

List Pending Activity and Assignees

Users may wonder on what is the state of their submitted process applications. We are going to attempt to address this issue by creating a list that will show the **application information together with the pending activity and the pending user**. This way, the requesters/users would be able to tell on the state of their applications/process instances.

In this exercise, we are using the HR Expenses Claim App that is bundled together in the Joget Enterprise edition with MySQL as the database.

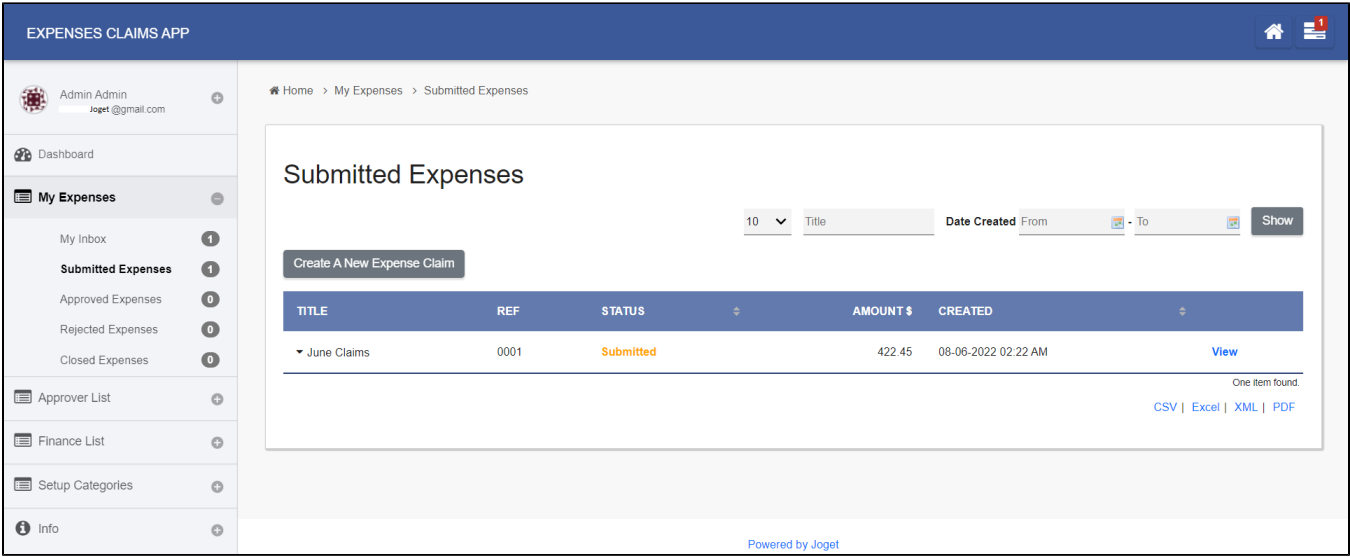


Figure 1: Viewing submitted application through Datalist

By default, users would be able to see the submitted applications by going through the "Personal Expenses" listing but one will not be able to tell what is the next activity in line and who is supposed to attend to it. This can be solved by creating a new Datalist.

1. Create a new Datalist.
2. Choose JDBC Datalist Database Binder.
3. In "Configure JDBC Datalist Database Binder", choose "Default Datasource" in "Datasource".
4. Apply the following query in "SQL SELECT Query"

```
SELECT a.*, sact.Name AS activityName, GROUP_CONCAT(DISTINCT sass.ResourceId SEPARATOR ', ') AS assignee
FROM app_fd_j_expense_claim a
INNER JOIN wf_process_link wpl ON wpl.originProcessId = a.id
INNER JOIN SHKActivities sact ON wpl.processId = sact.ProcessId
JOIN SHKActivityStates ssta ON ssta.oid = sact.State
INNER JOIN SHKAssignmentsTable sass ON sact.Id = sass.ActivityId
WHERE ssta.KeyValue = 'open.not_running.not_started'
GROUP BY a.id
```

Note: Please replace the code "app_fd_hr_expense_claim" with your own table name if you intend to use it for other application.

5. Set "Primary Key" to "a.id"
6. Click OK.

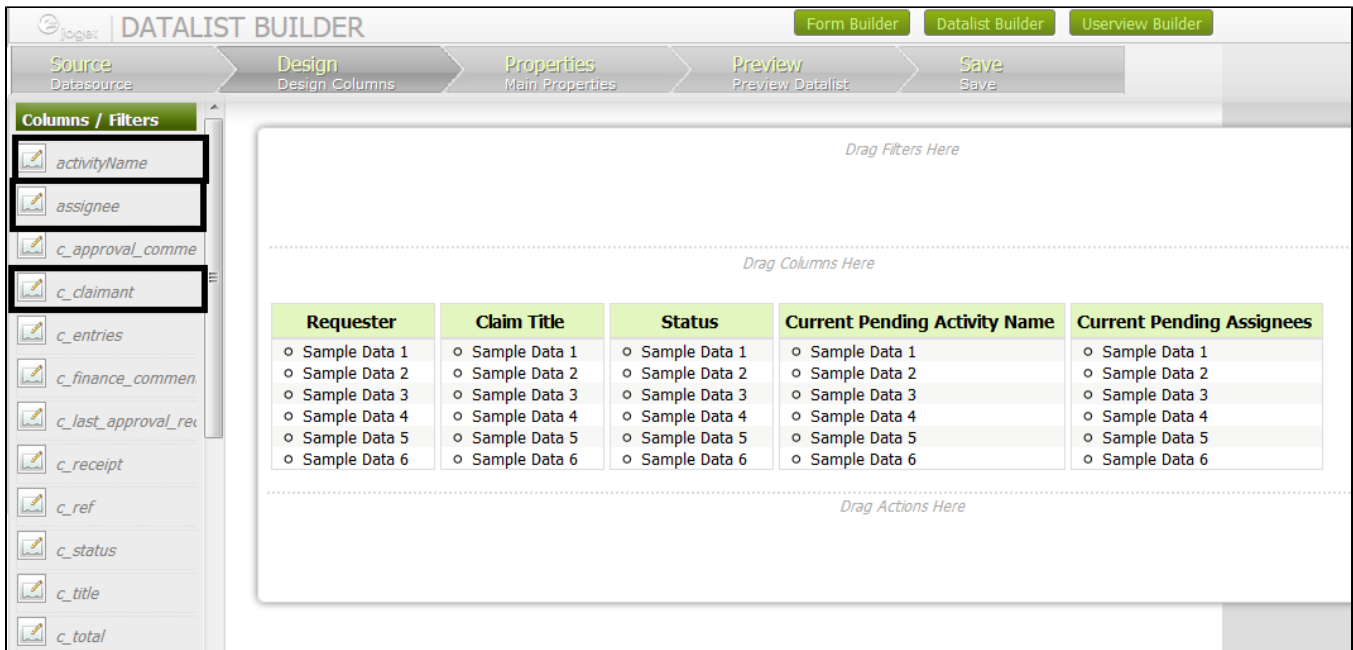


Figure 2: Adding the columns into the Datalist

Next, add in the columns intended, and most importantly, add "activityName" and "assignee" to reveal the pending activity and assignees.

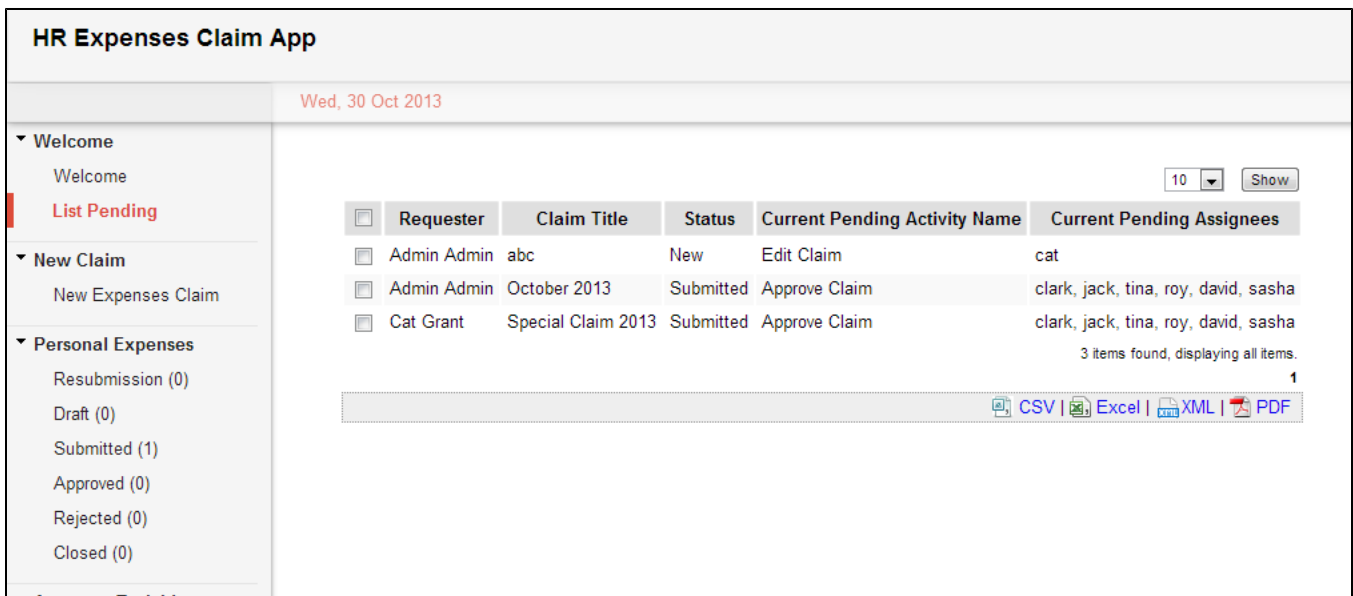


Figure 3: Datalist showing the pending activity and assignees

The Datalist will now list all the pending activities of everyone. Next, we are going filter the list such that user will only see what they submitted.

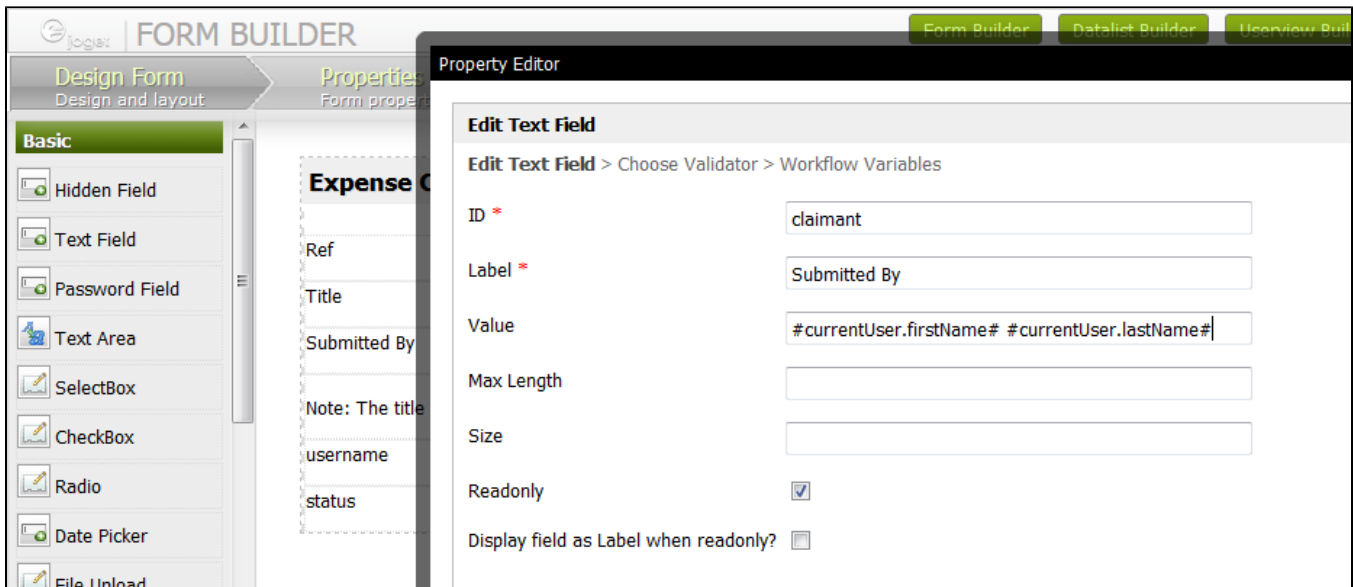


Figure 4: Retrieving the requester information

We may determine on who is the claimant by looking up the "claimant" field.

In the Datalist's "SQL SELECT Query", modify the code to the following:-

```
SELECT a.*, sact.Name AS activityName, GROUP_CONCAT(DISTINCT sass.ResourceId SEPARATOR ', ') AS assignee
FROM app_fd_hr_expense_claim a
JOIN SHKActivities sact on a.id = sact.ProcessId
JOIN SHKActivityStates ssta ON ssta.oid = sact.State
INNER JOIN SHKAssignmentsTable sass ON sact.Id = sass.ActivityId
WHERE ssta.KeyValue = 'open.not_running.not_started' AND a.c_claimant = '#currentUser.firstName# #currentUser.lastName#'
GROUP BY a.id
```

This is the new code added in.

```
AND a.c_claimant = '#currentUser.firstName# #currentUser.lastName#'
```

With the changes made above, we will now be able to list down the records related to the currently logged in user.

						10	Show
	Requester	Claim Title	Status	Current Pending Activity Name	Current Pending Assignees		
<input type="checkbox"/>	Admin Admin	abc	New	Edit Claim	cat		
<input type="checkbox"/>	Admin Admin	October 2013	Submitted	Approve Claim	clark, jack, tina, roy, david, sasha		
						2 items found, displaying all items.	1
						<input type="button" value="CSV"/> <input type="button" value="Excel"/> <input type="button" value="XML"/> <input type="button" value="PDF"/>	

Figure 5: Filtered List of Pending Activity and Assignee

Additional Information:

- The following query is for MSSQL to use.

```

SELECT dat.*, asg.activityName, asg.assignees FROM (SELECT id, activityName, assignees from
(SELECT a.id, sact.Name AS activityName, sass.ResourceId AS assignee
FROM app_fd_applications a
JOIN SHKActivities sact on a.id = sact.ProcessId
JOIN SHKActivityStates ssta ON ssta.oid = sact.State
INNER JOIN SHKAssignmentsTable sass ON sact.Id = sass.ActivityId
WHERE ssta.KeyValue = 'open.not_running.not_started'
group by sact.Name, sass.ResourceId, a.id)
AS A CROSS APPLY

(SELECT assignee + ',' FROM
(SELECT a.id, sact.Name AS activityName, sass.ResourceId AS assignee
FROM app_fd_applications a
JOIN SHKActivities sact on a.id = sact.ProcessId
JOIN SHKActivityStates ssta ON ssta.oid = sact.State
INNER JOIN SHKAssignmentsTable sass ON sact.Id = sass.ActivityId
WHERE ssta.KeyValue = 'open.not_running.not_started'
group by sact.Name, sass.ResourceId, a.id)
AS B WHERE A.id = B.id AND A.activityName = B.activityName FOR XML PATH('')) D (assignees) GROUP BY id,
activityName, assignees
) asg JOIN app_fd_applications dat ON asg.id = dat.id

```