

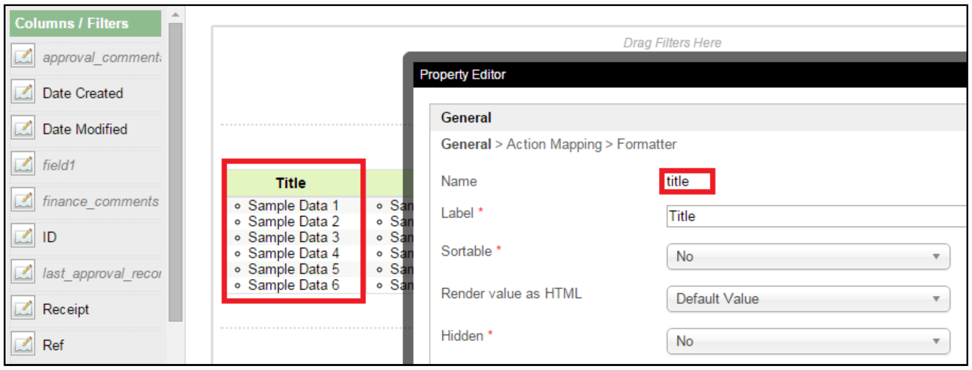
Filter Conditions

Join Type *	Field *	Operator *	Value
And	category	Equal	Meal
Or	category	Equal	Allw

Figure 2: Advanced Form Data Binder Properties - Advanced - Filter

Name	Description

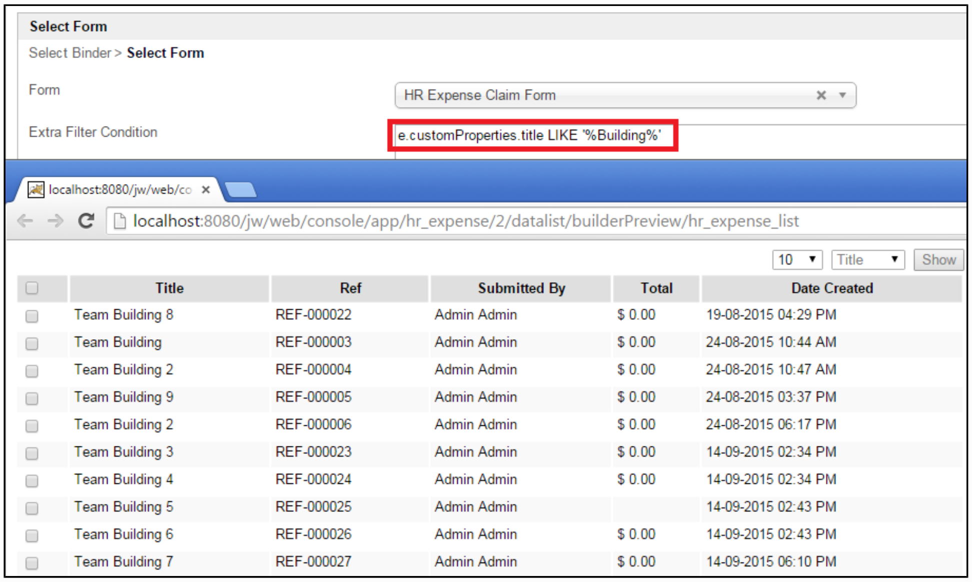
<table>
<tr>
<td>Filter Conditions</td>
<td>Filter Conditions
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Join Type</td><td><ul><li>And</li><li>Or</li></ul></td></tr>
<tr><td>Field</td><td>Field ID. (e.g. title)


Name: title
Label *: Title
Sortable *: No
Render value as HTML: Default Value
Hidden *: No</td></tr>
<tr><td>Operator</td><td><ul><li>Equal</li><li>Not Equal</li><li>Greater Than</li><li>Greater Than Or Equal</li><li>Less Than</li><li>Less Than Or Equal</li><li>Like</li><li>Not Like</li><li>In</li><li>Not In</li><li>Is True</li><li>Is False</li><li>Is Null</li><li>Is Not Null</li></ul></td></tr>
<tr><td>Value</td><td>Filter value</td></tr>
</table>
</td>
</tr>
<tr>
<td>Extra Conditions</td>
<td>Additional condition(s) for filtering the data set. HQL is the expected here.
ⓘ <b>Syntax Query</b>
Start your filter name with <b>e.customProperties.</b> followed by the field id (i.e. <b>title</b>)


Name: title
Label *: Title
Sortable *: No
Render value as HTML: Default Value
Hidden *: No</td>
</tr>
</table>

HQL is accepted

You may even use operator such as "LIKE" to narrow down your data set.



Select Form

Select Binder > **Select Form**

Form	HR Expense Claim Form
Extra Filter Condition	e.customProperties.title LIKE '%Building%'

localhost:8080/jw/web/console/app/hr_expense/2/datalist/builderPreview/hr_expense_list

□	Title	Ref	Submitted By	Total	Date Created
□	Team Building 8	REF-000022	Admin Admin	$ 0.00	19-08-2015 04:29 PM
□	Team Building	REF-000003	Admin Admin	$ 0.00	24-08-2015 10:44 AM
□	Team Building 2	REF-000004	Admin Admin	$ 0.00	24-08-2015 10:47 AM
□	Team Building 9	REF-000005	Admin Admin	$ 0.00	24-08-2015 03:37 PM
□	Team Building 2	REF-000006	Admin Admin	$ 0.00	24-08-2015 06:17 PM
□	Team Building 3	REF-000023	Admin Admin	$ 0.00	14-09-2015 02:34 PM
□	Team Building 4	REF-000024	Admin Admin	$ 0.00	14-09-2015 02:34 PM
□	Team Building 5	REF-000025	Admin Admin		14-09-2015 02:43 PM
□	Team Building 6	REF-000026	Admin Admin	$ 0.00	14-09-2015 02:43 PM
□	Team Building 7	REF-000027	Admin Admin	$ 0.00	14-09-2015 06:10 PM

Sample

```
e.customProperties.title = 'Trip'
```

Hash variable is accepted here.

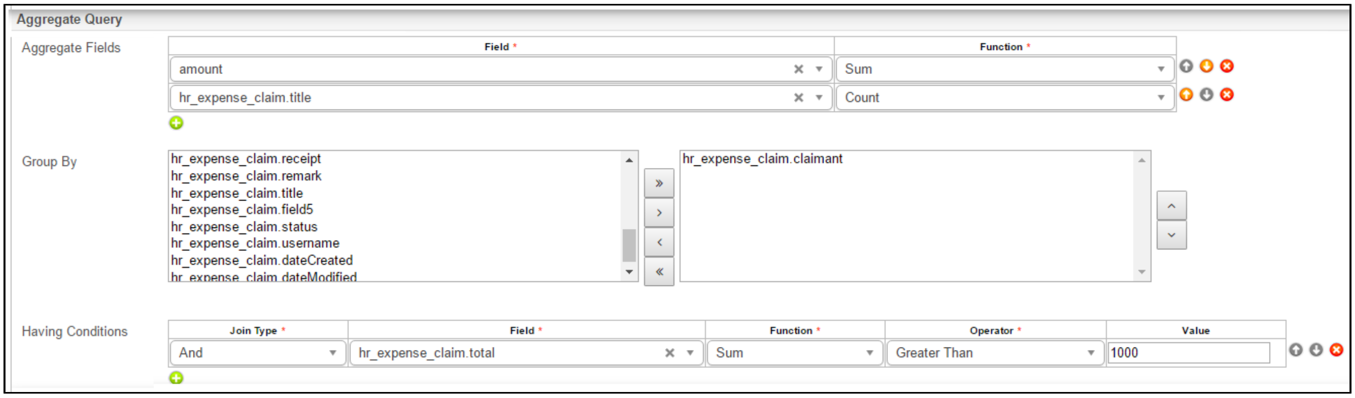
Sample

```
e.customProperties.submitted_by = '#currentUser.id#'
```

Userview Key can be used as part of the condition.

Sample

```
e.customProperties.category_id = '#userviewKey#'
```



Aggregate Query

Aggregate Fields

Field *	Function *
amount	Sum
hr_expense_claim.title	Count

Group By

Available: hr_expense_claim.receipt, hr_expense_claim.remark, hr_expense_claim.title, hr_expense_claim.field5, hr_expense_claim.status, hr_expense_claim.username, hr_expense_claim.dateCreated, hr_expense_claim.dateModified

Selected: hr_expense_claim.claimant

Having Conditions

Join Type *	Field *	Function *	Operator *	Value
And	hr_expense_claim.total	Sum	Greater Than	1000

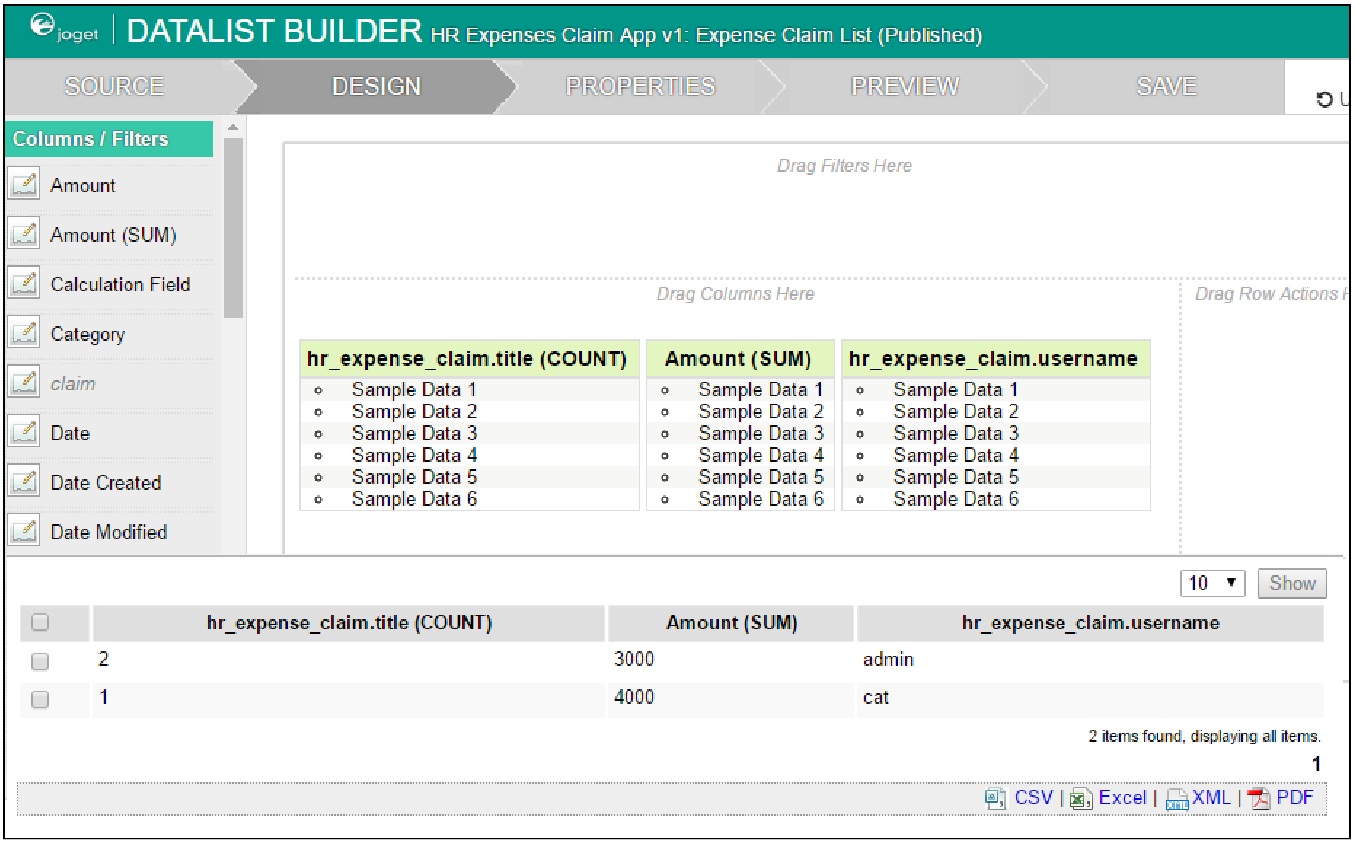
Figure 3: Advanced Form Data Binder Properties - Advanced - Aggregate Query

New Feature

This is a new feature in Joget Workflow v6.

Name	Description
Aggregate Fields	Select field is to aggregated. • Count • Count Distinct • Sum • Min • Max • Avg In the sample screenshot above, the "amount" field will be put into the "Sum" function, and "Count" will be applied to "title".
Group By	Add **grouping** clause/function to the eventual data set. This can be used together with **Aggregate Fields** above. In the sample screenshot above, the "amount" field will be summed up by "claimant", shown in per record row.
Having Conditions	The **HAVING** clause enables you to specify conditions that filter which **group** results appear in the final results. The WHERE clause places conditions on the selected columns, whereas the **HAVING** clause places conditions on **groups** created **by** the **GROUP BY** clause. Read more at http://www.dofactory.com/sql/having

The configurations shown in **Figure 3** will produce the following sample result.

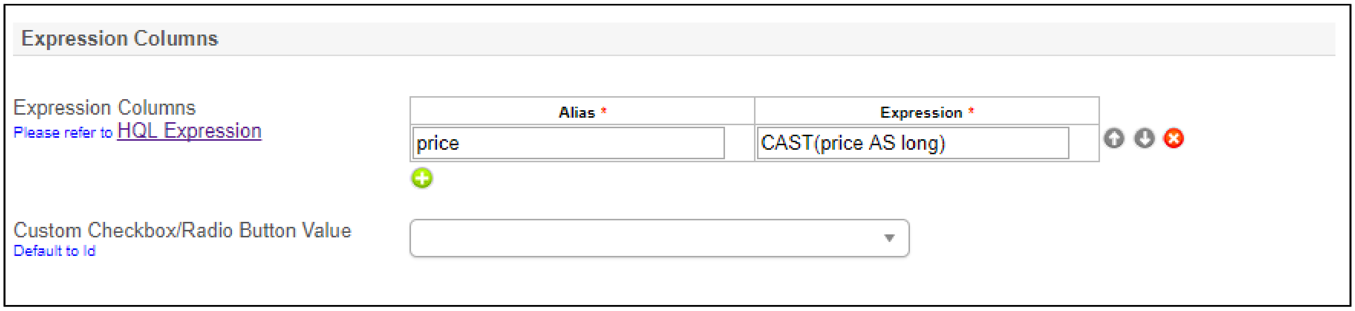


☐	hr_expense_claim.title (COUNT)	Amount (SUM)	hr_expense_claim.username
☐	2	3000	admin
☐	1	4000	cat

2 items found, displaying all items.

1

Figure 4: Sample result



Expression Columns

Expression Columns Please refer to HQL Expression	Alias *	Expression *
	price	CAST(price AS long)
Custom Checkbox/Radio Button Value Default to Id		

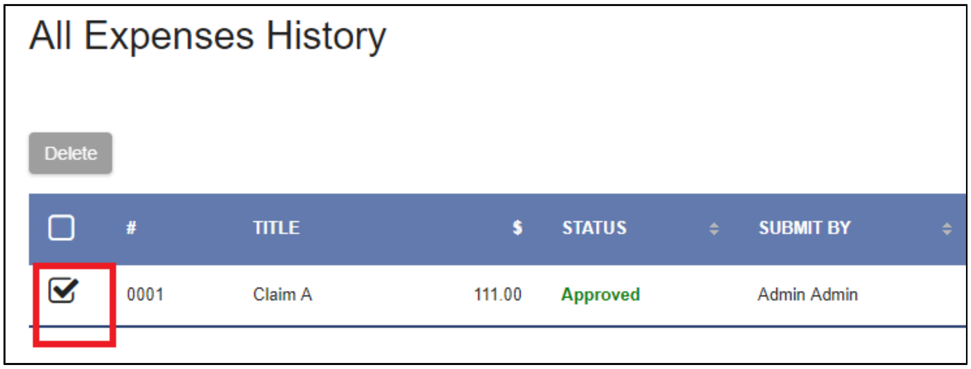
Figure 5: Advanced Form Data Binder Properties - Advanced - Expression Columns

New Feature

This is a new feature in Joget Workflow v6.

Name	Description

Expression Columns	Additional column can be added in this expression columns. This is especially useful when you need to perform additional computation on multiple columns.
	Sample
	`CAST(price AS long)`
	Sample
	`CAST(e.customProperties.sales_price AS long) - CAST(e.customProperties.price AS long)`

Custom Checkbox /Radio Button Value	Define custom record ID to be used to pass over to column action. Defaulted to ID.  All Expenses History Delete ☐ # TITLE $ STATUS SUBMIT BY ☑ 0001 Claim A 111.00 Approved Admin Admin