



# Joget DX

# Doing More with Your Process Design

# Prerequisites

1. Knowledge about designing process using the Process Builder

# Content

1. Process Flow Design Considerations
2. Type of Routes
3. Participant Mapping using Workflow Variable
4. Mapping Routes Without Workflow Variable
5. Deadline using Workflow Variable
6. Complex Multiple Approval Based on Form Data
7. Process Enhancement Plugin
8. Process Completion Data Handling

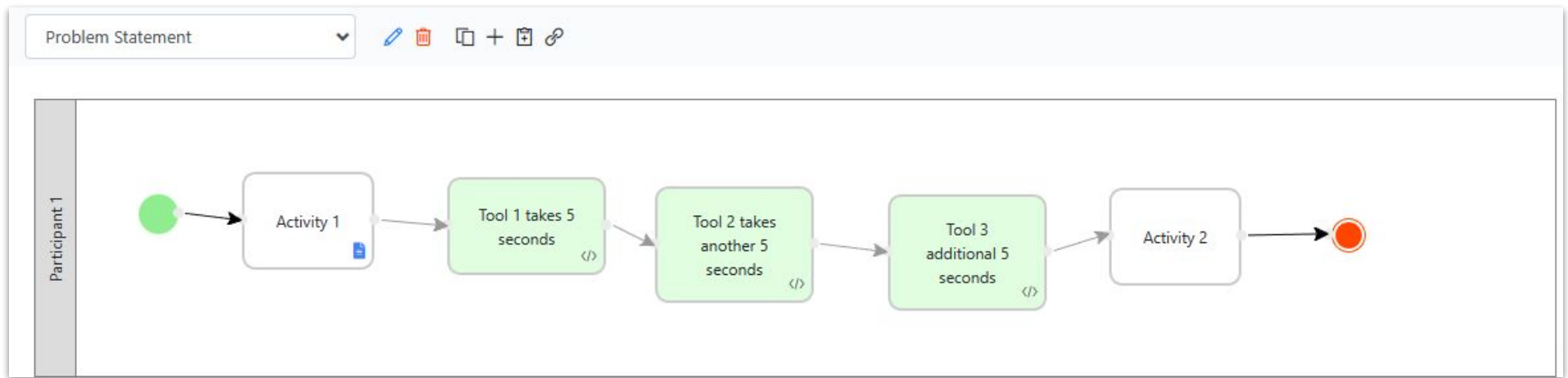


# Chapter 1

## Process Flow Design Considerations

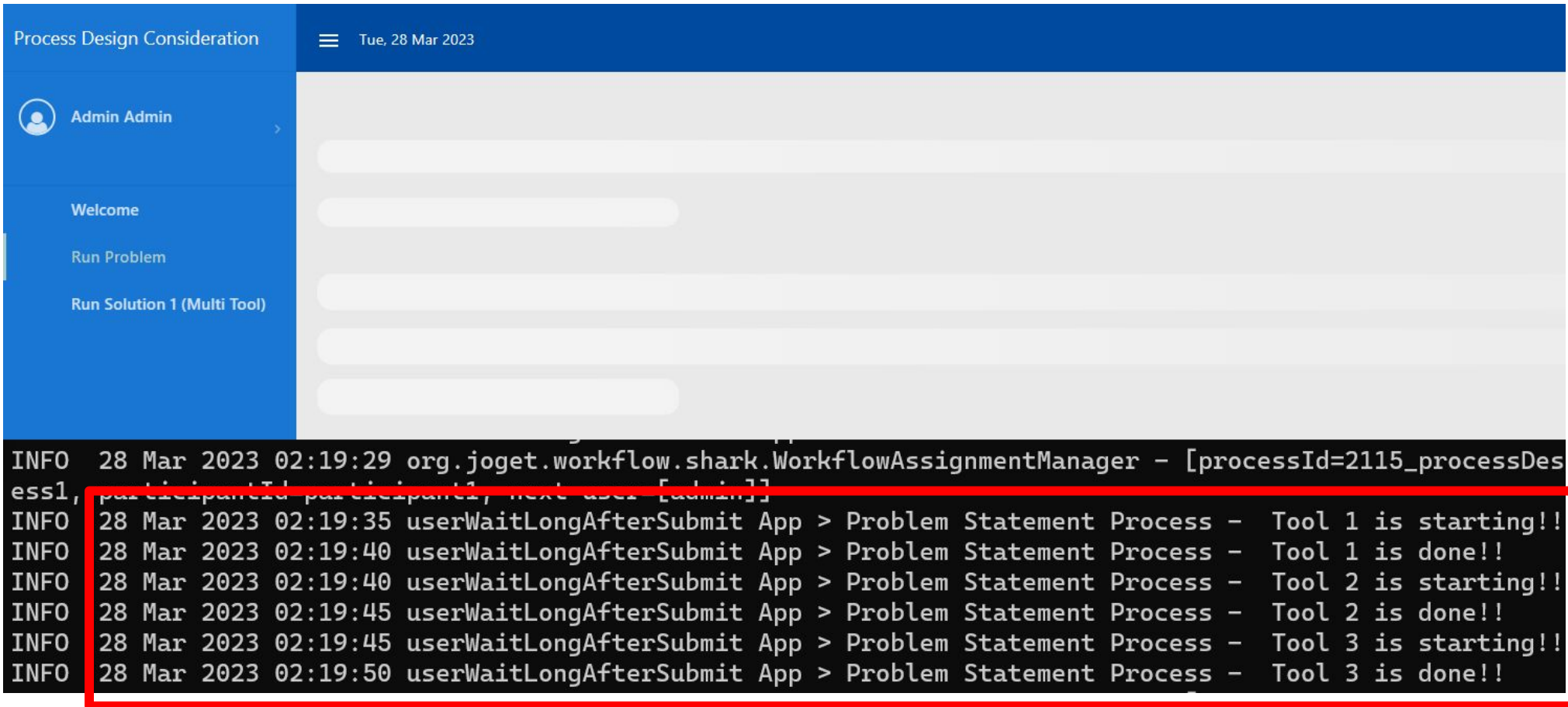
# Problem Statement

- Discuss various process designs with performance and user experience at the back of our minds.



# Problem Statement

- User will have to wait until all tools are finished



Process Design Consideration

Tue, 28 Mar 2023

Admin Admin

Welcome

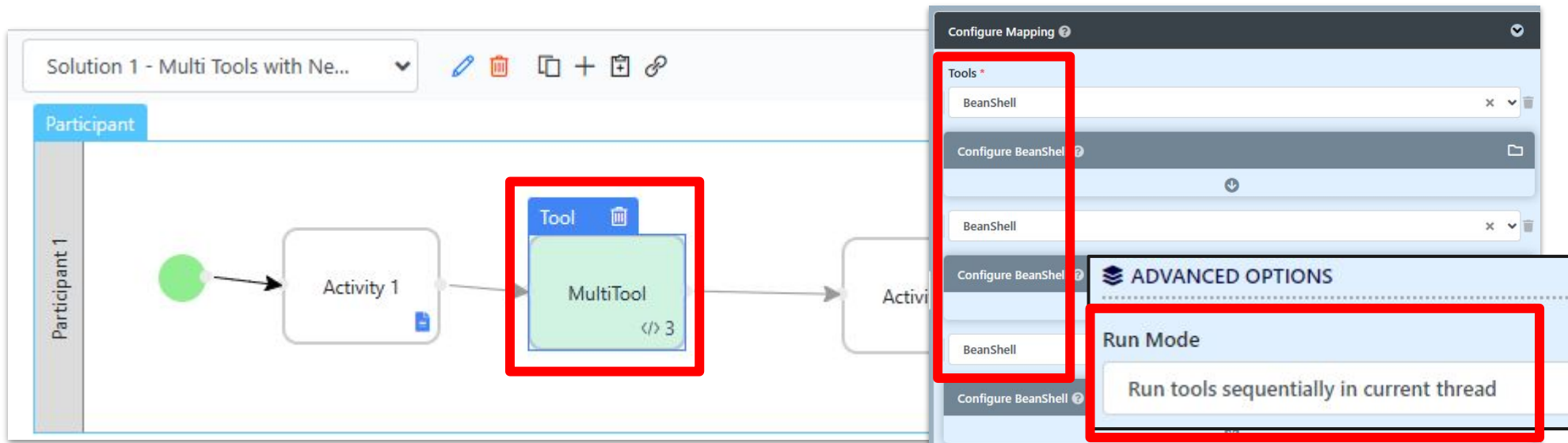
Run Problem

Run Solution 1 (Multi Tool)

```
INFO 28 Mar 2023 02:19:29 org.joget.workflow.shark.WorkflowAssignmentManager - [processId=2115_processDes
ess1, participantId=participant1, next user [admin]]
INFO 28 Mar 2023 02:19:35 userWaitLongAfterSubmit App > Problem Statement Process - Tool 1 is starting!!
INFO 28 Mar 2023 02:19:40 userWaitLongAfterSubmit App > Problem Statement Process - Tool 1 is done!!
INFO 28 Mar 2023 02:19:40 userWaitLongAfterSubmit App > Problem Statement Process - Tool 2 is starting!!
INFO 28 Mar 2023 02:19:45 userWaitLongAfterSubmit App > Problem Statement Process - Tool 2 is done!!
INFO 28 Mar 2023 02:19:45 userWaitLongAfterSubmit App > Problem Statement Process - Tool 3 is starting!!
INFO 28 Mar 2023 02:19:50 userWaitLongAfterSubmit App > Problem Statement Process - Tool 3 is done!!
```

# Solution 1 – Multi Tools with New Thread (No Dependency on Tools)

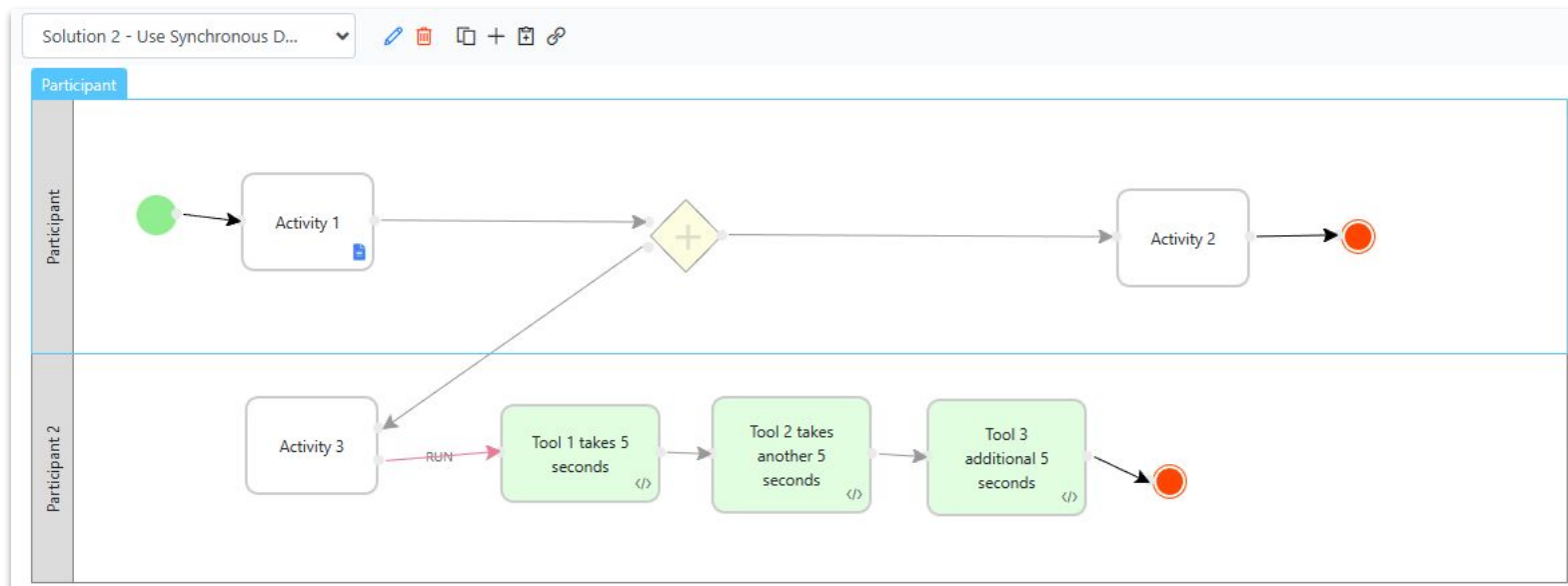
- Use Multi Tools and have all the tools placed under it. It is important to set the **Run Mode** to **Run tools sequentially in a new single thread**.
- Since the tools will run in a new thread, Joget will not wait for it to complete and move on to Activity 2.



The screenshot displays the Joget Designer interface. On the left, a workflow diagram for 'Participant 1' shows a sequence: a start node (green circle) leads to 'Activity 1', which then leads to a 'MultiTool' component. The 'MultiTool' component is highlighted with a red box. To the right, the 'Configure Mapping' dialog is open. It shows a list of tools under the 'Tools \*' section, with 'BeanShell' selected. Below this, the 'ADVANCED OPTIONS' section is visible, and the 'Run Mode' is set to 'Run tools sequentially in current thread', which is also highlighted with a red box.

# Solution 2 – Use Synchronous Deadline (No Dependency on Tools)

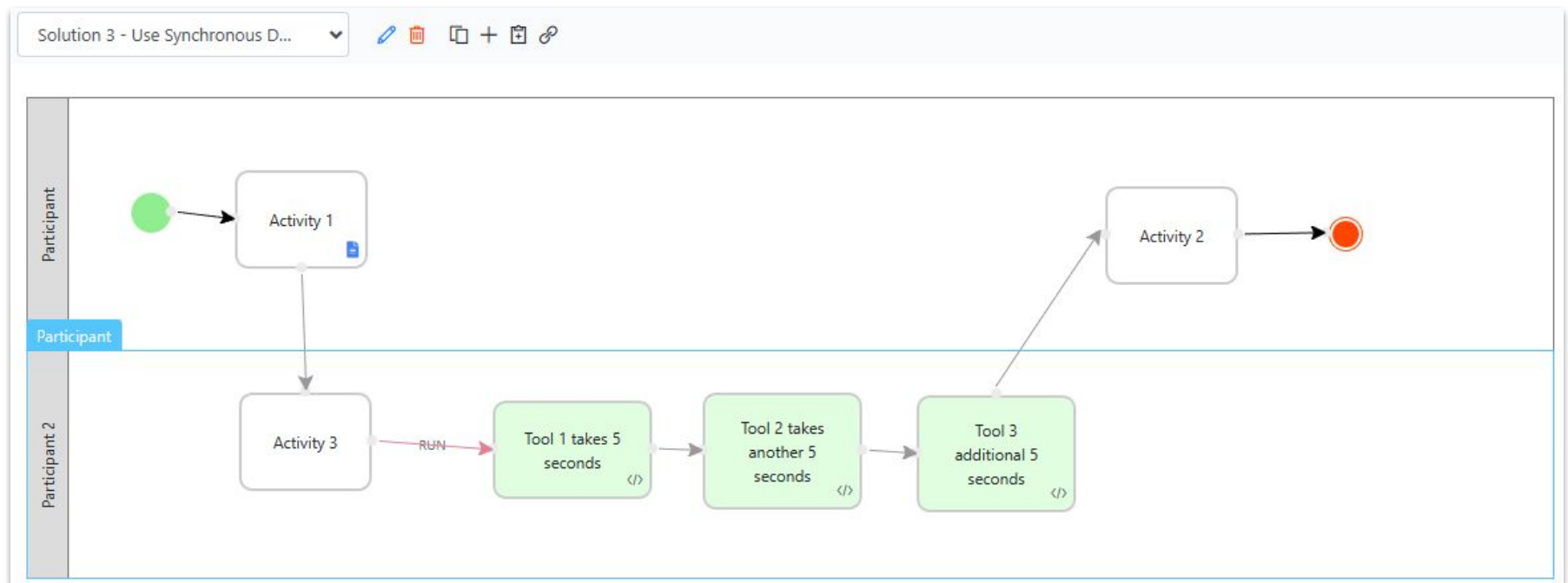
- In this solution, we are adding a parallel route and a new activity (Activity 3).
- Activity 3 is used as a dummy activity and mapped to a dummy user to serve as a "checkpoint" so that users do not need to wait at the "Please wait..." screen.





# Solution 3 – Use Synchronous Deadline (With Dependency on Tools)

- By retaining the synchronous deadline in Solution 1 earlier, upon completing the assignment Activity 1, users will be able to inbox listing or wherever they came from and continue their work



# References

- <https://dev.joget.org/community/display/DX8/Process+Flow+Design+Considerations+for+Performance+and+User+Experience>

# Materials

- A sample app for the multitool is available in the **Materials** folders as **19.1.jwa**

# Chapter Review

1. Understand the process flow design considerations

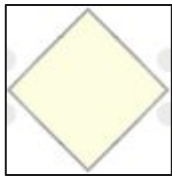


# Chapter 2

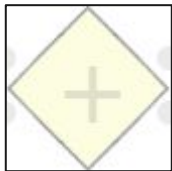
## Type of Routes

# Routes

- Route is used to split or merge a process flow
- The 2 types of routes supported by Process Builder are:
  - Exclusive (XOR)

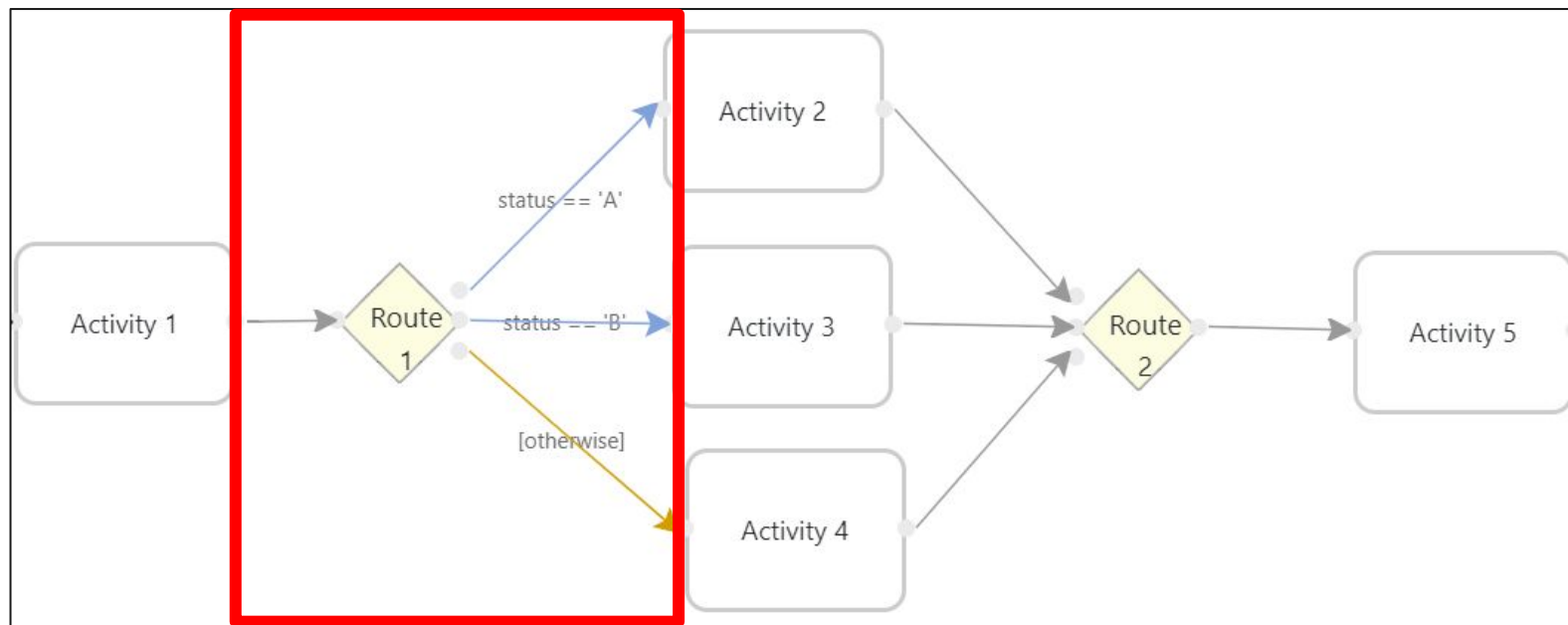


- Parallel (AND)



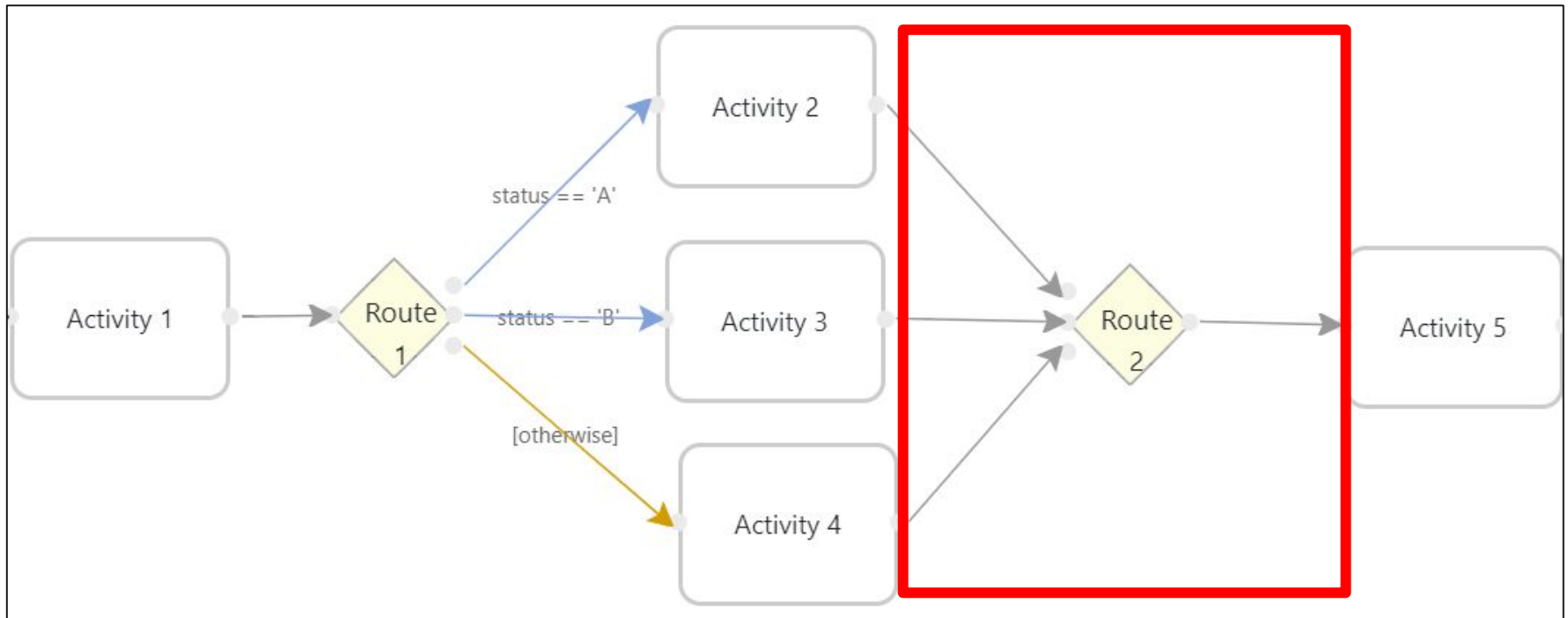
# Exclusive Route (XOR)

- After “Activity 1”, only 1 path will be taken, **either** Activity 2, 3, or 4, based on matching condition



# Exclusive Route (XOR)

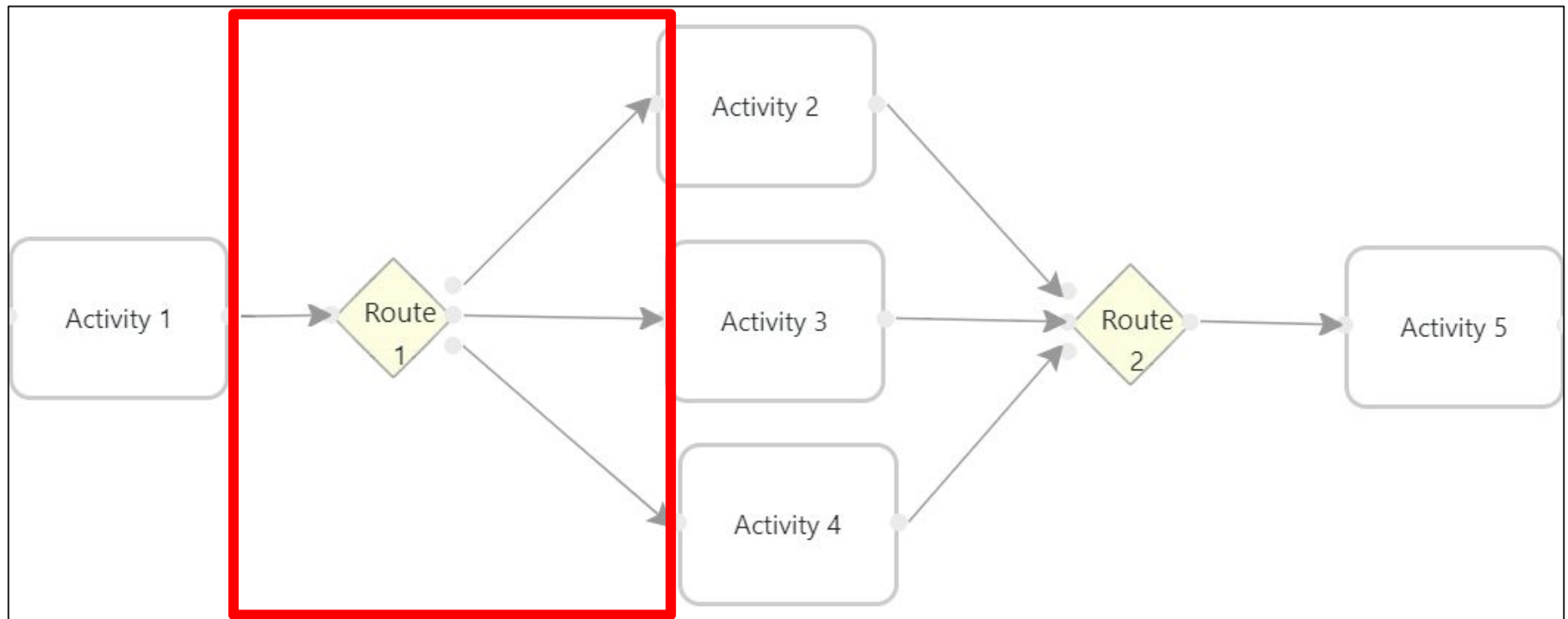
- After completing either one of the activities, Activity 5 will be executed



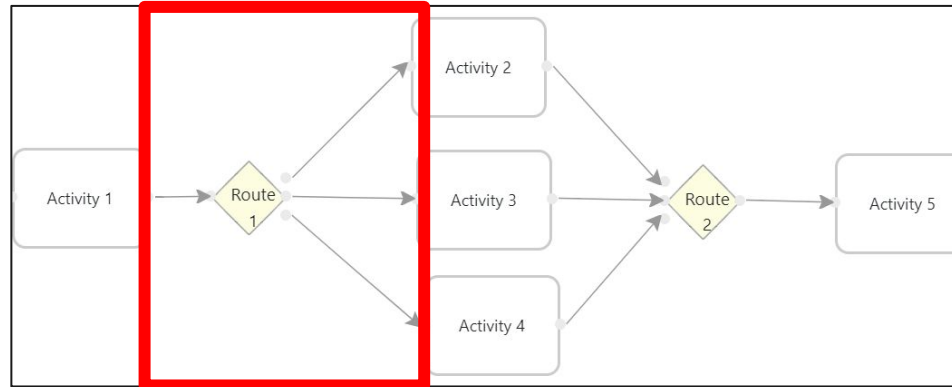


# Exclusive Route (XOR)

- What if there's no condition set at all?



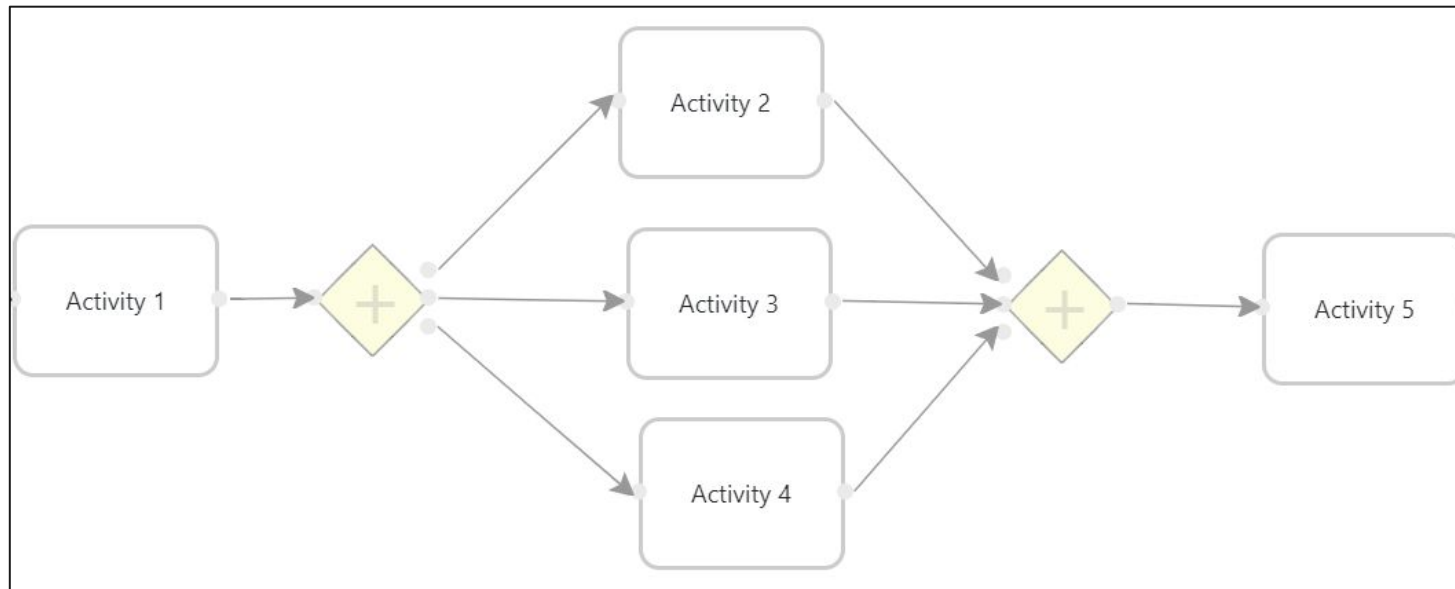
# Exclusive Route (XOR)



- FIRST matchable condition takes precedence, even when multiple routes are TRUE
- If NO matchable, FIRST default condition takes precedence (e.g: blank condition)
- If all transitions are NOT true and NO 'otherwise' condition exist, instance now stuck at "Route 1" and instance is marked as CLOSED & COMPLETED

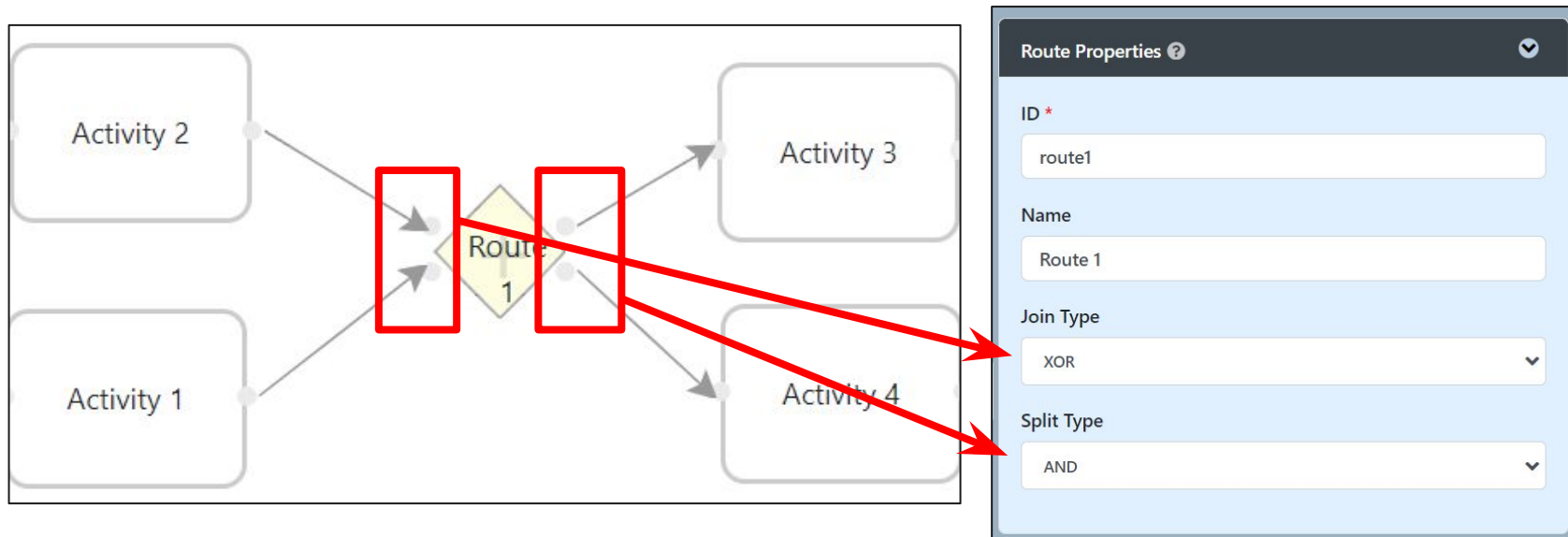
# Parallel Route (AND)

- After “Activity 1”, **ALL** matching conditions OR defaults (next 3 activities) will be executed “simultaneously”
- “Activity 5” will not be performed until Activity 2, 3, and 4 are completed



# Discussion

- List down all combinations of Join and Split type and how it would affect the process flow



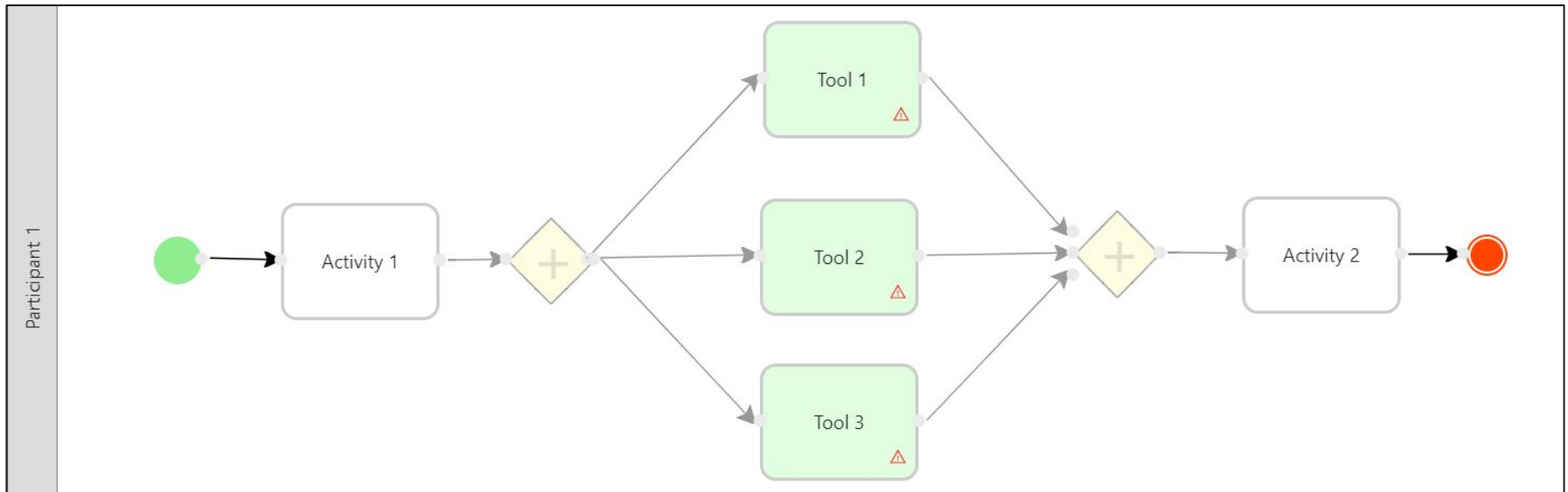
# Discussion

Only 4 possible combinations (Split – Join):

1. XOR – XOR
2. XOR – AND
3. AND – AND
4. AND – XOR

# Discussion

- If we set 3 Tools to run parallel, will all 3 tools be executed simultaneously?



# Discussion

- Executed “simultaneously” is actually a misnomer, it actually runs sequentially

```
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 Start===== 2020-01-30 at 15:20:23 SGT
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 =0
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 =1
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 =2
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 =3
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 =4
INFO 30 Jan 2020 15:20:23 Tool 1 - Tool 1 End===== 2020-01-30 at 15:20:23 SGT
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 Start===== 2020-01-30 at 15:20:23 SGT
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 =0
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 =1
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 =2
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 =3
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 =4
INFO 30 Jan 2020 15:20:23 Tool 2 - Tool 2 End===== 2020-01-30 at 15:20:23 SGT
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 Start===== 2020-01-30 at 15:20:24 SGT
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 =0
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 =1
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 =2
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 =3
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 =4
INFO 30 Jan 2020 15:20:24 Tool 3 - Tool 3 End===== 2020-01-30 at 15:20:24 SGT
```

# Not parallel, then what?

- If you want to run it concurrently, you will need to use the multi tool plugin and set to multithread

```
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 Start===== 2020-01-30 at 15:24:28 SGT
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 Start===== 2020-01-30 at 15:24:28 SGT
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 Start===== 2020-01-30 at 15:24:28 SGT
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =0
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =1
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =2
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =3
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =0
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =4
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =5
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =0
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =1
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =2
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =3
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =4
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =5
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =6
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =7
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =8
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 =9
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =1
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =6
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =7
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =8
INFO 30 Jan 2020 15:24:28 Multi Tool 2 - Multi Tool 2 End===== 2020-01-30 at 15:24:28 SGT
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =2
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 =9
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =3
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =4
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =5
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =6
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =7
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =8
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 =9
INFO 30 Jan 2020 15:24:28 Multi Tool 3 - Multi Tool 3 End===== 2020-01-30 at 15:24:28 SGT
INFO 30 Jan 2020 15:24:28 Multi Tool 1 - Multi Tool 1 End===== 2020-01-30 at 15:24:28 SGT
```

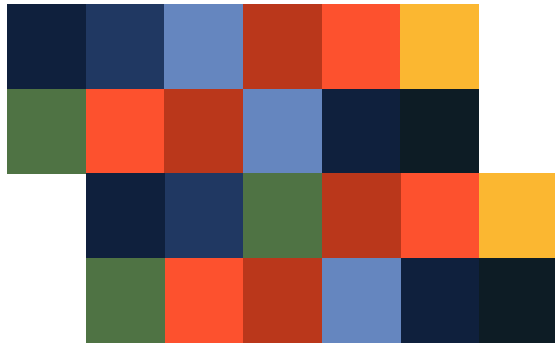


# Materials

- A sample app for the multitool is available in the **Materials** folders as **19.2.jwa**

# Chapter Review

1. Understand the use of Route and its configurations



# Chapter 3

## Participant Mapping using Workflow Variable

# Participant Mapping Using Workflow Variable

- Dynamically map to participant using workflow variable.
- Workflow variable may consists of:-
  1. Group code
  2. Username
  3. Department code
  4. Head of Department (Department code)
- Supports multiple assignments. (Semicolon separated value [Eg. jack;david;clark])

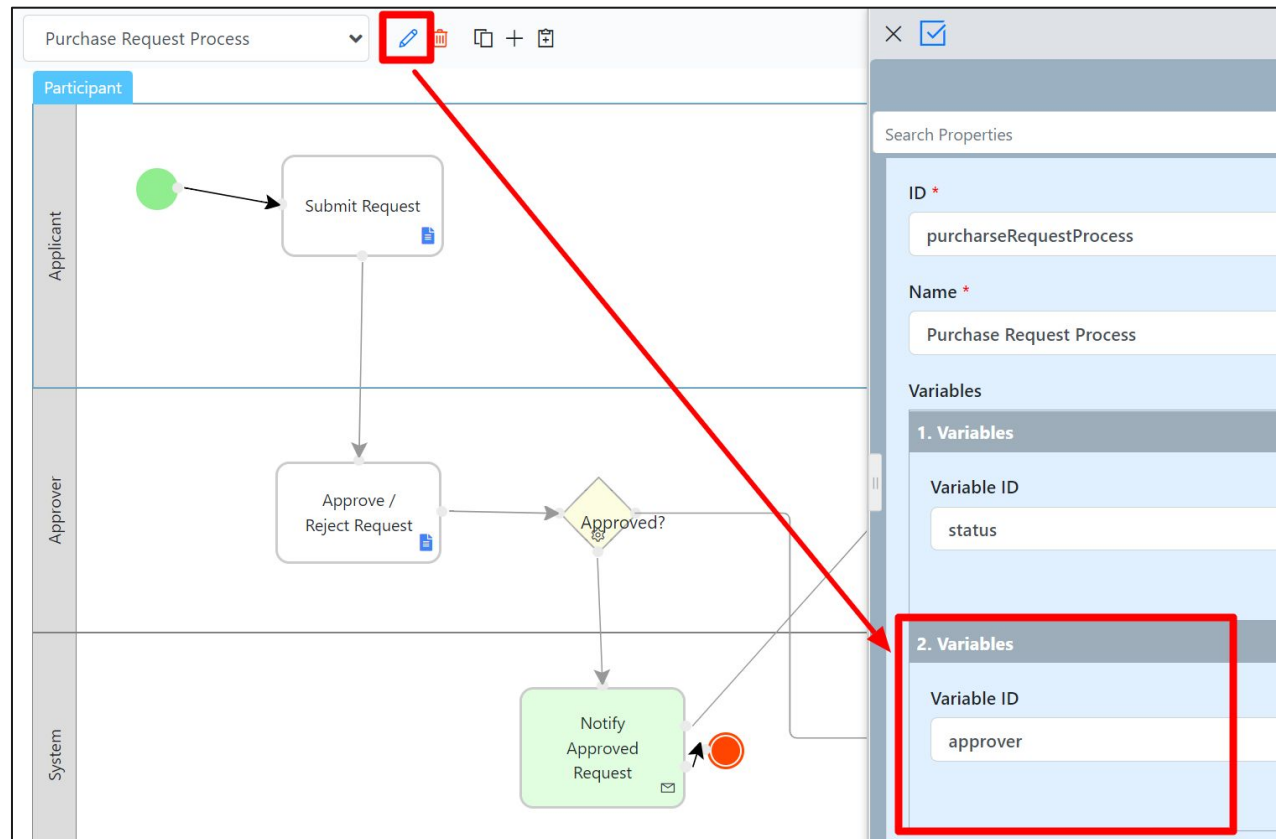
# Problem Statement

- A new role is added to existing process but the participant can only be decided during the runtime (In the last activity before the activity in the new role's swimlane)

**Example use case:** Selecting your approver before submitting your purchase request

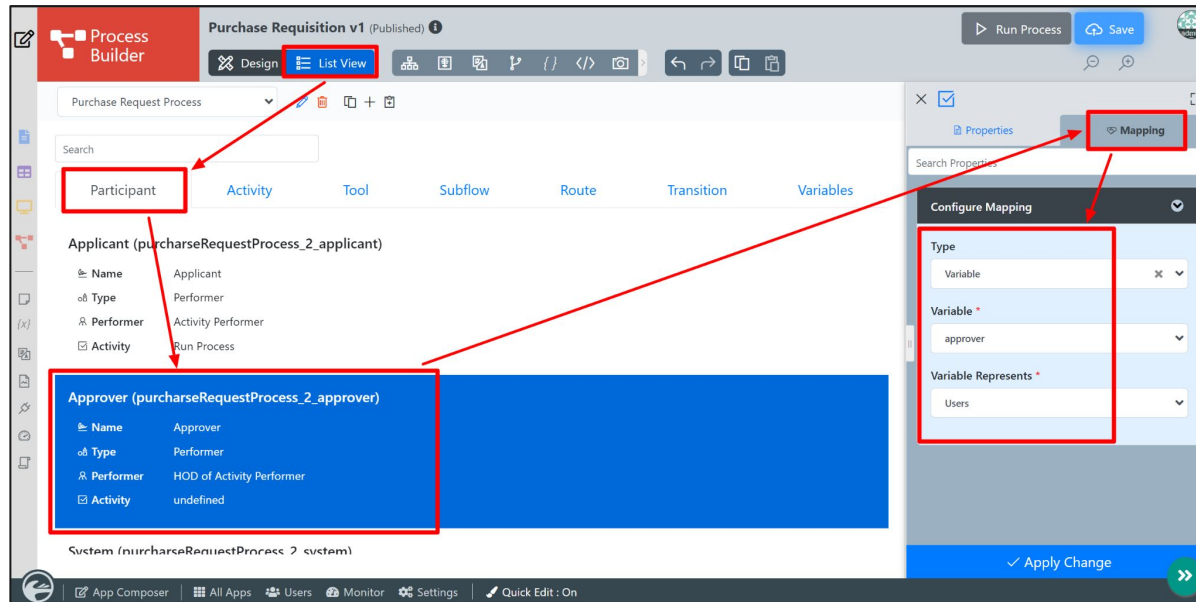
# Exercise 2: Implement Dynamic Participant Capability

- Add a new workflow variable – approver



# Mapping Participant to Workflow Variable

- Change “approver” participant mapping
- Go to “Participant” tab and select “approver” variable id that is created recently
- Then select “Users” for value representation



The screenshot displays the Joget Process Builder interface for a workflow titled "Purchase Requisition v1". The "List View" tab is active, showing a list of participants. The "Approver (purchaseRequestProcess\_2\_approver)" participant is highlighted in blue. A red box highlights the "Participant" tab in the top navigation bar. Another red box highlights the "Mapping" tab in the right-hand "Properties" panel. A third red box highlights the "Configure Mapping" dialog, which shows the following configuration:

Type	Variable *	Variable Represents *
Variable	approver	Users

At the bottom of the dialog, there is a green button labeled "Apply Change".

# Mapping Form Field to Workflow Variable

- Any input element can be used to assign value to the Workflow Variable
- Insert a Select Box element into the form and tie to the workflow variable created earlier

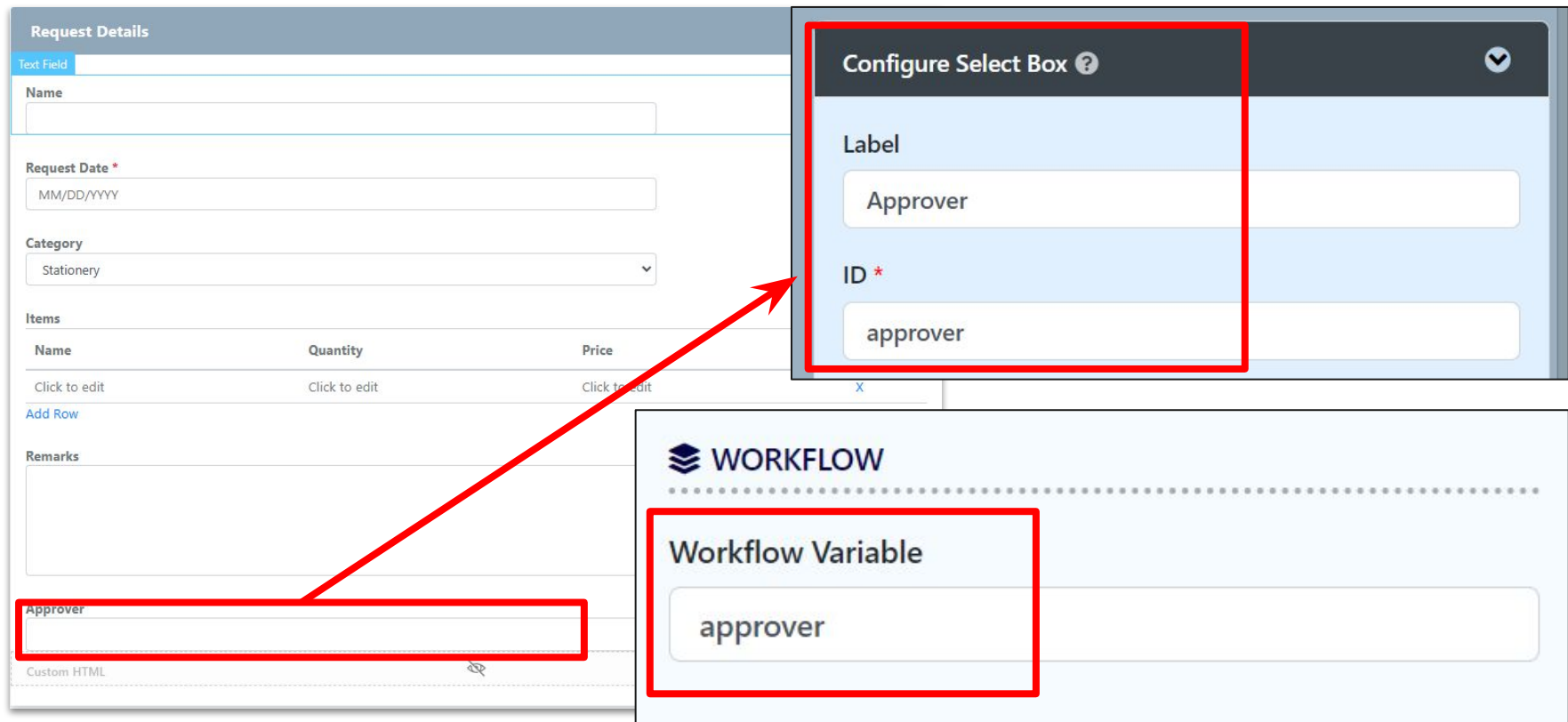
\* Input elements that allow mapping to Workflow Variable are:-

- |                      |                     |
|----------------------|---------------------|
| – Checkbox           | – Textarea          |
| – Date Picker        | – Text Field        |
| – Hidden Field       | – Calculation Field |
| – ID Generator Field | – Multi Select Box  |
| – Radio              | – Popup Select Box  |
| – Select Box         | – Time Picker       |



# Mapping Form Field to Workflow Variable

- Insert a Select Box element into the form and tie to it



**Request Details**

Text Field

Name

Request Date \*

MM/DD/YYYY

Category

Stationery

Items

Name	Quantity	Price
Click to edit	Click to edit	Click to edit

Add Row

Remarks

Approver

Custom HTML

**Configure Select Box ?**

Label

Approver

ID \*

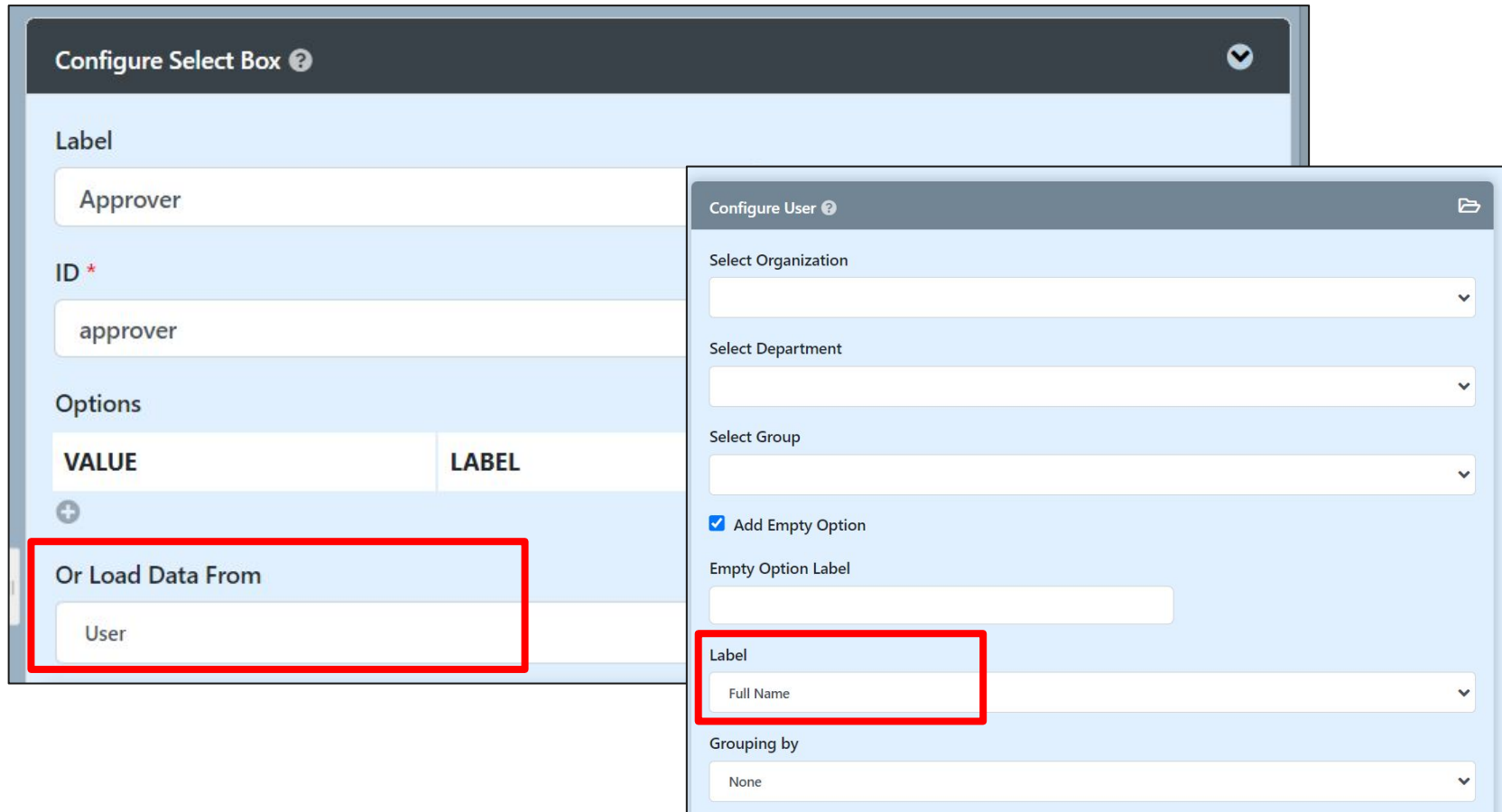
approver

**WORKFLOW**

Workflow Variable

approver

# Configure Select Box to populate user list using User Options Binder



The image shows two overlapping configuration windows in the Joget interface. The background window is titled 'Configure Select Box' and has a dark header bar with a question mark icon and a dropdown arrow. It contains the following fields:

- Label:** A text input field containing 'Approver'.
- ID \*:** A text input field containing 'approver'.
- Options:** A table with two columns: 'VALUE' and 'LABEL'. Below the table is a '+' icon.
- Or Load Data From:** A dropdown menu with 'User' selected. This section is highlighted with a red rectangle.

The foreground window is titled 'Configure User' and has a dark header bar with a question mark icon and a folder icon. It contains the following fields:

- Select Organization:** A dropdown menu.
- Select Department:** A dropdown menu.
- Select Group:** A dropdown menu.
- Add Empty Option:** A checked checkbox.
- Empty Option Label:** A text input field.
- Label:** A dropdown menu with 'Full Name' selected. This section is highlighted with a red rectangle.
- Grouping by:** A dropdown menu with 'None' selected.

# Verify participant mapping

- Run the process to test whether the participant mapping works as designed
- The new participant should be assigned to the username as specified in the form
- Try out the process monitor! Try to check for who is the assigned approver and if it's correct

# Chapter Review

1. Able to assign Participant by using Workflow Variable's value



# Chapter 4

## Mapping Routes Without Workflow Variable

# Why??

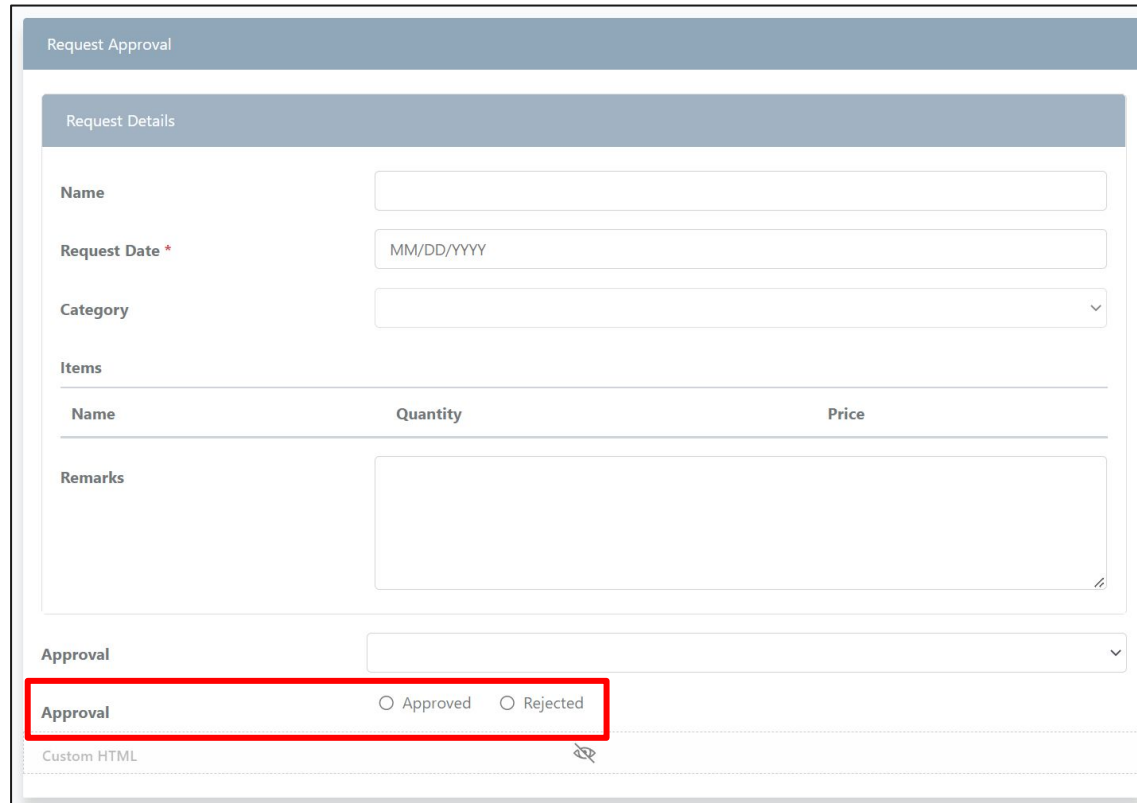
- Workflow variables must be defined in process design and deployed.
- Changes to workflow variables requires process deployment.
- With this method:
  - No need to redeploy process, hence no process migration to worry about
  - Flexibility to adapt to changes

## Exercise 3

- For this example, we will use **Form Data Hash Variable** to read field value(s) directly to perform routing
- No need to set up transition rules directly in the process design. Use the **Simple Rules Decision Plugin**

# Solution

1. Find the approval field ID, to formulate the hash variable: **#form.purchase\_requests.approval#**



Request Approval

Request Details

Name

Request Date \*


Category

Items

Name	Quantity	Price
<div>Remarks <input type="text"/></div>		

Approval

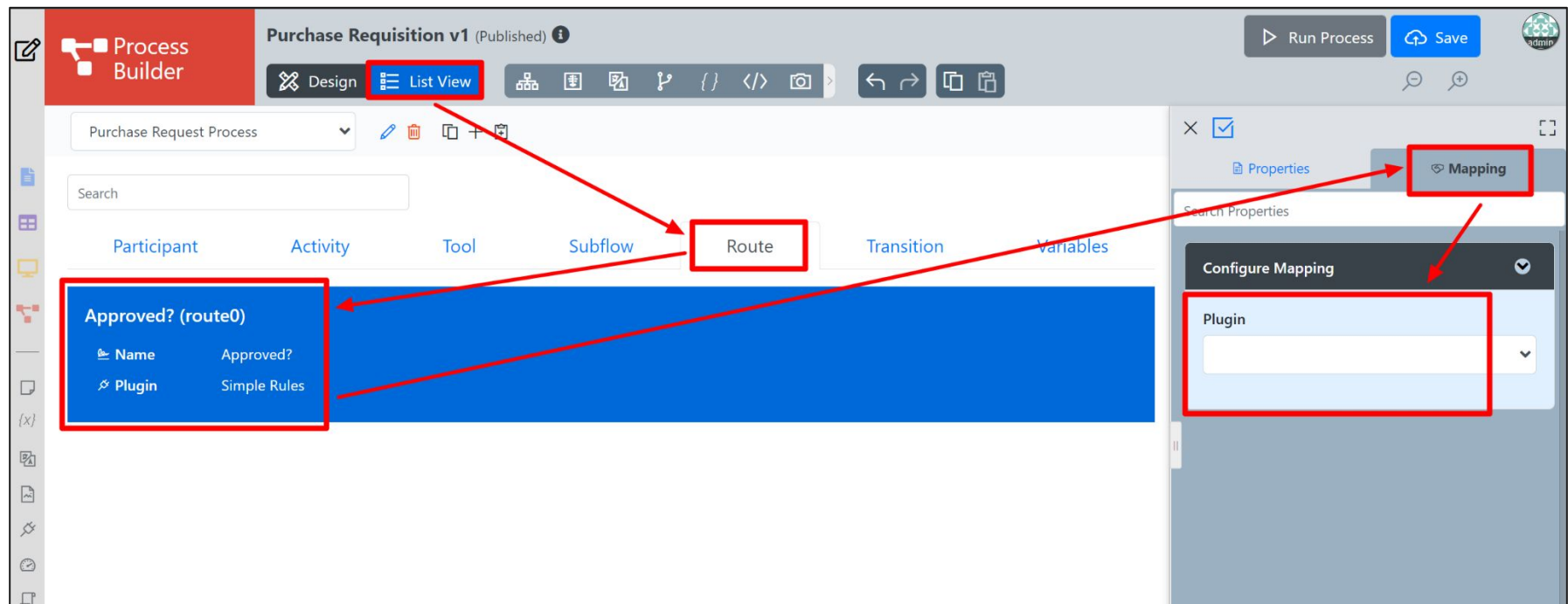
**Approval** ☐ Approved ☐ Rejected

Custom HTML 



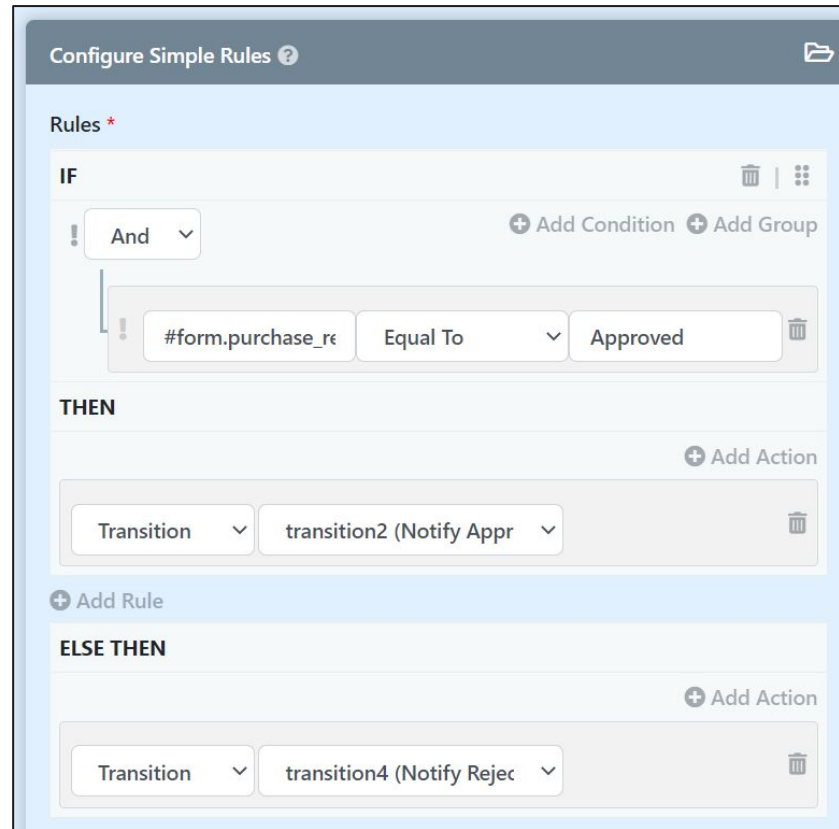
# Solution

2. Go to the process design, edit the approval route, and select the Simple Rules Decision Plugin



# Solution

3. Configure accordingly using the hash variable:  
**#form.purchase\_requests.approval#**



The screenshot displays the 'Configure Simple Rules' interface in Joget. The title bar reads 'Configure Simple Rules' with a help icon. Below the title, there is a 'Rules' section with a red asterisk. The rule is configured as follows:

- IF** section:
  - Logic: **And** (dropdown)
  - Condition: **#form.purchase\_re** (text input) **Equal To** (dropdown) **Approved** (text input)
  - Buttons: **+ Add Condition**, **+ Add Group**
- THEN** section:
  - Button: **+ Add Action**
  - Action: **Transition** (dropdown) **transition2 (Notify Appr** (dropdown)
- ELSE THEN** section:
  - Button: **+ Add Action**
  - Action: **Transition** (dropdown) **transition4 (Notify Rejec** (dropdown)

At the bottom, there is a **+ Add Rule** button.

# Solution

4. Now let's verify in runtime to see if it is working!

**Note:** Using a decision plugin will override the transition rules set for the specific route in the process design.

# Materials

- The completed app with the working exercise is available in the 'Materials' folders as **19.3 and 19.4.jwa**

# Chapter Review

1. Able to route process flow accordingly without using workflow variable



# Chapter 5

## Deadline using Workflow Variable

# Use Case

1. Dynamic Deadline
2. Sample business rules:
  - each record has a different due date, send an email reminder based on the due date

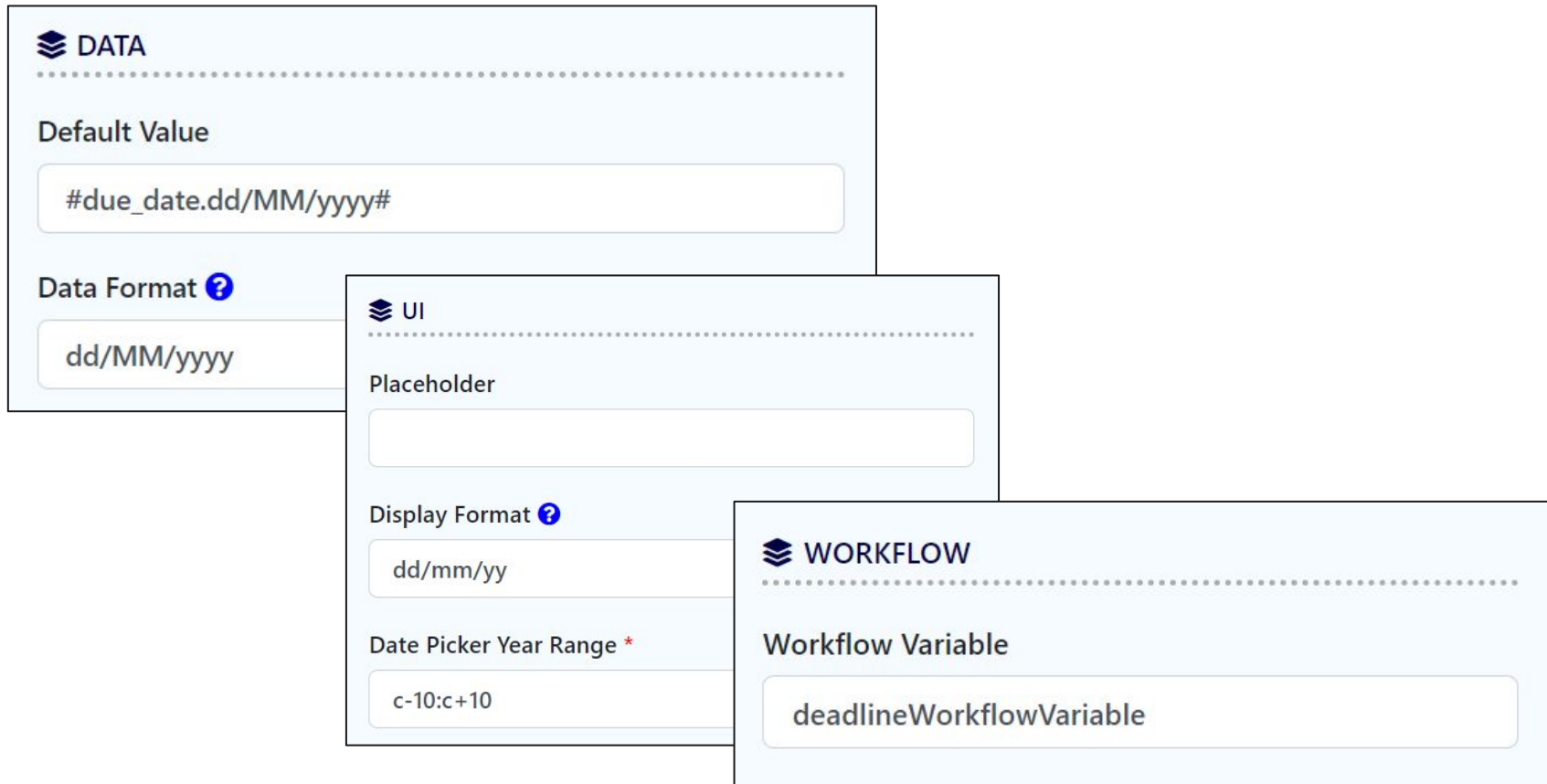
# Exercise 4

- Create a Form with a Date Picker
  - Ensure Data Format is the same as Deadline Duration Unit
  - If users are from different time zones:
    - Consider using “Date and Time stored in UTC”
    - Ensure user profile has its Time Zone set
    - (Optional) Enable User Time Zone in Date Picker properties
- Ensure Deadline properties are configured accordingly:
  - Duration Unit: Date (with correct format)
  - Duration Limit: workflowVariable (any name you define your Workflow Variable)



# Solution

1. Configure the **Date Picker** accordingly in the Form



The image shows three overlapping configuration panels for a Date Picker widget in Joget. The panels are labeled DATA, UI, and WORKFLOW.

**DATA Panel:**

- Default Value:** #due\_date.dd/MM/yyyy#
- Data Format ?** dd/MM/yyyy

**UI Panel:**

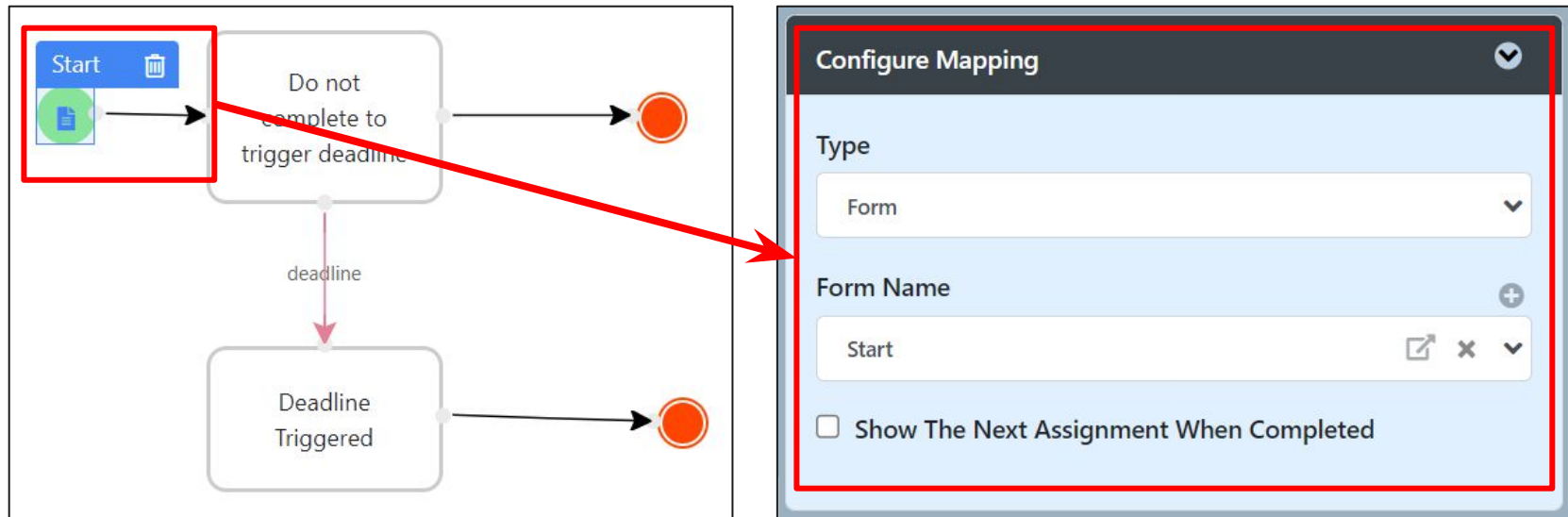
- Placeholder:** (empty text box)
- Display Format ?** dd/mm/yy
- Date Picker Year Range \*** c-10:c+10

**WORKFLOW Panel:**

- Workflow Variable:** deadlineWorkflowVariable

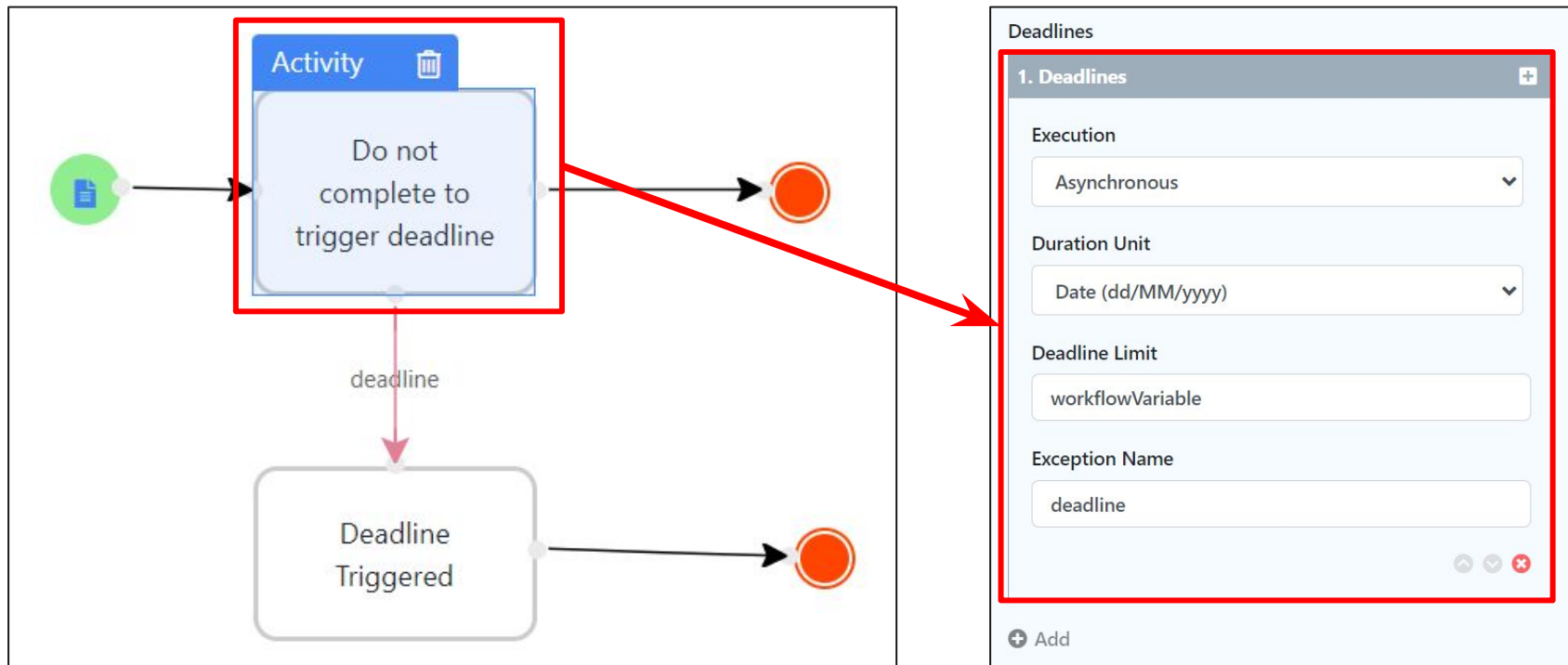
# Solution

2. Map the **Start** node to the Form where the Date Picker is added, so that the selected date is stored in the next activity as the deadline trigger



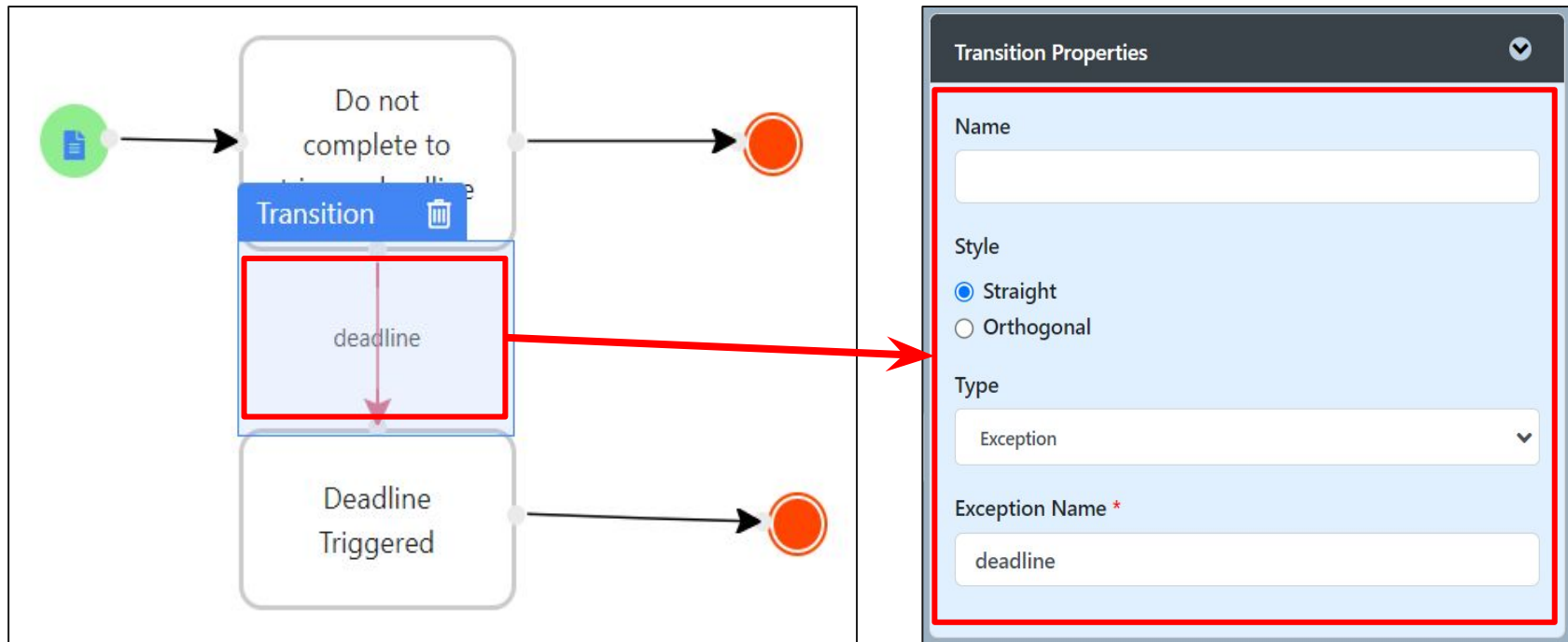
# Solution

3. Build a Process as follows and ensure the **Data Format** in the **Date Picker** configuration is the same as the **Duration Unit** of the “Do Not Complete Form To Trigger Deadline” activity.



# Solution

4. When the Form is completed, the deadline will be triggered



# Materials

- The completed app with the working exercise is available in the 'Materials' folders as **19.5.jwa**

# Chapter Review

1. Instead of entering a fixed duration for the Deadline, learn how to Deadlines can be set and triggered using the Workflow Variable method



## Chapter 6

# Complex Multiple Approval Based on Form Data

# Use Case

1. Requires multiple approval based on configurable business rules
2. Sample business rules:
  - if the amount is more than 500, Cat Grant will be the approver
  - if the request comes from the US, Clark Kent will need to approved it too



## Exercise 5

- Create a Form to allow configuration of business rule and set who will be the approver and set priority (who will be the 1st approver, 2nd approver, etc)
- Create a Tool plugin to generate the approvers based on the configuration

# Solution

Approver Rules

Tue, 15 Nov 2022

Home > Manage Rule

10 Show

Person	Priority	Rule	
<input type="checkbox"/> terry	10	"#form.approver_requests.name#" == "Sai Parn"	<a href="#">Edit</a>
<input type="checkbox"/> cat	10	parseFloat("#form.approver_requests.amount#") == 500	<a href="#">Edit</a>
<input type="checkbox"/> clark	20	"#form.approver_requests.region#" == "US"	<a href="#">Edit</a>
<input type="checkbox"/> etta			
<input type="checkbox"/> julia			

[New](#) [Delete](#)

Approver Rules

Tue, 15 Nov 2022

Home > Manage Rule

Rule

Rule \*

`parseFloat("#form.approver_requests.amount#") == 500`

Person \*

Cat Grant (cat)

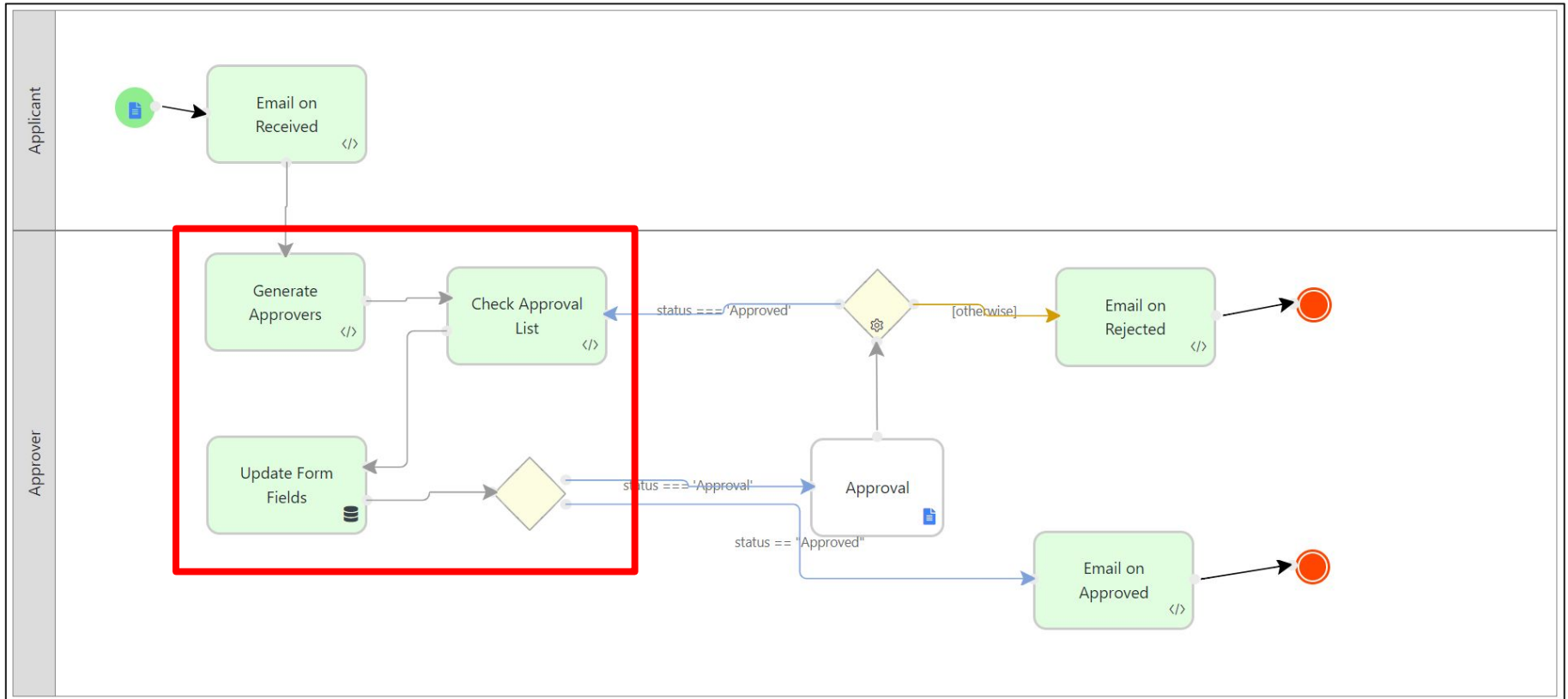
Priority \*

10

[Save](#) [Cancel](#)

Print

# Solution



# Materials

- Check out the completed app to study its code
- Completed app is available at **19.6.jwa**

# Chapter Review

1. Understand how to select or prioritise an approver among multiple approvers in the Process Builder



# Chapter 7

## Process Enhancement Plugin

# Process Enhancement Plugin

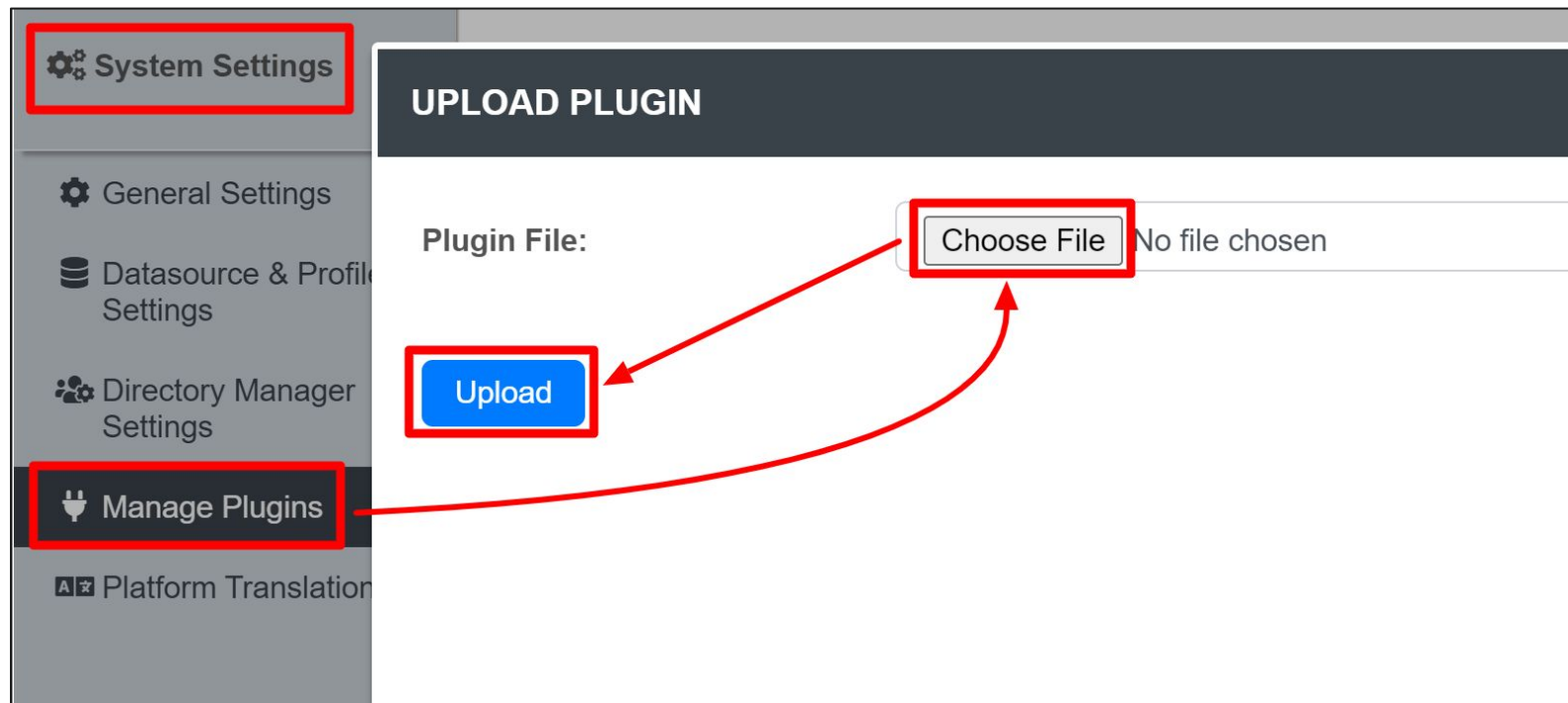
- Set of 4 plugins to automatically generate approval/rejection action buttons in the List and Form screens

Reference:

<https://dev.joget.org/community/display/marketplace/Process+Enhancement+Plugin>

# Installing Process Enhancement Plugin

- Download the plugin from **Joget Marketplace** and import it into your Joget platform via **Manage Plugins** in the **System Settings**





## Exercise 6

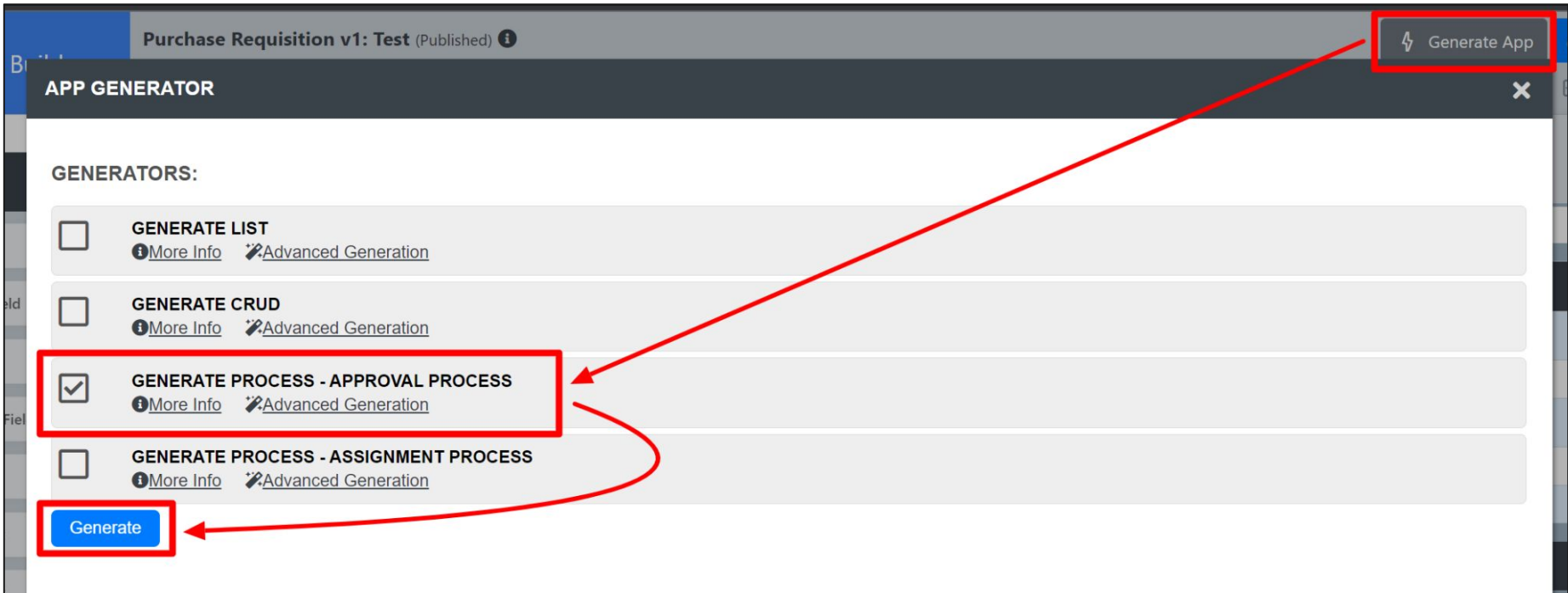
1. Create a new Form and use the App Generator to **Generate Process – Approval Process**
2. Launch the App, click on **Submit Request**, fill up the Form and click **Submit**
3. Click on **Approval**

# Start with creating a Form

1. Create two fields in the Form
  - Hidden Field
    - ID: **status**
    - Workflow Variable: **status**
  - Text Field
    - Label: **Request Note**
    - ID: **request\_note**

# Generate Approval Process

2. Save the Form and use **App Generator** to **Generate Process - Approval Process**



The screenshot shows the 'APP GENERATOR' window for a 'Purchase Requisition v1: Test' form. In the top right corner, a button labeled 'Generate App' with a lightning bolt icon is highlighted with a red box. A red arrow points from this button to the 'GENERATE PROCESS - APPROVAL PROCESS' option in the list below. This option is also highlighted with a red box and has its checkbox checked. Below the list, a blue 'Generate' button is highlighted with a red box. A red arrow points from the 'GENERATE PROCESS - APPROVAL PROCESS' option to this 'Generate' button.

Purchase Requisition v1: Test (Published) ⓘ

**APP GENERATOR** ⓘ

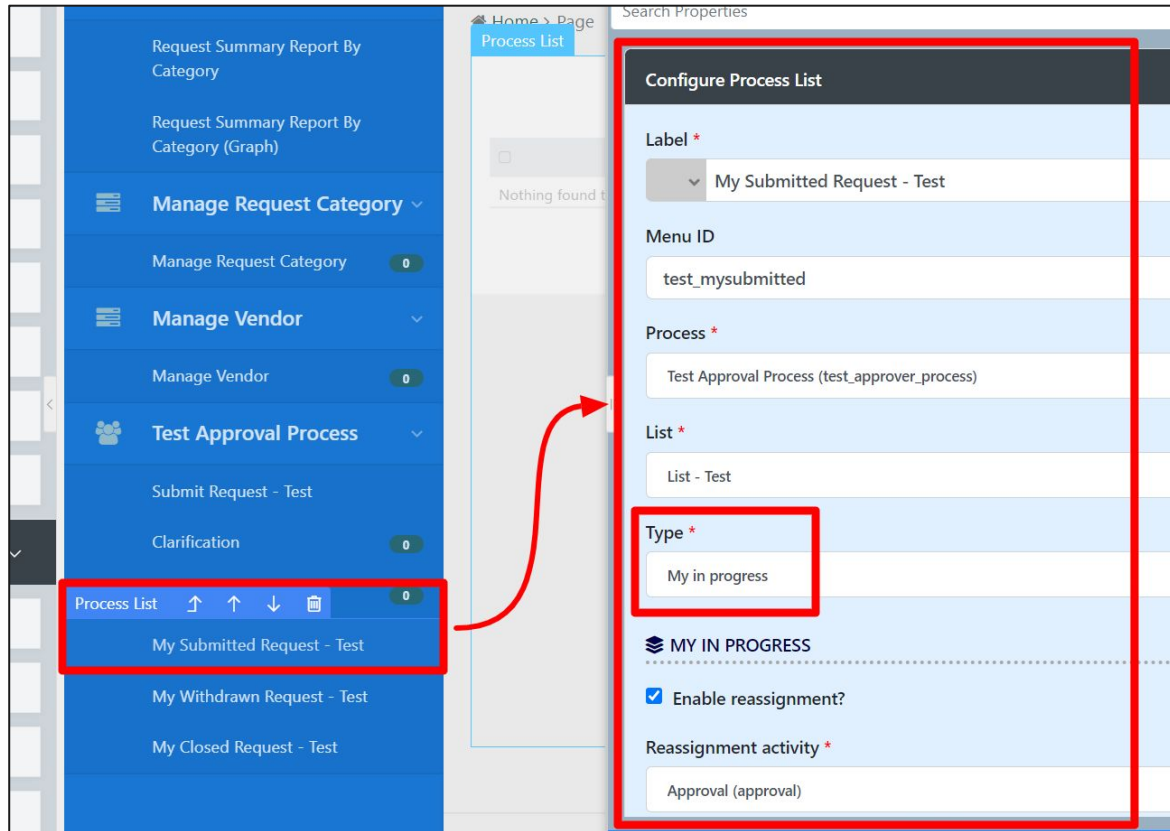
**GENERATORS:**

- ☐ **GENERATE LIST**  
 ⓘ More Info ⓘ Advanced Generation
- ☐ **GENERATE CRUD**  
 ⓘ More Info ⓘ Advanced Generation
- ☒ **GENERATE PROCESS - APPROVAL PROCESS**  
 ⓘ More Info ⓘ Advanced Generation
- ☐ **GENERATE PROCESS - ASSIGNMENT PROCESS**  
 ⓘ More Info ⓘ Advanced Generation

**Generate**

# Process List

3. The **Type** field determines what type of tasks it will display and determines the type of buttons will be available

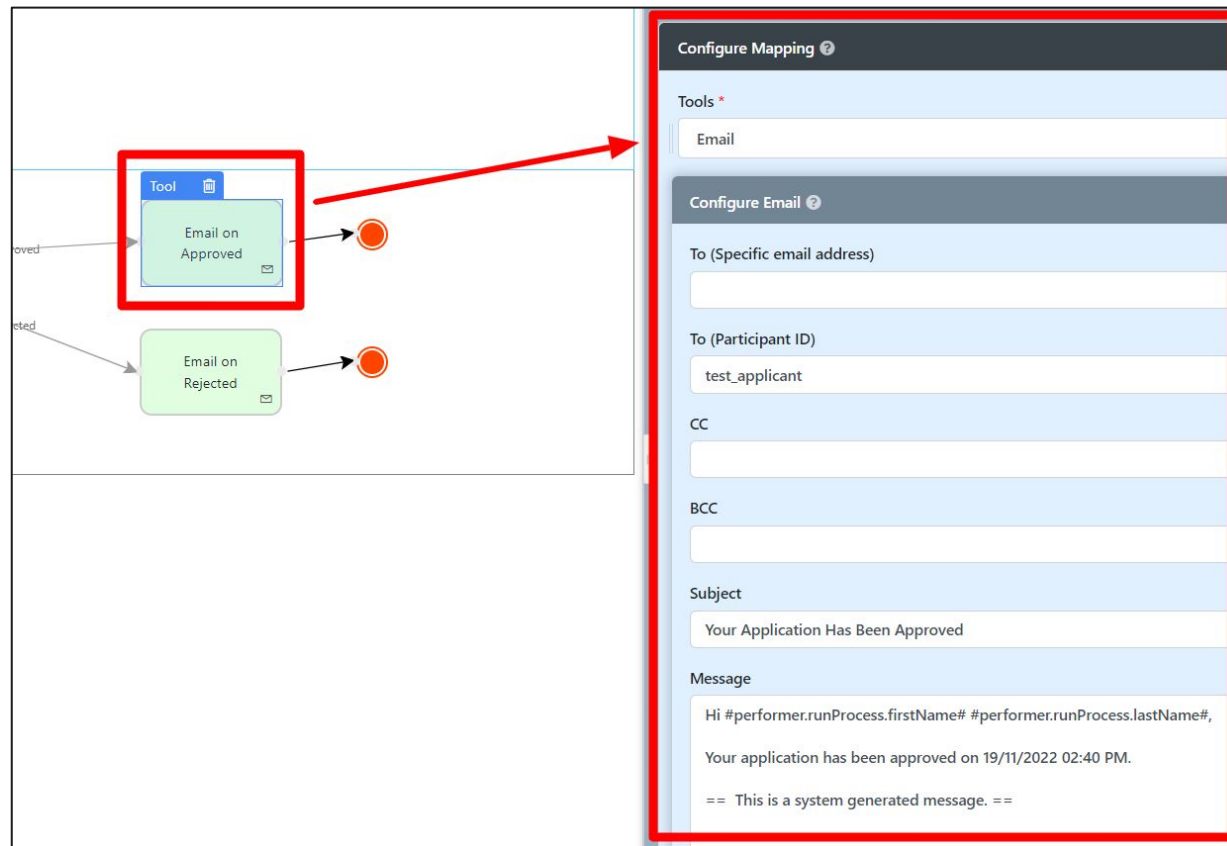


The screenshot displays the Joget Process List configuration interface. On the left, a sidebar lists various menu items, including 'Process List', which is highlighted with a red box. A red arrow points from this box to the 'Configure Process List' panel on the right. The 'Configure Process List' panel is also outlined with a red border and contains the following fields:

- Label \***: My Submitted Request - Test
- Menu ID**: test\_mysubmitted
- Process \***: Test Approval Process (test\_approver\_process)
- List \***: List - Test
- Type \***: My in progress (highlighted with a red box)
- MY IN PROGRESS** (header)
- ☒ **Enable reassignment?**
- Reassignment activity \***: Approval (approval)

# Sending Email of Approval Status

4. Configure the mapping accordingly, for sending email of the Approval Status as either **Approved** or **Rejected**



The diagram illustrates the configuration of email mapping for approval status. On the left, a workflow diagram shows two paths: 'Approved' and 'Rejected'. The 'Approved' path leads to a tool labeled 'Email on Approved', which is highlighted with a red box. A red arrow points from this tool to the 'Configure Mapping' panel on the right. The 'Rejected' path leads to a tool labeled 'Email on Rejected'.

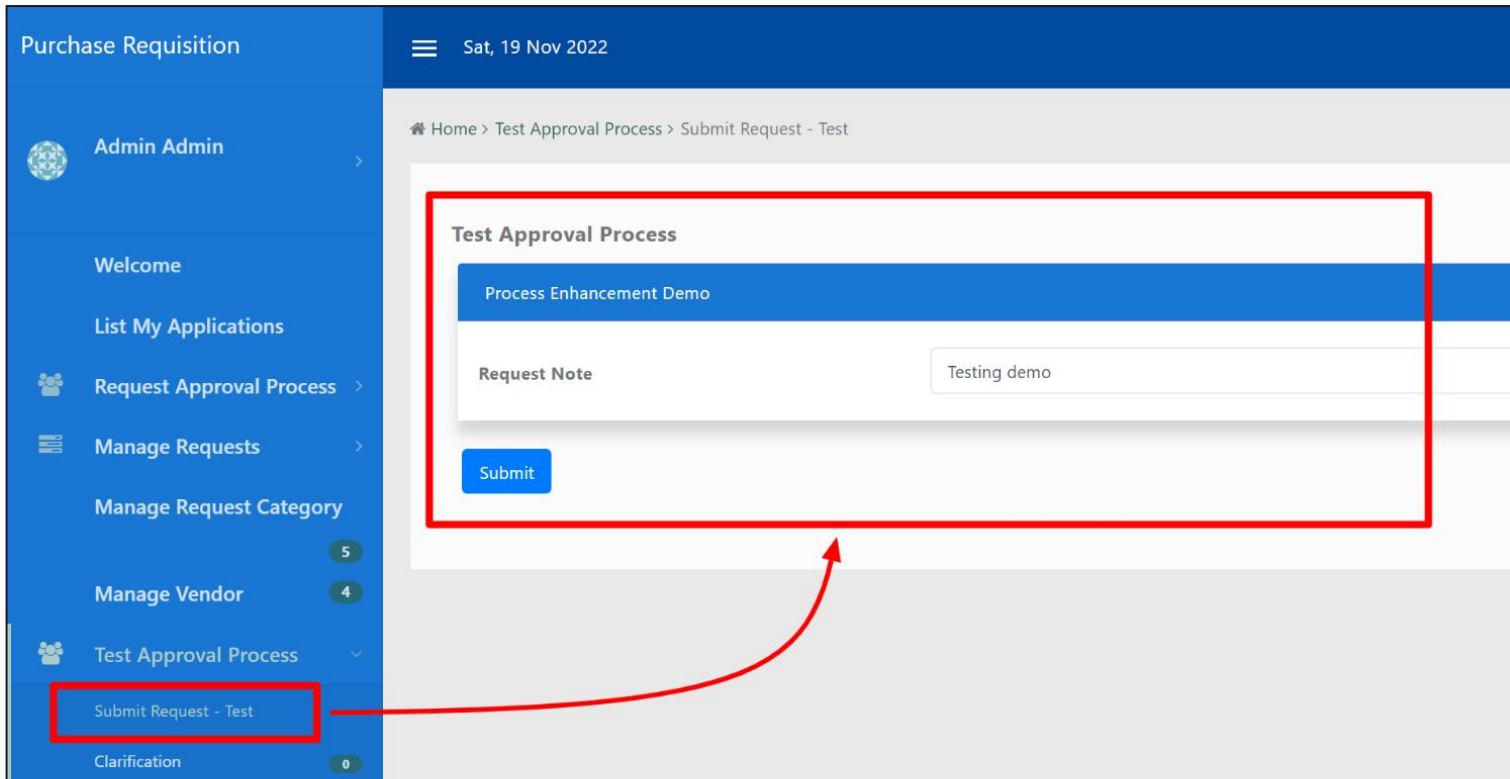
The 'Configure Mapping' panel on the right is also highlighted with a red box and contains the following configuration details:

- Tools \***
  - Email
- Configure Email**
  - To (Specific email address)**: [Empty field]
  - To (Participant ID)**: test\_applicant
  - CC**: [Empty field]
  - BCC**: [Empty field]
  - Subject**: Your Application Has Been Approved
  - Message**:

```
Hi #performer.runProcess.firstName# #performer.runProcess.lastName#,  
  
Your application has been approved on 19/11/2022 02:40 PM.  
  
== This is a system generated message. ==
```

# Testing out

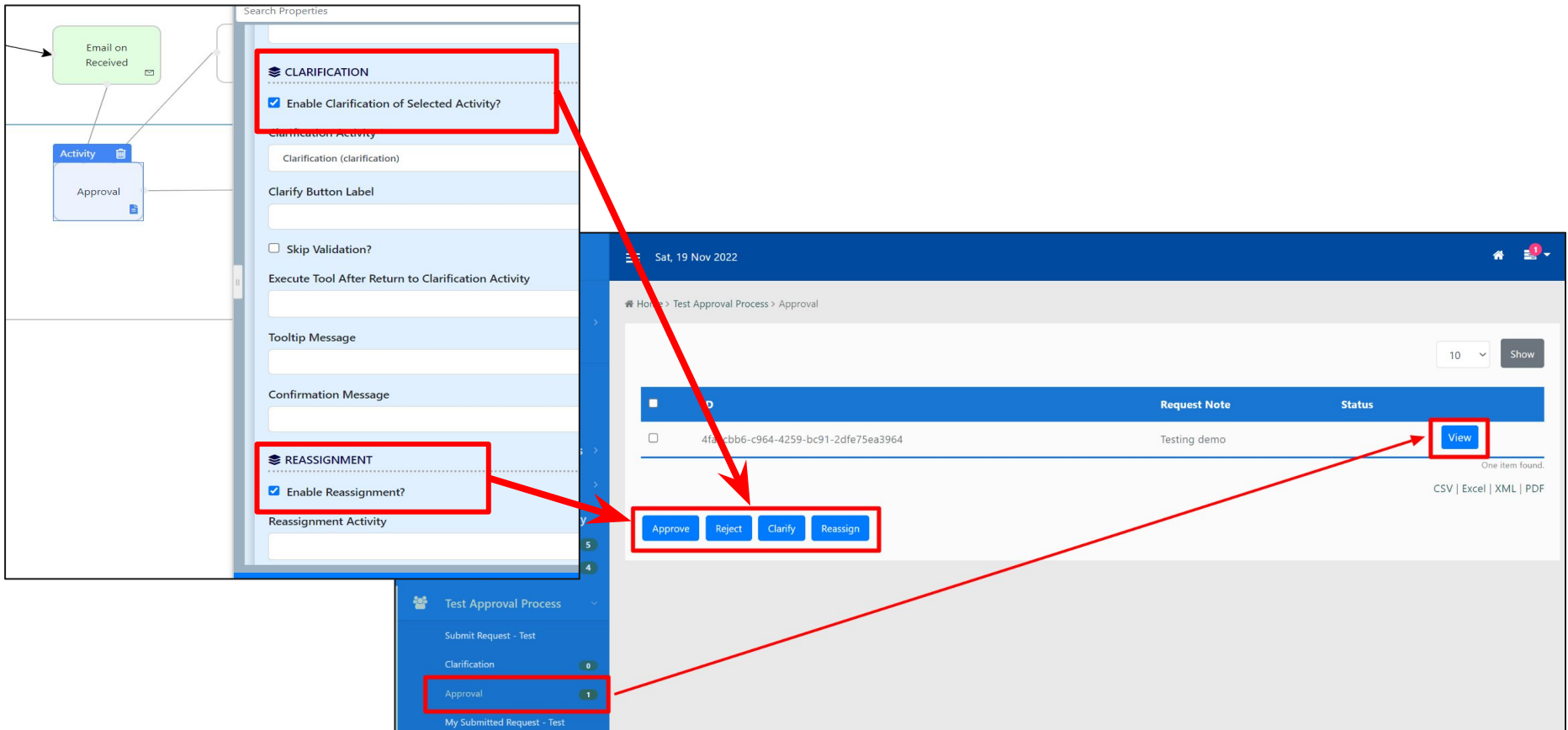
5. Try submitting a Process after the Approval Process Menu is created



The screenshot displays the Joget application interface. On the left is a blue sidebar menu with the title 'Purchase Requisition'. The menu items include 'Admin Admin', 'Welcome', 'List My Applications', 'Request Approval Process', 'Manage Requests', 'Manage Request Category', 'Manage Vendor', 'Test Approval Process', 'Submit Request - Test', and 'Clarification'. The 'Submit Request - Test' item is highlighted with a red rectangle. A red arrow points from this item to the main content area. The main content area has a dark blue header with a hamburger menu icon and the date 'Sat, 19 Nov 2022'. Below the header is a breadcrumb trail: 'Home > Test Approval Process > Submit Request - Test'. The main content area contains a form titled 'Test Approval Process'. The form has a blue header bar with the text 'Process Enhancement Demo'. Below this is a 'Request Note' field with the text 'Testing demo'. At the bottom of the form is a blue 'Submit' button. The entire form is enclosed in a red rectangle.

# Testing out

## 6. Automatically mapped Activities to Form after Approval Process is generated



The image illustrates the configuration and runtime of an approval process in Joget.

**Workflow Editor (Left):**

- CLARIFICATION:** The checkbox "Enable Clarification of Selected Activity?" is checked. The "Clarification Activity" is set to "Clarification (clarification)".
- REASSIGNMENT:** The checkbox "Enable Reassignment?" is checked. The "Reassignment Activity" is set to "Reassignment (reassignment)".

**Runtime Interface (Right):**

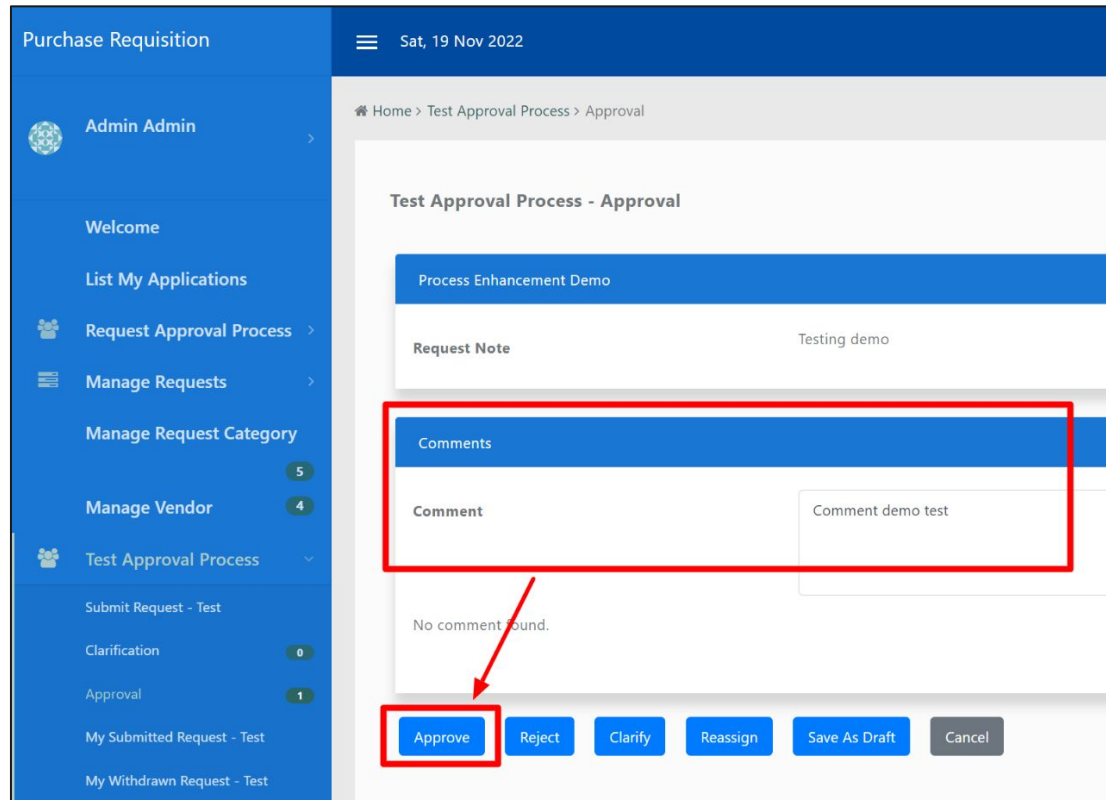
- The "Approval" activity is selected in the left sidebar.
- The main form displays a table with one row:
 

	Request Note	Status
<input type="checkbox"/>	4f3cbb6-c964-4259-bc91-2dfe75ea3964	Testing demo
- Below the table, the "Approve", "Reject", "Clarify", and "Reassign" buttons are visible.
- The "View" button next to the table row is highlighted.

Red arrows indicate the mapping from the workflow editor settings to the runtime form elements: from "Enable Clarification of Selected Activity?" to the "Clarify" button, from "Enable Reassignment?" to the "Reassign" button, and from the "Approval" activity to the "View" button.

# Testing out

7. When clicked on **View** hyperlink, you can give comments to the Process and click on **Approve**



Purchase Requisition

Admin Admin

Welcome

List My Applications

Request Approval Process

Manage Requests

Manage Request Category

Manage Vendor

Test Approval Process

Submit Request - Test

Clarification

Approval

My Submitted Request - Test

My Withdrawn Request - Test

Sat, 19 Nov 2022

Home > Test Approval Process > Approval

Test Approval Process - Approval

Process Enhancement Demo

Request Note

Testing demo

Comments

Comment

Comment demo test

No comment found.

Approve

Reject

Clarify

Reassign

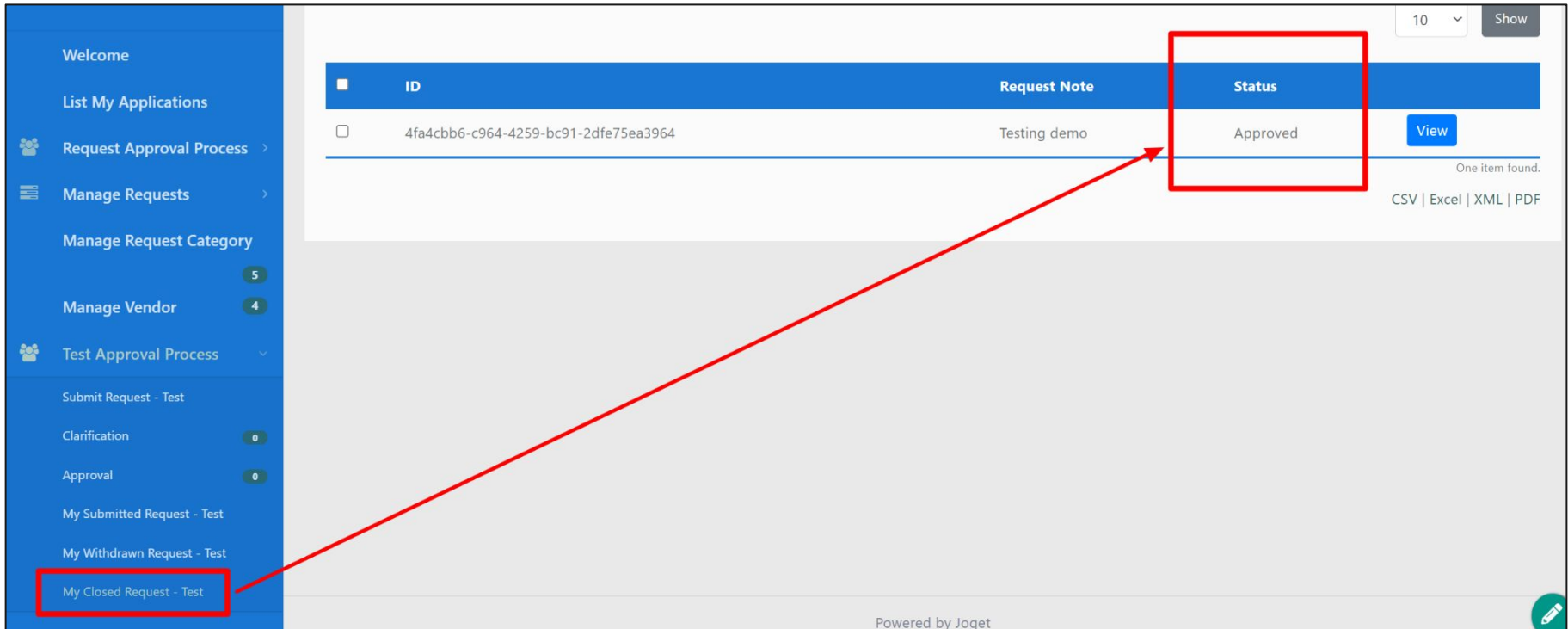
Save As Draft

Cancel



# Testing out

8. Check **My Closed Request** to see the updated **Status** is now set to **Approve**



The screenshot displays the Joget Request Management interface. On the left is a blue sidebar with a menu. The menu items are: Welcome, List My Applications, Request Approval Process (with a sub-menu icon), Manage Requests (with a sub-menu icon), Manage Request Category, Manage Vendor (with a badge showing 5), Test Approval Process (with a sub-menu icon), Submit Request - Test, Clarification (with a badge showing 0), Approval (with a badge showing 0), My Submitted Request - Test, My Withdrawn Request - Test, and My Closed Request - Test (which is highlighted with a red box). A red arrow points from this highlighted menu item to the main content area. The main content area shows a table with the following data:

ID	Request Note	Status	
4fa4cbb6-c964-4259-bc91-2dfe75ea3964	Testing demo	Approved	<a href="#">View</a>

Below the table, it says "One item found." and "CSV | Excel | XML | PDF". At the bottom right, it says "Powered by Joget".

# Materials

- Check out the completed app to study its code
- Completed app is available at **19.7.jwa**

# Chapter Review

1. Understand and being able to use Process Enhancement Plugin effectively

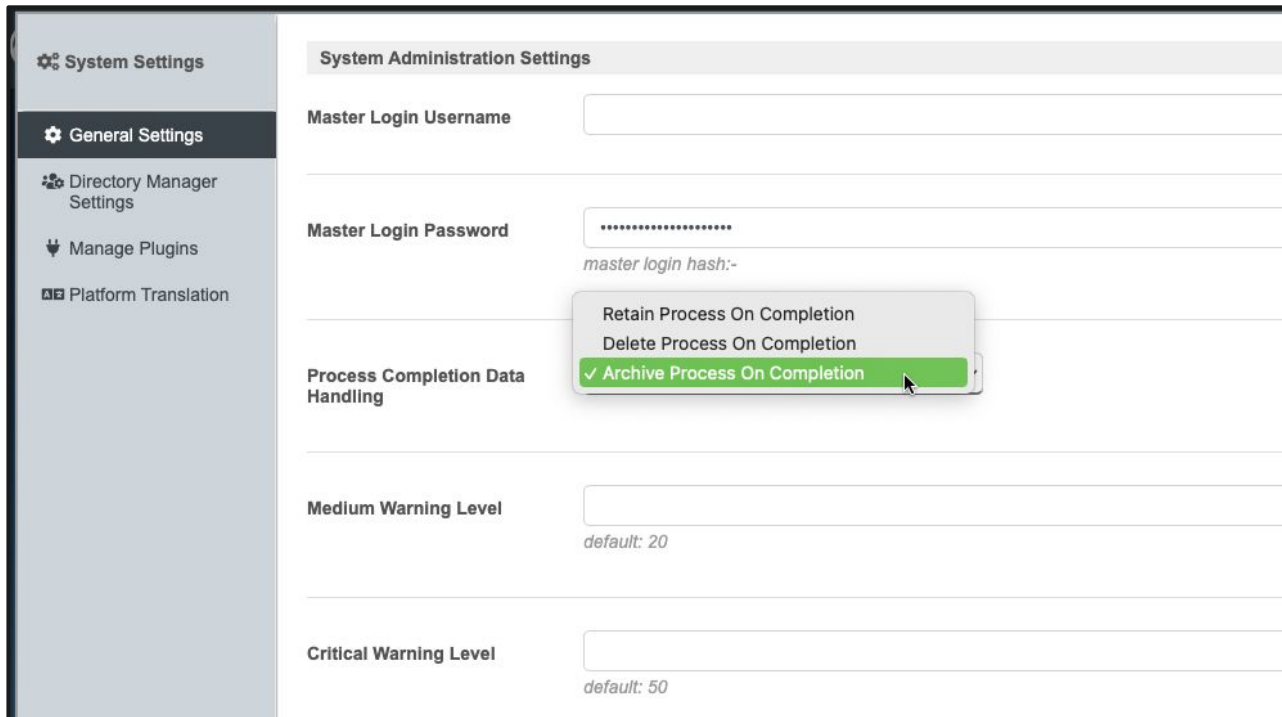


# Chapter 8

## Process Completion Data Handling

# Process Completion Data Handling

- New in DX 8



The screenshot displays the 'System Administration Settings' page in Joget DX 8. The left sidebar contains a menu with 'System Settings' (selected), 'General Settings', 'Directory Manager Settings', 'Manage Plugins', and 'Platform Translation'. The main content area is titled 'System Administration Settings' and includes the following fields:

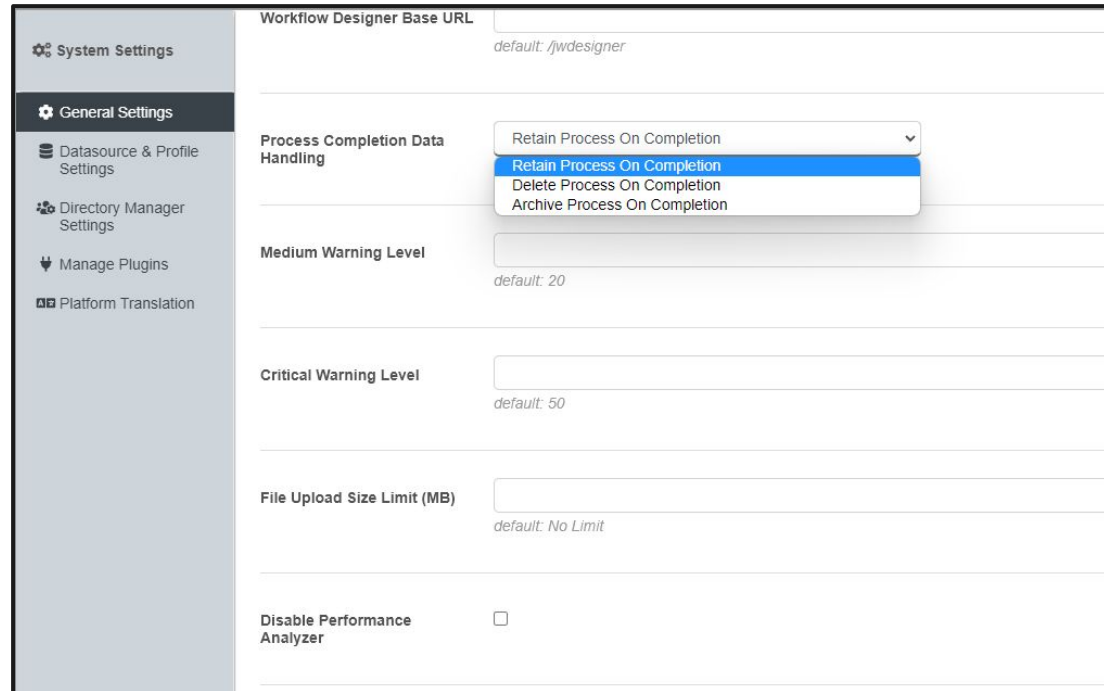
- Master Login Username**: A text input field.
- Master Login Password**: A password input field with a masked value '.....' and a label 'master login hash:-' below it.
- Process Completion Data Handling**: A dropdown menu with three options: 'Retain Process On Completion', 'Delete Process On Completion', and 'Archive Process On Completion' (which is highlighted in green and has a checkmark).
- Medium Warning Level**: A text input field with a label 'default: 20' below it.
- Critical Warning Level**: A text input field with a label 'default: 50' below it.

# Process Completion Data Handling

- Improve performance with process related functions by either:-
  - Retain Process On Completion
  - Delete Process On Completion
  - Archive Process On Completion
- The latter 2 options are recommended to improve performance.

# Auto Archive

Auto Archive automatically archives the completed processes to achieve better operational performance. Choose retain process on completion.

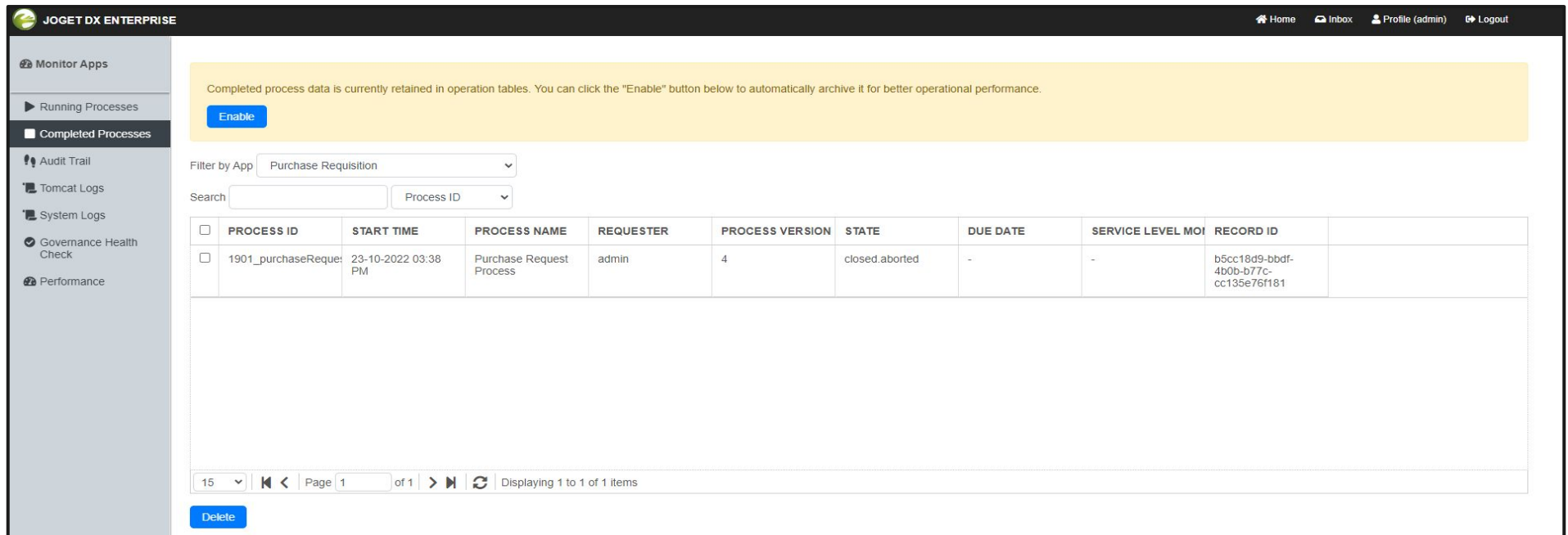


The screenshot displays the 'System Settings' interface in Joget. The left sidebar contains a menu with the following items: 'System Settings' (selected), 'General Settings', 'Datasource & Profile Settings', 'Directory Manager Settings', 'Manage Plugins', and 'Platform Translation'. The main content area is titled 'Workflow Designer Base URL' with a text input field containing the default value '/jwdesigner'. Below this, the 'Process Completion Data Handling' section features a dropdown menu with three options: 'Retain Process On Completion' (highlighted in blue), 'Delete Process On Completion', and 'Archive Process On Completion'. The 'Medium Warning Level' section has a text input field with the default value '20'. The 'Critical Warning Level' section has a text input field with the default value '50'. The 'File Upload Size Limit (MB)' section has a text input field with the default value 'No Limit'. The 'Disable Performance Analyzer' section has an unchecked checkbox.

Setting	Value
Workflow Designer Base URL	/jwdesigner
Process Completion Data Handling	Retain Process On Completion
Medium Warning Level	20
Critical Warning Level	50
File Upload Size Limit (MB)	No Limit
Disable Performance Analyzer	<input type="checkbox"/>

# Auto Archive

Next, click on the enable button.



Completed process data is currently retained in operation tables. You can click the "Enable" button below to automatically archive it for better operational performance.

[Enable](#)

Filter by App: Purchase Requisition

Search:  Process ID

<input type="checkbox"/>	PROCESS ID	START TIME	PROCESS NAME	REQUESTER	PROCESS VERSION	STATE	DUE DATE	SERVICE LEVEL MOI	RECORD ID
<input type="checkbox"/>	1901_purchaseReque	23-10-2022 03:38 PM	Purchase Request Process	admin	4	closed.aborted	-	-	b5cc18d9-bbdf-4b0b-b77c-cc135e76f181

15 ⏪ ⏩ Page 1 of 1 🔄 Displaying 1 to 1 of 1 items

[Delete](#)

Reference: <https://dev.joget.org/community/display/DX8/Completed+Processes>



# Module Review

1. Understand the use of Route and its configurations
2. Able to assign Participant by using Workflow Variable's value
3. Able to route process flow accordingly without using workflow variable
4. Complex Multiple Approval based on form data
5. Understand and being able to use Process Enhancement Plugin effectively

# Stay Connected With Joget

- [www.joget.org](http://www.joget.org)
- [community.joget.org](http://community.joget.org)
- [twitter.com/jogetworkflow](https://twitter.com/jogetworkflow)
- [facebook.com/jogetworkflow](https://facebook.com/jogetworkflow)
- [youtube.com/jogetworkflow](https://youtube.com/jogetworkflow)