
Joget DX

Doing More with Your Process Design

 <http://facebook.com/jogetworkflow>

 <http://twitter.com/jogetworkflow>

Prerequisites

1. Knowledge about designing process using the Process Builder.

Content

1. Type of Routes
2. Participant Mapping using Workflow Variable
3. Mapping Routes Without Workflow Variable
4. Deadline using Workflow Variable
5. Complex Multiple Approval Based on Form Data
6. Process Enhancement Plugin

Chapter 1

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)

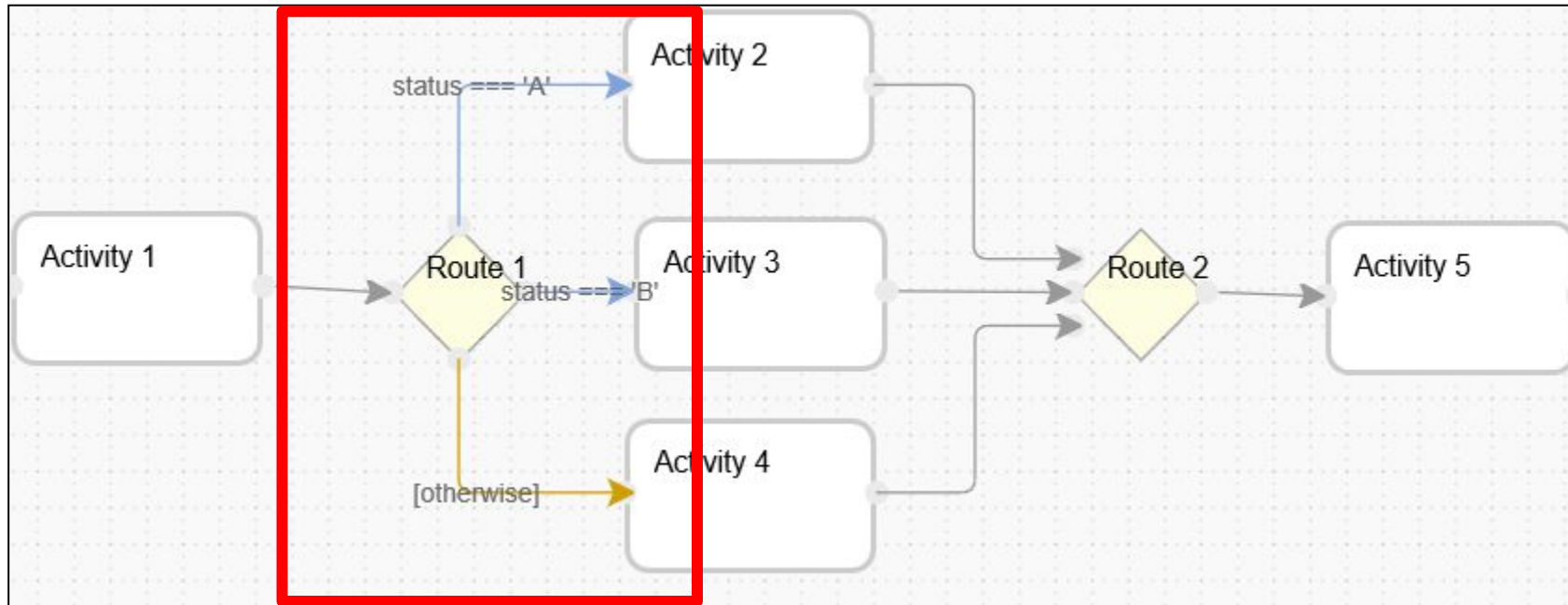


– Parallel (AND)



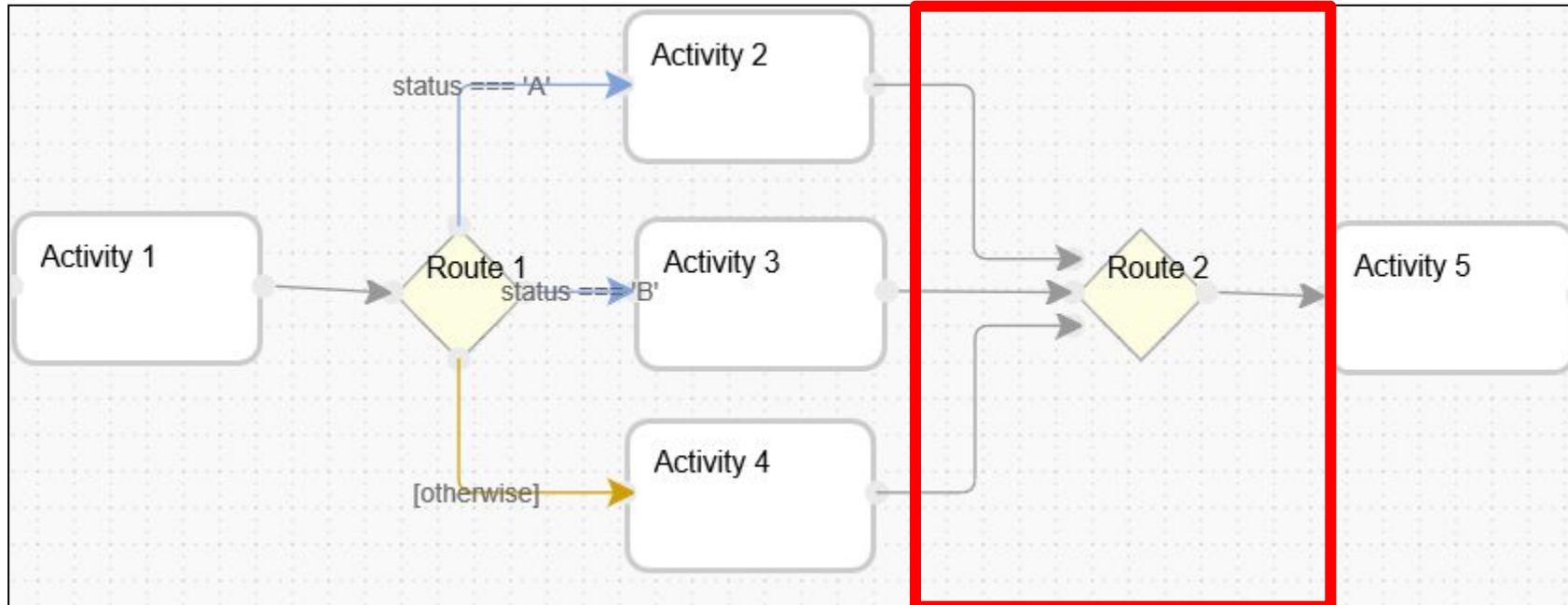
Exclusive Route (XOR)

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



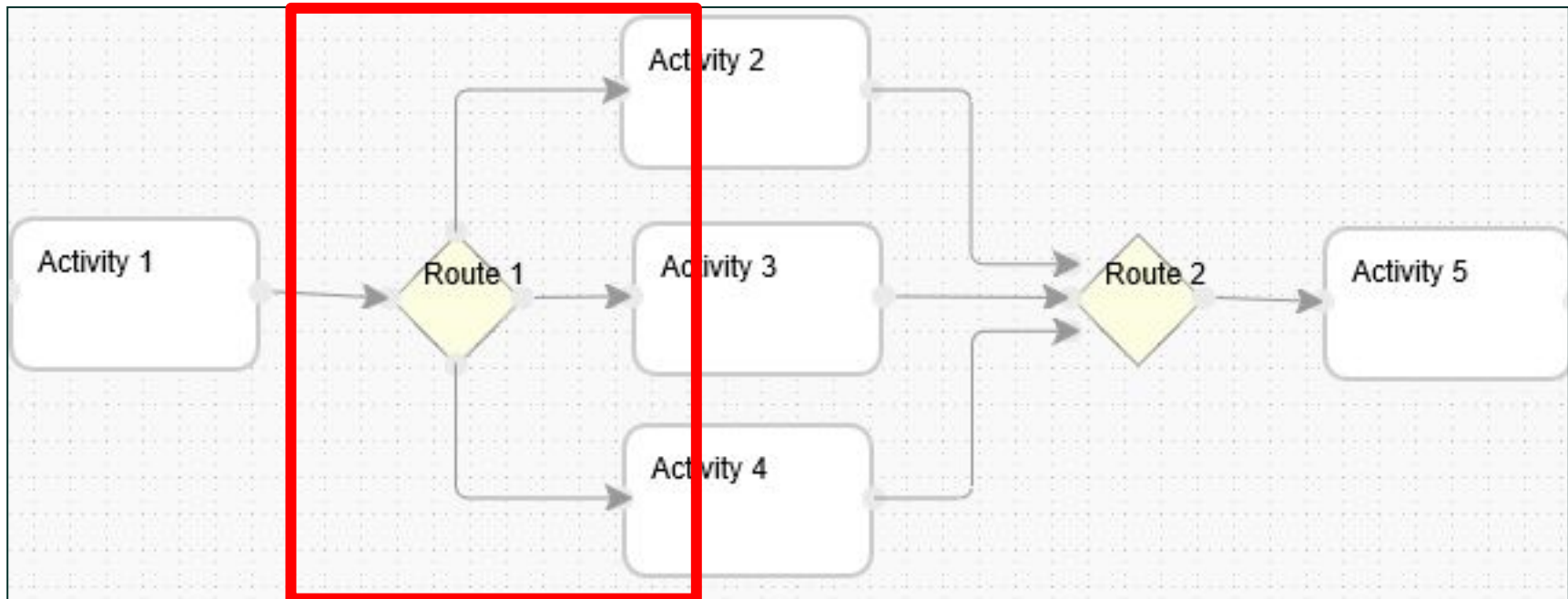
Exclusive Route (XOR)

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

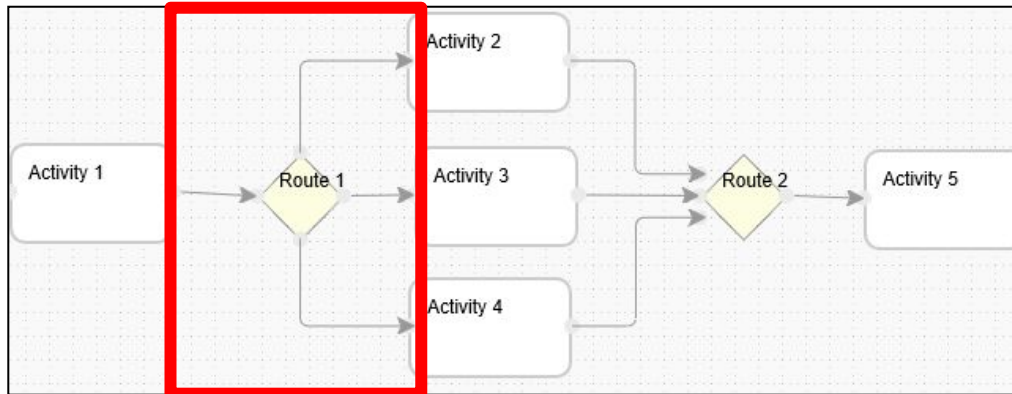


Exclusive Route

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



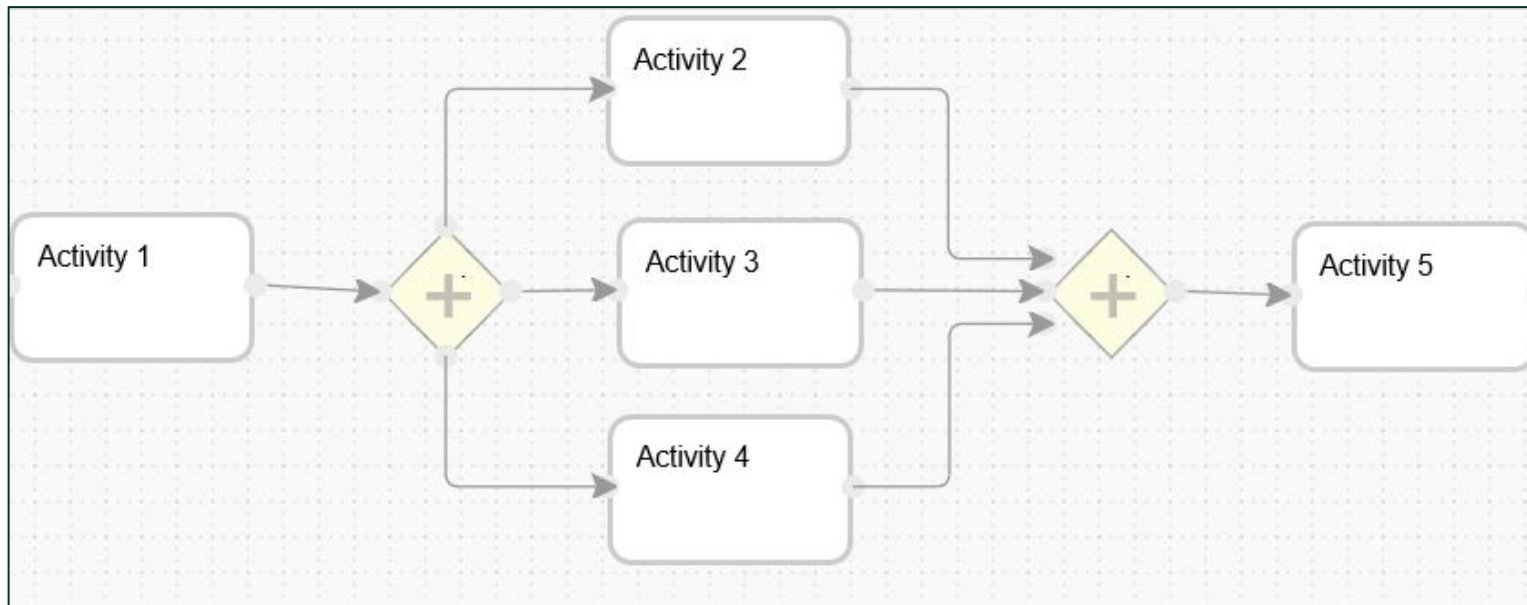
Exclusive Route



- 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

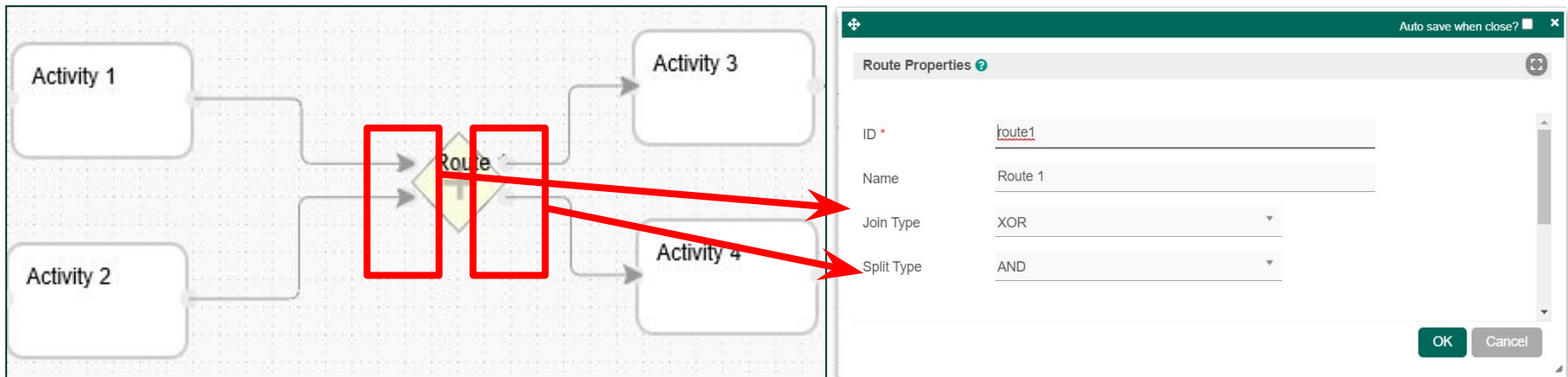
Parallel Route (AND)

- 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

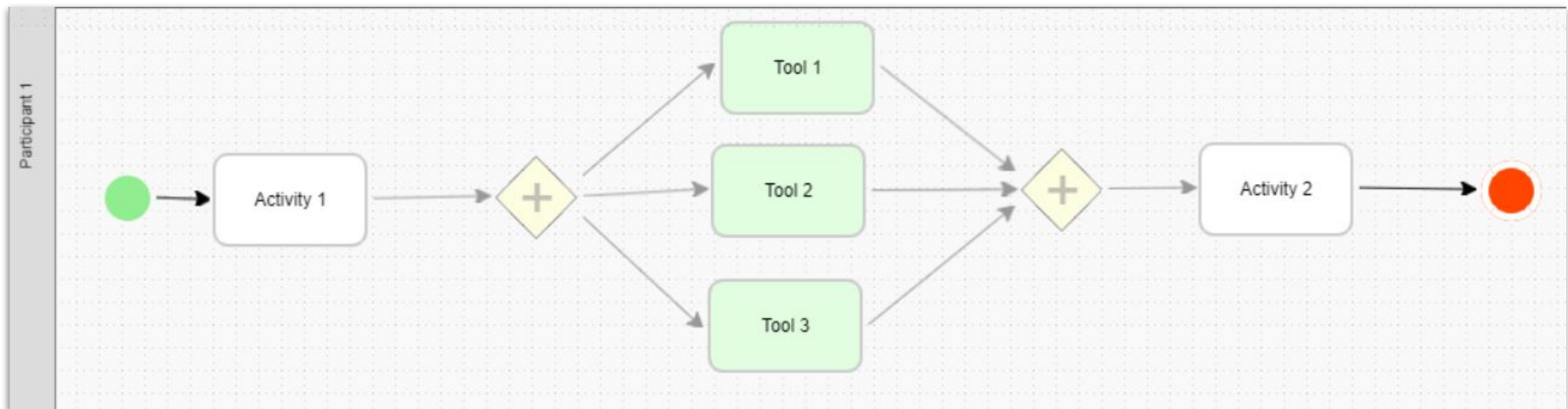
Only 4 possible combinations (Split - Join):

1. XOR - XOR
2. XOR - AND
3. AND - AND
4. AND - XOR

What happen if I do _____? You ask me!!

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 **19.1.jwa**

Chapter Review

1. Understand the use of Route and its configurations.

Chapter 2

Participant Mapping using Workflow Variable

Participant Mapping Using Workflow Variable

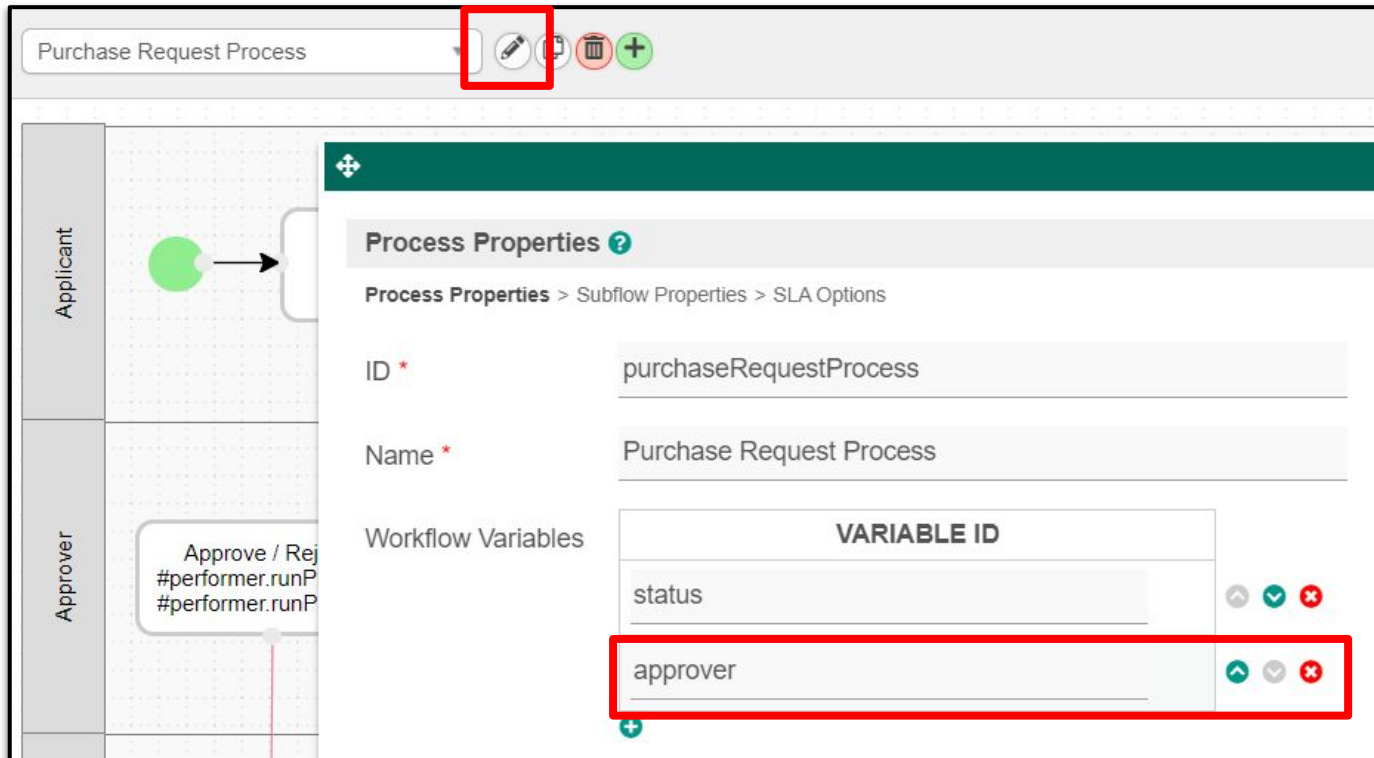
- Dynamically map to participant using workflow variable.
- Workflow variable may consists of:-
 1. Group code
 2. Username
 3. Department code
 4. Head of Department (Department code)
- Supports multiple assignments. (Semicolon separated value [Eg. jack;david;clark])

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: Implement Dynamic Participant Capability

- Add a new workflow variable - approver



Purchase Request Process

Applicant

Approver

Approve / Rej
#performer.runP
#performer.runP

Process Properties ?

Process Properties > Subflow Properties > SLA Options

ID * purchaseRequestProcess

Name * Purchase Request Process

Workflow Variables

VARIABLE ID	
status	⬆️ ⬆️ ⬆️
approver	⬆️ ⬆️ ⬆️

Map Participant to Workflow Variable

- Change “approver” participant mapping. Go to “Map to Workflow Variable” tab and select “approver” variable id that’s created recently. Then, select “Username” for value representation.

MAP PARTICIPANTS TO USERS - APPROVER (APPROVER)

Map to User or Group Map to Org Chart Map to Workflow Variable Map to Plugin

Variable ID:

The value of the Workflow Variable represents:

Group Code

Username

Department Code

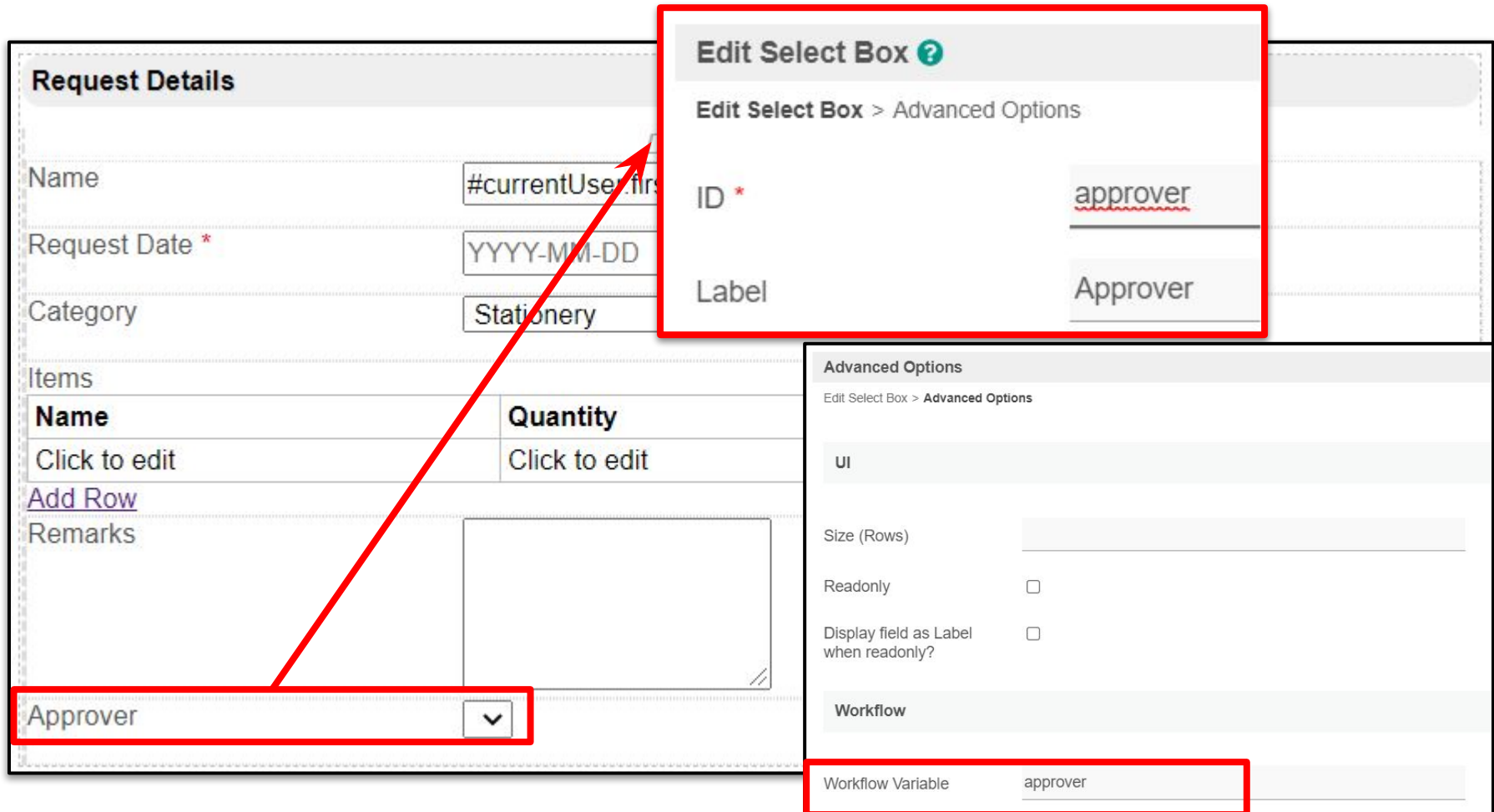
Head of Department (Department Code)

Mapping Form Field to Workflow Variable

- Any *input element can be used to assign value to the Workflow Variable.
- Insert a selectbox element into the form and tie to the workflow variable created earlier.
- * Input elements that allow mapping to Workflow Variable are:-
 - Checkbox
 - Date Picker
 - Hidden Field
 - ID Generator Field
 - Radio
 - Select Box
 - Textarea
 - Text Field
 - Calculation Field
 - Multi Select Box
 - Popup Select Box
 - Time Picker

Mapping Form Field to Workflow Variable

- Insert a select box element into the form and tie to it.



The image shows a form editor interface for a 'Request Details' form. A red arrow points from a select box labeled 'Approver' at the bottom of the form to the 'Edit Select Box' configuration panel. The configuration panel is divided into two sections: 'Advanced Options' and 'Workflow'.

Advanced Options

- ID * : approver
- Label : Approver

Workflow

- Workflow Variable : approver

The form itself contains the following fields:

- Name: #currentUser.firstName
- Request Date *: YYYY-MM-DD
- Category: Stationery
- Items table:

Name	Quantity
Click to edit	Click to edit
Add Row	
Remarks	
- Approver: [Dropdown menu]

Configure select box to populate user list using User Options Binder

Edit Select Box [?](#)

Edit Select Box > Advanced Options

ID ^{*}

Label

Options (Hardcoded)

VALUE	LABEL
+	

Or Choose Options Binder

- Bean Shell Form Binder
- Default Form Options Binder
- Department Options Binder
- Grade Options Binder
- Group Options Binder
- JDBC Binder
- Organization Options Binder
- User Options Binder**

Configure User Options Binder [?](#)

Edit Select Box > **Configure User Options Binder** > Advanced Options

Select Organization

Select Department

Select Group

Add Empty Option

Empty Option Label

Label

Grouping by

Use AJAX for cascade options?

- Full Name with Username
- Username
- Full Name**

Exercise: 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.

Materials

- The completed app with the working exercise is available in the 'materials' folders as **19.2.1.jwa**

Chapter Review

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

Chapter 3

Mapping Routes Without Workflow Variable

Why??

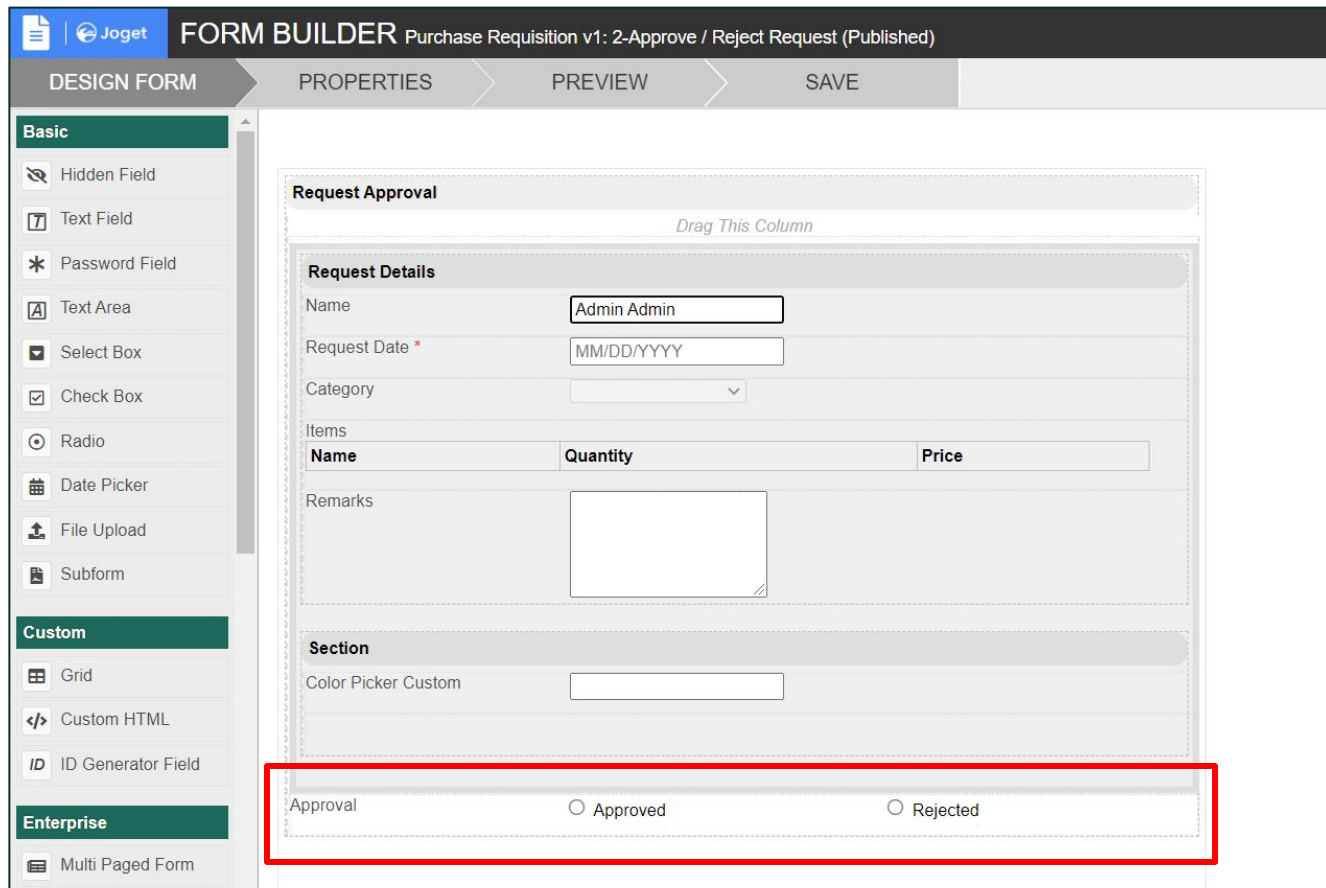
- Workflow variables must be defined in process design and deployed.
- Changes to workflow variables requires process deployment.
- With this method:
 - No need to redeploy process, hence no process migration to worry about
 - Flexibility to adapt to changes

Sample Solution

- 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#



The screenshot displays the Joget Form Builder interface for a form titled "Purchase Requisition v1: 2-Approve / Reject Request (Published)". The interface is divided into several sections:

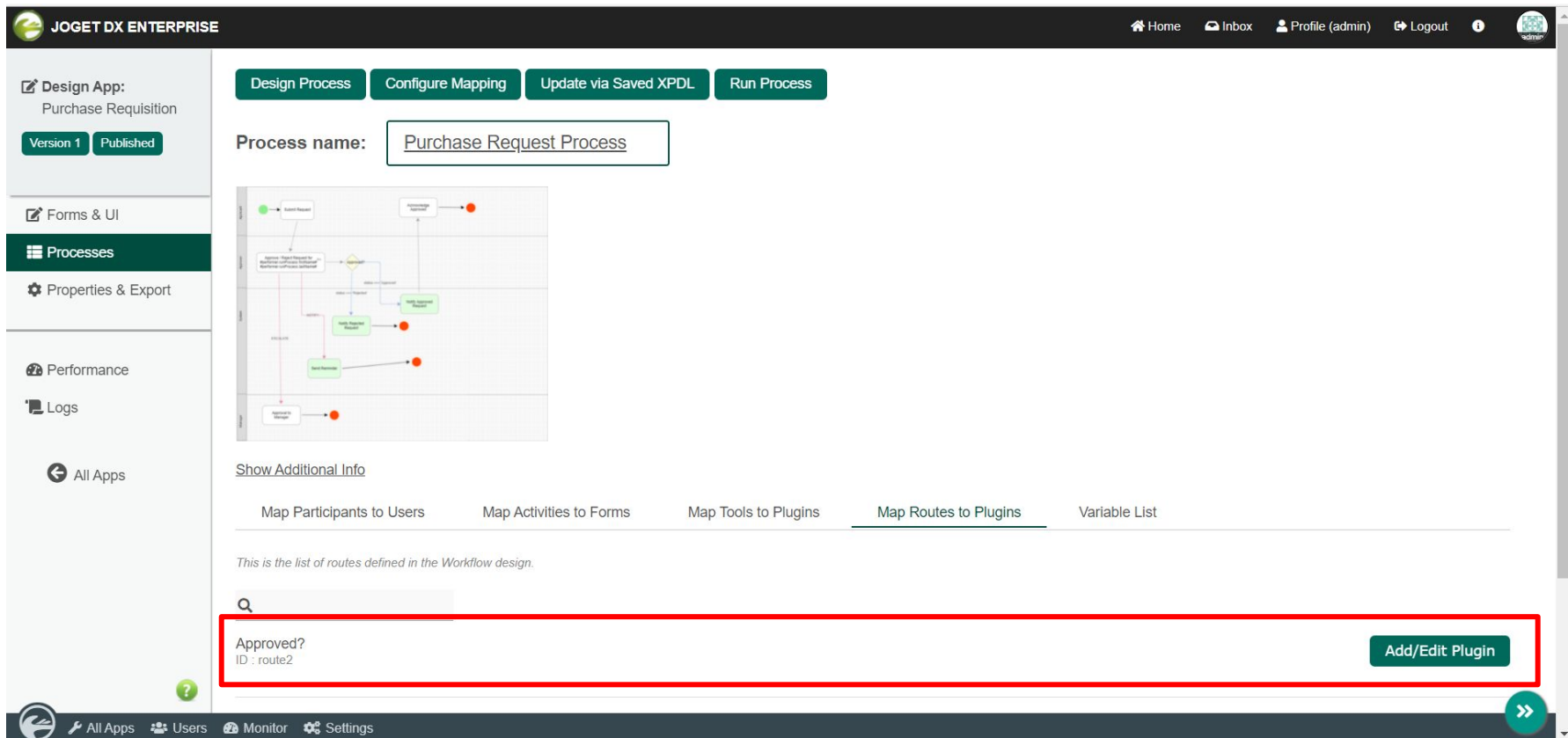
- DESIGN FORM**: The active tab, showing a visual representation of the form.
- PROPERTIES**: A tab for configuring the form's attributes.
- PREVIEW**: A tab for viewing the form as it will appear to users.
- SAVE**: A tab for saving the form configuration.

The form itself is titled "Request Approval" and contains the following elements:

- Request Details**: A section containing:
 - Name**: A text input field with the value "Admin Admin".
 - Request Date ***: A date picker field with the format "MM/DD/YYYY".
 - Category**: A dropdown menu.
 - Items**: A table with columns for "Name", "Quantity", and "Price".
 - Remarks**: A text area for additional comments.
- Section**: A section containing a "Color Picker Custom" field.
- Approval**: A field at the bottom, highlighted with a red box, containing two radio buttons: "Approved" and "Rejected".

Solution

- Go to the process design, edit the approval route. Select the Simple Rules Decision Plugin.



The screenshot displays the Joget DX Enterprise interface. The top navigation bar includes 'Home', 'Inbox', 'Profile (admin)', and 'Logout'. The left sidebar contains navigation options: 'Design App: Purchase Requisition', 'Forms & UI', 'Processes', 'Properties & Export', 'Performance', 'Logs', and 'All Apps'. The main content area shows the 'Purchase Request Process' design. At the top, there are buttons for 'Design Process', 'Configure Mapping', 'Update via Saved XPDL', and 'Run Process'. Below these, the 'Process name' is 'Purchase Request Process'. A workflow diagram is visible, showing a sequence of activities and decision points. Below the diagram, there are tabs for 'Map Participants to Users', 'Map Activities to Forms', 'Map Tools to Plugins', 'Map Routes to Plugins', and 'Variable List'. The 'Map Routes to Plugins' tab is selected. A search bar is present, and a list of routes is shown. One route, 'Approved?' with ID 'route2', is highlighted with a red box. To the right of this entry is an 'Add/Edit Plugin' button. The bottom navigation bar includes 'All Apps', 'Users', 'Monitor', and 'Settings'.

Solution

3) Configure accordingly.

PLUGIN CONFIGURATION - APPROVED? (ROUTE2)
✕

Configure Simple Rules Decision ?
⊞

Rules *

IF 🗑️ | ☰

! And ▾ + Add Condition + Add Group

!

Equal To ▾

🗑️

THEN + Add Action

▾

▾

🗑️

+ Add Rule

ELSE THEN + Add Action

▾

▾

🗑️

Submit
Change Plugin

Solution - Done!

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.

Chapter Review

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

Chapter 4

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

Solution

- Create a form to 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: #form.**tableName**.datePickerFieldId#

Solution

3) Configure accordingly.

Chapter 5

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.

Solution

- 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 Home > / Manage Rule

Admin Admin admin 10 Show

<input type="checkbox"/>	PERSON	PRIORITY	RULE	
<input type="checkbox"/>	admin	10	"#form approver_requests.name#" == "Sai Parn"	Edit
<input type="checkbox"/>	cat	10	parseFloat("#form approver_requests.amount#") == 500	Edit
<input type="checkbox"/>	clark	20	"#form approver_requests.region#" == "US"	Edit
<input type="checkbox"/>	etta	20	"#form approver_requests.region#" == "SEA"	Edit
<input type="checkbox"/>	julia	30	"#currentUser.username#" == "admin"	Edit

5 items found, displaying all items.

APPROVER RULES Home > / Manage Rule

Admin Admin admin

Welcome

Manage Rule 5

Application Approval Process

Manage Application 8

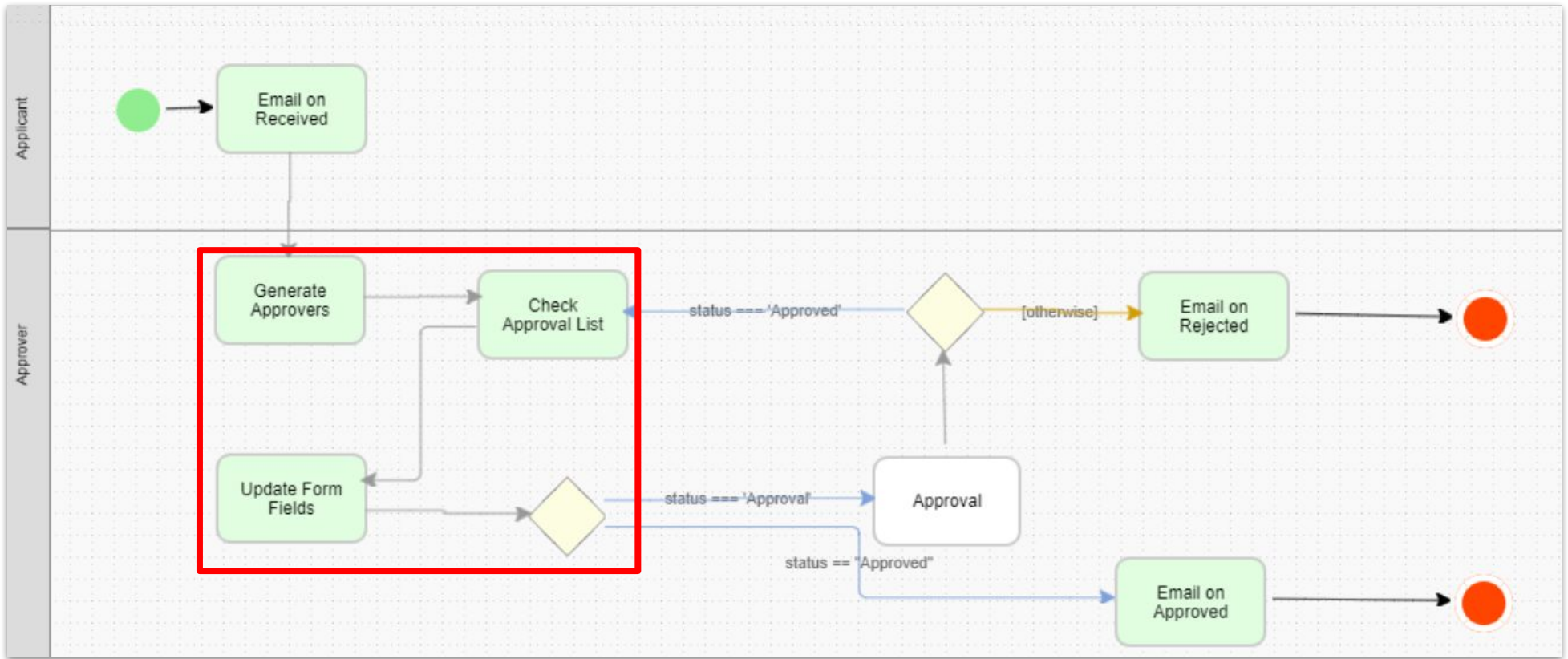
Rule

Rule

Person

Priority

Solution



Materials

- Check Out the completed app to study its code.
- Completed app is available at **19.5.jwa**



Chapter 6

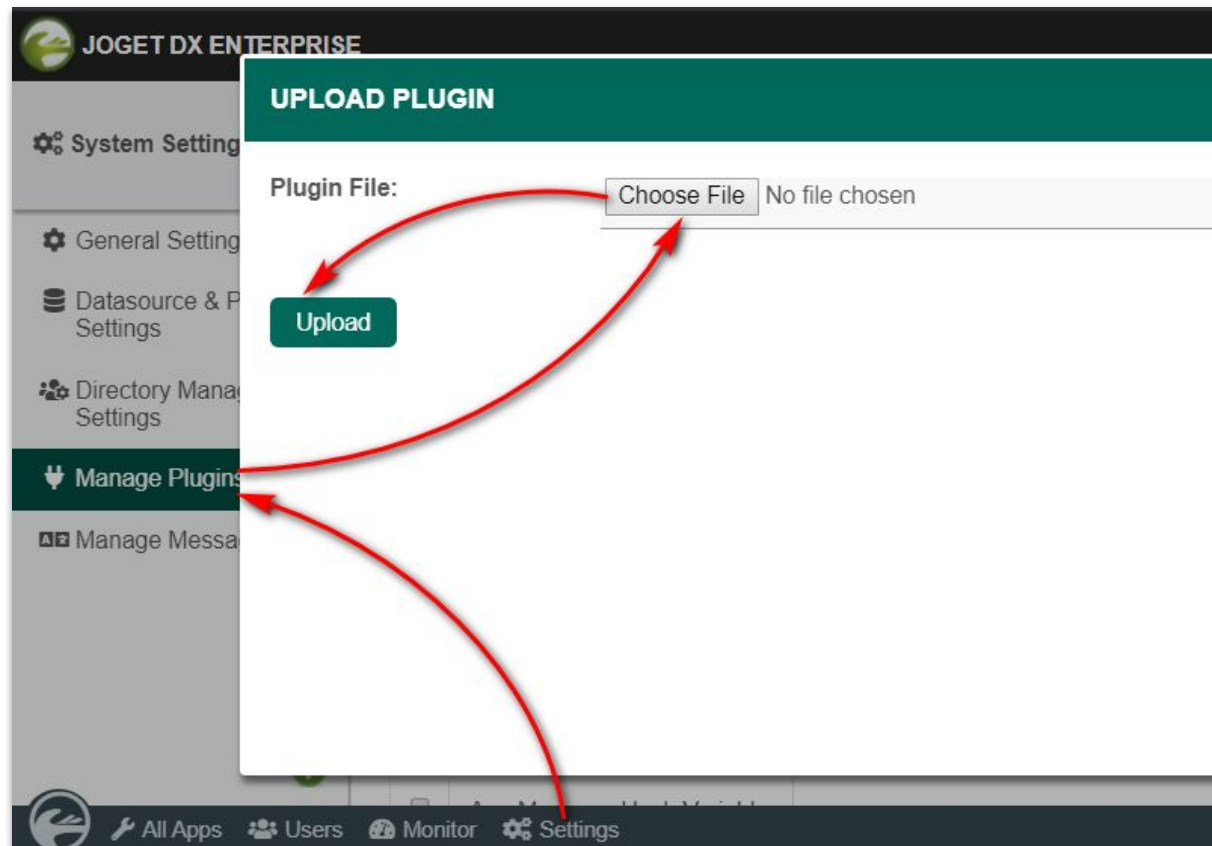
Process Enhancement Plugin

Process Enhancement Plugin

1. Set of 4 plugins to automatically generate approval/rejection action buttons in the datalist 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 Plugin

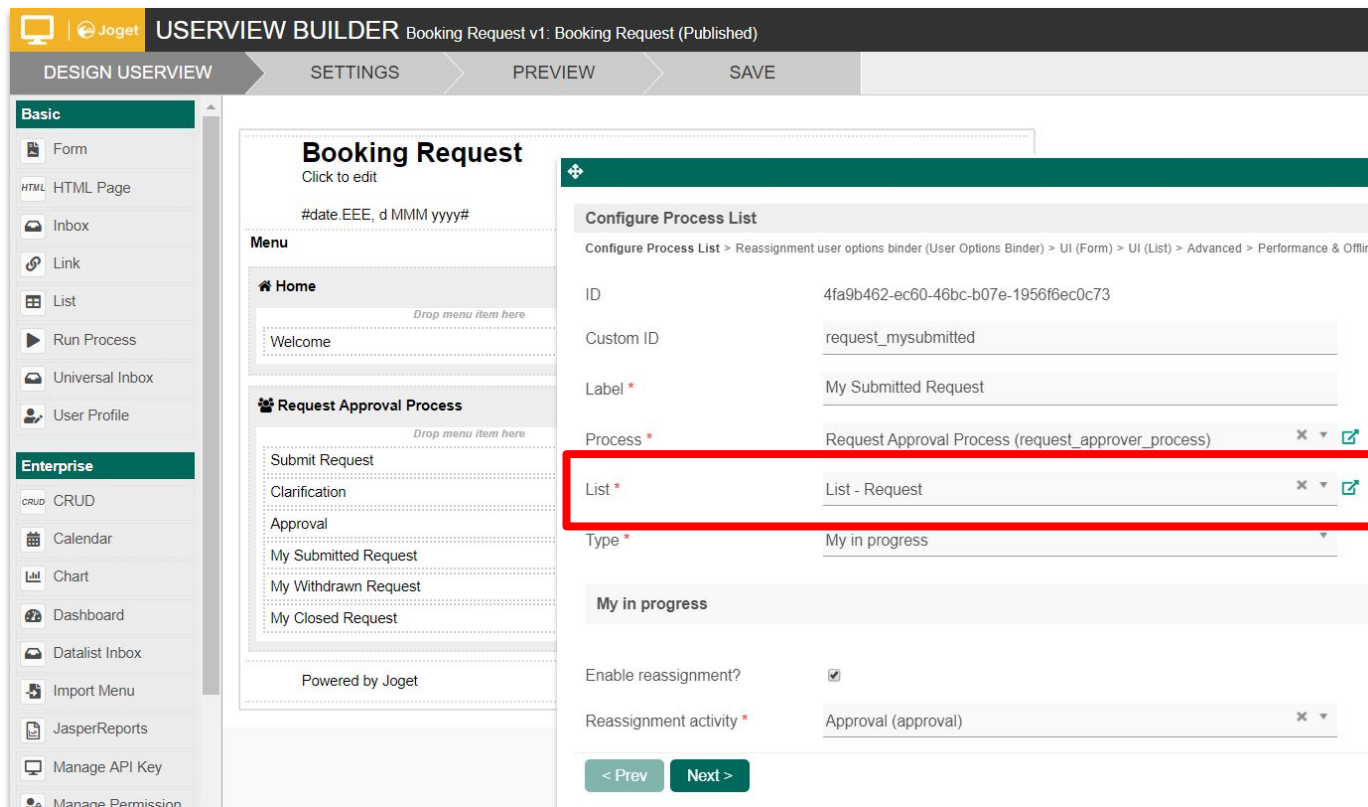


Exercise

1. Create a new App 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.**

Process List

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



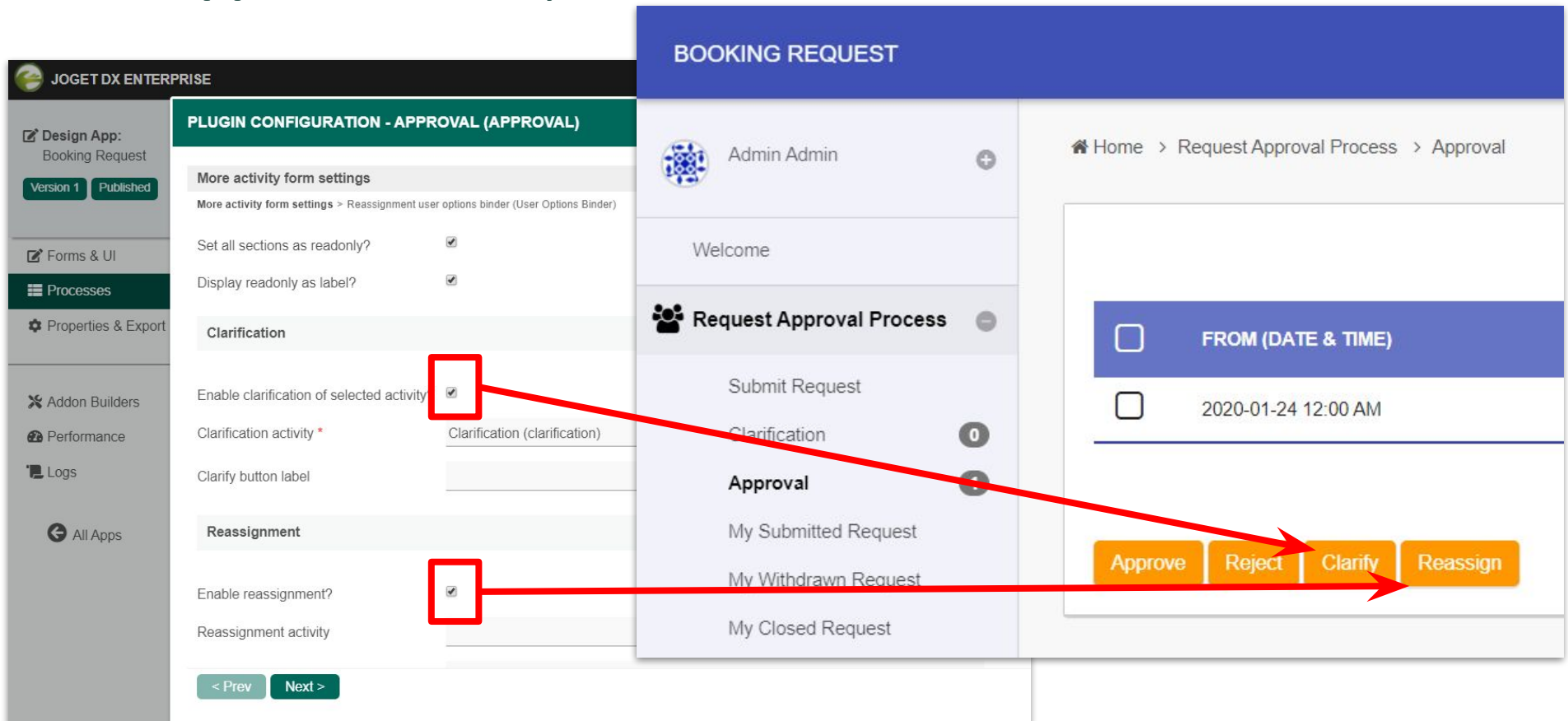
The screenshot shows the 'USERVIEW BUILDER' interface for a 'Booking Request' process. The main configuration area is titled 'Configure Process List' and includes the following fields:

- ID:** 4fa9b462-ec60-46bc-b07e-1956f6ec0c73
- Custom ID:** request_mysubmitted
- Label *:** My Submitted Request
- Process *:** Request Approval Process (request_approver_process)
- List *:** List - Request (highlighted with a red box)
- Type *:** My in progress
- Enable reassignment?:**
- Reassignment activity *:** Approval (approval)

The left sidebar shows a menu structure with 'Home' and 'Request Approval Process' sections. The 'Request Approval Process' section includes items like 'Submit Request', 'Clarification', 'Approval', 'My Submitted Request', 'My Withdrawn Request', and 'My Closed Request'. The bottom of the interface has 'Powered by Joget' and navigation buttons for '< Prev' and 'Next >'.

Map Activities to Form - More Settings

- Go to Processes > Map Activities to Form > More Settings in the **Approval** Activity.



The image displays two overlapping screenshots from the Joget DX Enterprise interface. The left screenshot shows the 'PLUGIN CONFIGURATION - APPROVAL (APPROVAL)' settings page. The right screenshot shows a 'BOOKING REQUEST' form with an approval process mapped to it.

Left Screenshot: PLUGIN CONFIGURATION - APPROVAL (APPROVAL)

- More activity form settings**
 - More activity form settings > Reassignment user options binder (User Options Binder)
 - Set all sections as readonly?
 - Display readonly as label?
- Clarification**
 - Enable clarification of selected activity (highlighted with a red box)
 - Clarification activity * Clarification (clarification)
 - Clarify button label
- Reassignment**
 - Enable reassignment? (highlighted with a red box)
 - Reassignment activity

Right Screenshot: BOOKING REQUEST

- User: Admin Admin
- Welcome
- Request Approval Process (minus icon)
- Submit Request
- Clarification (0)
- Approval** (minus icon)
- My Submitted Request
- My Withdrawn Request
- My Closed Request

Form View:

- Home > Request Approval Process > Approval
- FROM (DATE & TIME)
- 2020-01-24 12:00 AM
- Buttons: Approve, Reject, Clarify, Reassign

Red arrows indicate the mapping: one arrow points from the 'Enable clarification of selected activity' checkbox to the 'Clarify' button, and another points from the 'Enable reassignment?' checkbox to the 'Reassign' button.

Chapter Review

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

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
- community.joget.org
- twitter.com/jogetworkflow
- facebook.com/jogetworkflow
- youtube.com/jogetworkflow