

RADS

Designing your first Process

<https://github.com/radservice/rad-community/fork>

Prerequisites

- Installed RADS.
- Understand various components of RADS.
- General understanding of RADS as an end user.

Content

1. Business Process Design (BPE)
2. Designing your First Process with Process Builder
3. Running your First Automated Process
4. Participant Mapping
5. Process Monitoring

A short horizontal bar with a black left half and an orange right half.

Chapter I

Business Process Design (BPE)

Business Process Management

- BPM is a management approach focused on aligning all aspects of an organization with the wants and needs of clients.
- Wants & needs = business goals.
- BPM attempts to improve processes continuously.

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

Identify a Process



- E.g:
 - Purchase Request Process
 - Purchase Order Process

Note: Let's start simple by focusing **only** on **purchase request process** first.

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart


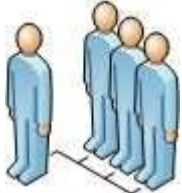

Start & End of Purchase Request Process

- **Start:** 
 - Applicant submits **purchase request** form
- **End:** 
 - Applicant informed with **purchase request** result

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

Process Participants






-  Request applicant.
-  Request approver.
-  Purchasing department (PD).

Note: The approver can also be the same participant that handles the purchase orders.

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

As-Is Process Activities

-  Applicant **submits** purchase request form to PD.
-  PD staff **checks** available stock.
-  PD staff **approves / rejects** purchase request.
-  PD staff **informs applicant** on the request status.
-  PD staff **escalates** approved requests to the purchaser in the department to perform purchase ordering.

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

Business Rules

1. If **requested item in stock is low**:
 - Approver processes purchase requestElse if **many more stock is available**:
 - Reject purchase request

2. If purchase request is **approved**:
 - Escalate approved request to purchaser to handle ordering.
 - Inform applicant on the approved requestElse if purchase request is **rejected**:
 - Inform applicant on the rejected request

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

To-Be Process Activities

As-Is

1. Applicant **submits purchase request** form to PD
2. PD **checks available stock**
3. PD **informs** applicant on **existing abundance of stock**
4. PD **approves / rejects** purchase request
5. PD **informs** applicant on **request status**
6. PD **escalates** approved requests **to purchaser** to handle purchase ordering

To-Be

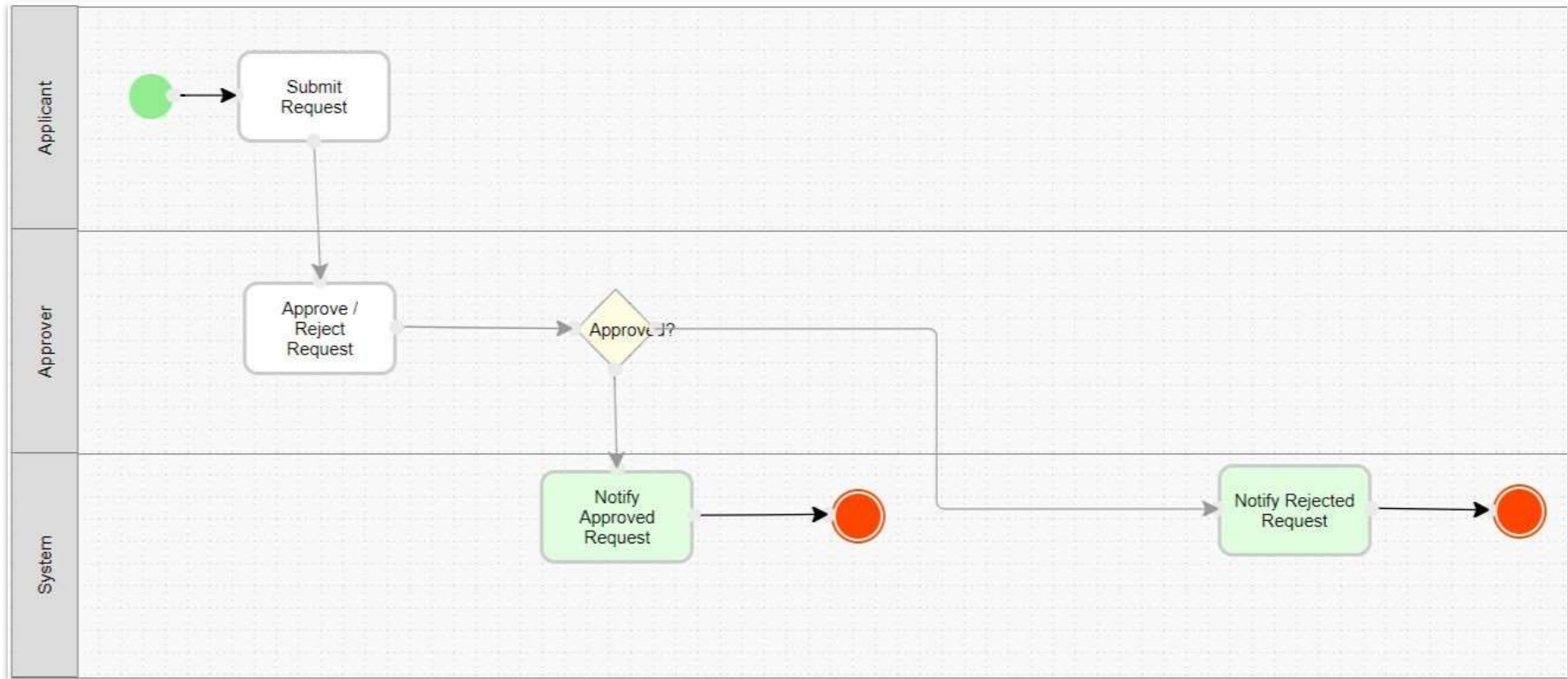
1. Applicant **submits purchase request** form to PD
2. System **checks available stock**
3. System **notifies** applicant on **existing abundance of stock**
4. PD **approves / rejects** purchase request
5. System **notifies** *applicant* on **request status & purchaser** on **new approved request** via email

improved

Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

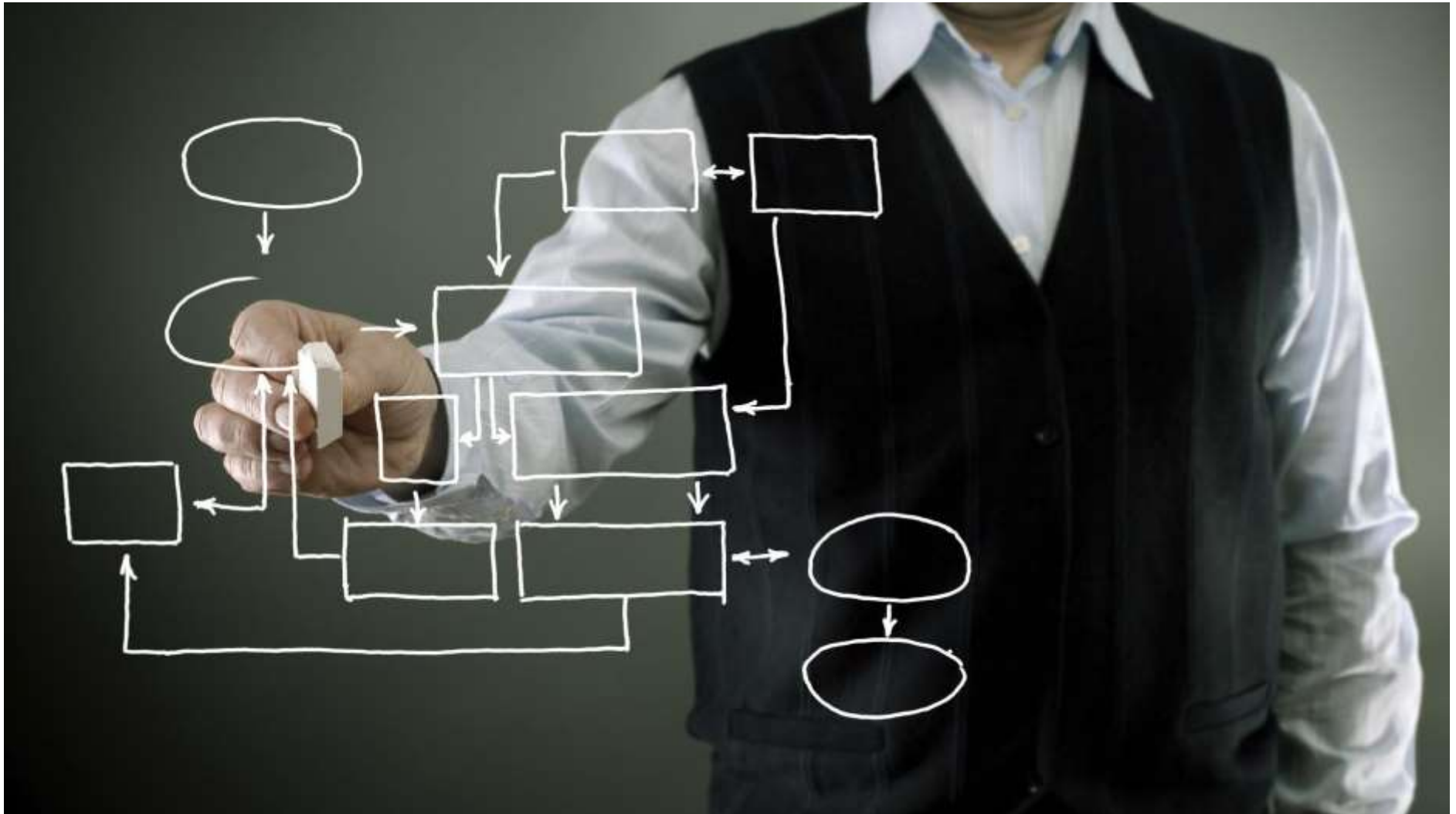
To-Be Purchase Request Process



Business Process Engineering

1. Identify a **process**
2. Identify **start** and **end** of the process
3. Identify process **participants**
4. Identify **as-is** process **activities**
5. Identify **business rules**
6. Identify **areas for optimization**: process activities and business rules that can be:
 - Improved
 - Eliminated
7. Draw the improved / **to-be** process as flowchart

Now, Identify a Process, and Engineer It!



Sample Process Statement

- Jane needs to get some stationeries.

The request that she submits must be first approved by PD staff before being sent to another respective staff in the department to handle the order.

The department will then notify Jane accordingly.

Hands-On: Business Process Engineering

1. Process
Name

2. Start and
End

3. Participants

4. As-Is
Activities

5. Business
Rules

7. To-Be
Process

6. Optimize

Chapter 1 Review

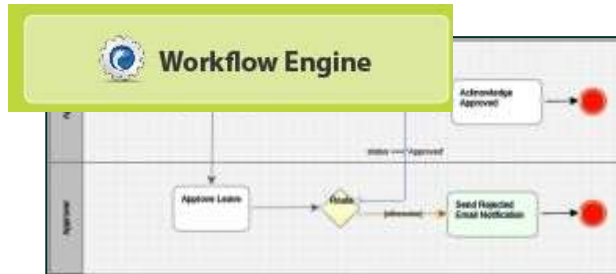
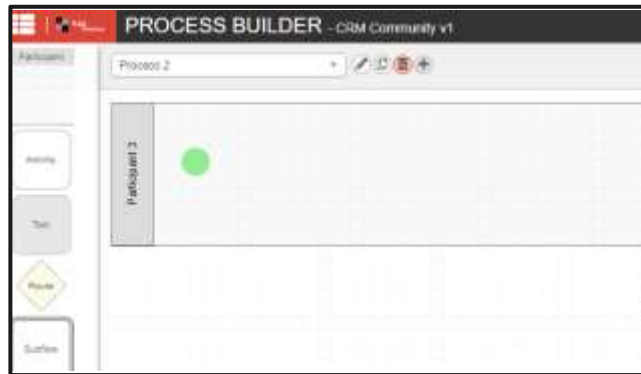
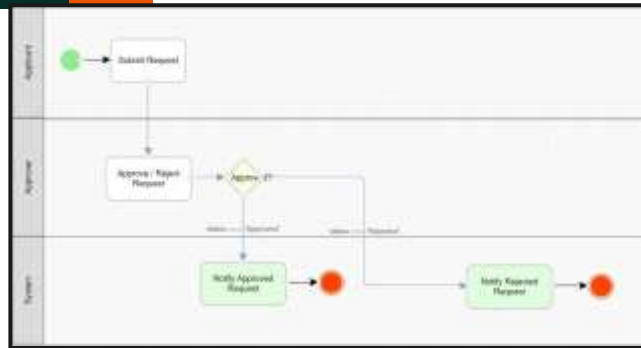
- General understanding on how to best design a Process.
- With the finished process design, one can now bring the finished process design/flow and implement it in RADS.



Chapter 2

Designing your first Process with Process Builder

Business Process Automation



Business Process Design



Process Design



Process

Create Your First App

- Design Apps -> Design New App

DESIGN NEW APP

App Details

App ID | purchaseRequisition

App Name | Purchase Requisition

Show Advanced Options

Save Cancel

Naming convention:
Use **camelCase** for better legibility for App ID, Process ID, and Activity ID

Naming Convention

- Use **camel case** for App ID, Process ID, Activity ID.
For example:-
 - **purchaseRequisition**
 - **purchaseRequestProcess**
 - **submitRequest**
- Do not use **snake case** as it conflicts with process instance ID naming convention:-
1_purchaseRequisition_purchaseRequestProcess_submitRequest
- Maintain consistent naming convention that you have decided across the board.

Create Your First App

- Purchase Requisition Application
 - Purchase Request Process
 - Applicant **submits**
 - Approver **approves**
 - Applicant is **notified** of the request result

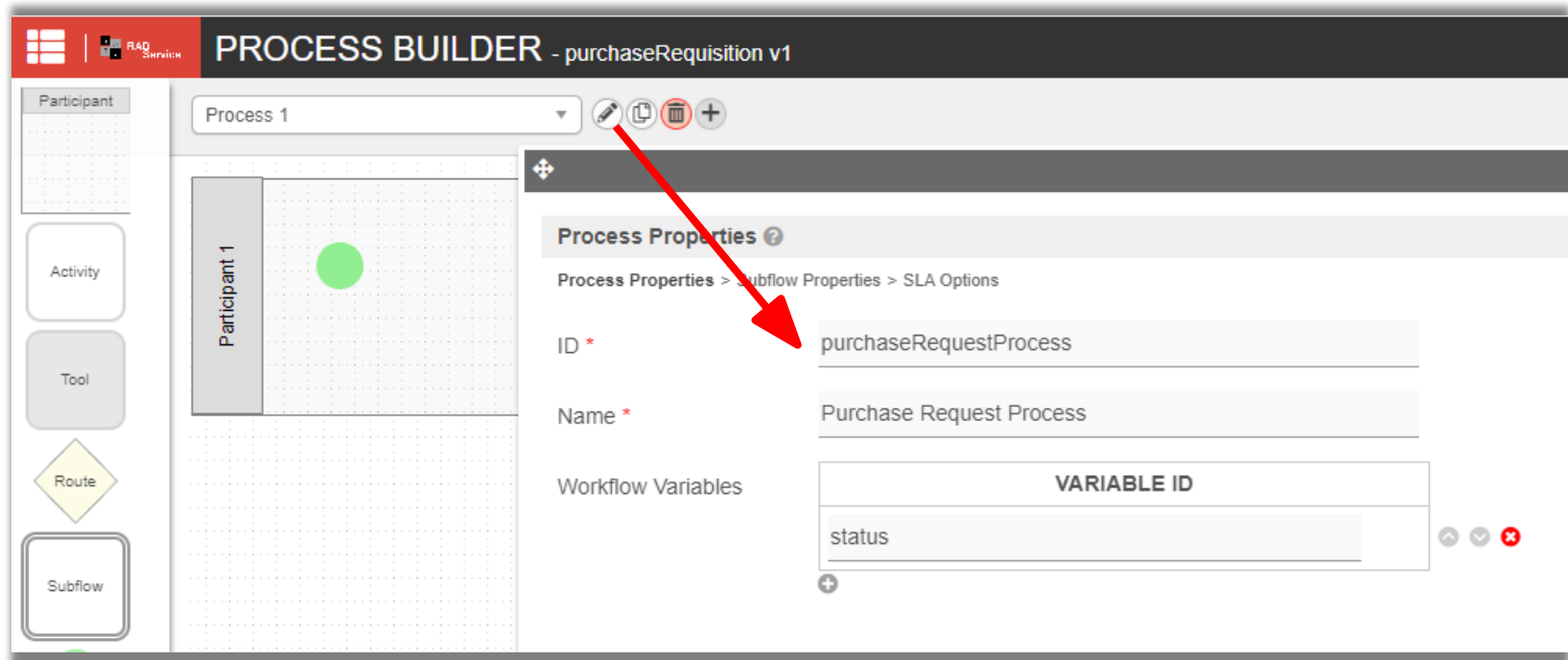
Design Your First Process

- Launch Process Builder – **Processes > Design Processes**

The screenshot displays the RAD Service interface. On the left, a sidebar contains four main sections: 'Design App: purchaseRequisition' (with 'Version 1' and 'Not Published' buttons), 'Forms & UI', 'Processes' (highlighted with a red box), and 'Properties & Export'. The 'Processes' section is further divided into 'Design Processes' (highlighted with a red box) and 'Update via Saved XPD'. A red arrow points from the 'Design Processes' button to the 'PROCESS BUILDER' window. The 'PROCESS BUILDER' window is titled 'purchaseRequisition v1' and shows a 'Process 1' dropdown menu. Below the dropdown, there is a grid area with a green circle and a vertical bar labeled 'Participant 1'. The bottom of the interface shows a 'Participant' table, an 'Activity' button, a 'Tool' button, and a 'Route' button.

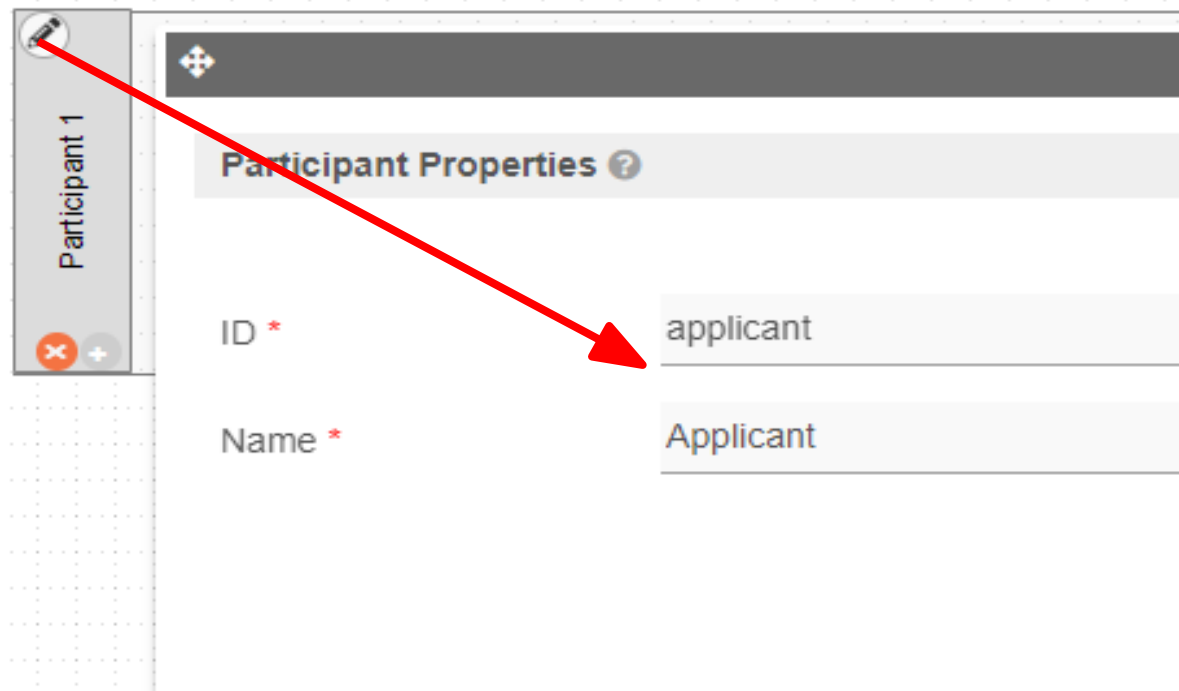
Defining Your Process

- Rename Process ID and Name
 - Use **camel case** for ID - **purchaseRequestProcess**
 - Do not use **snake case** as it conflicts with process instance ID naming convention



Defining Your Participant

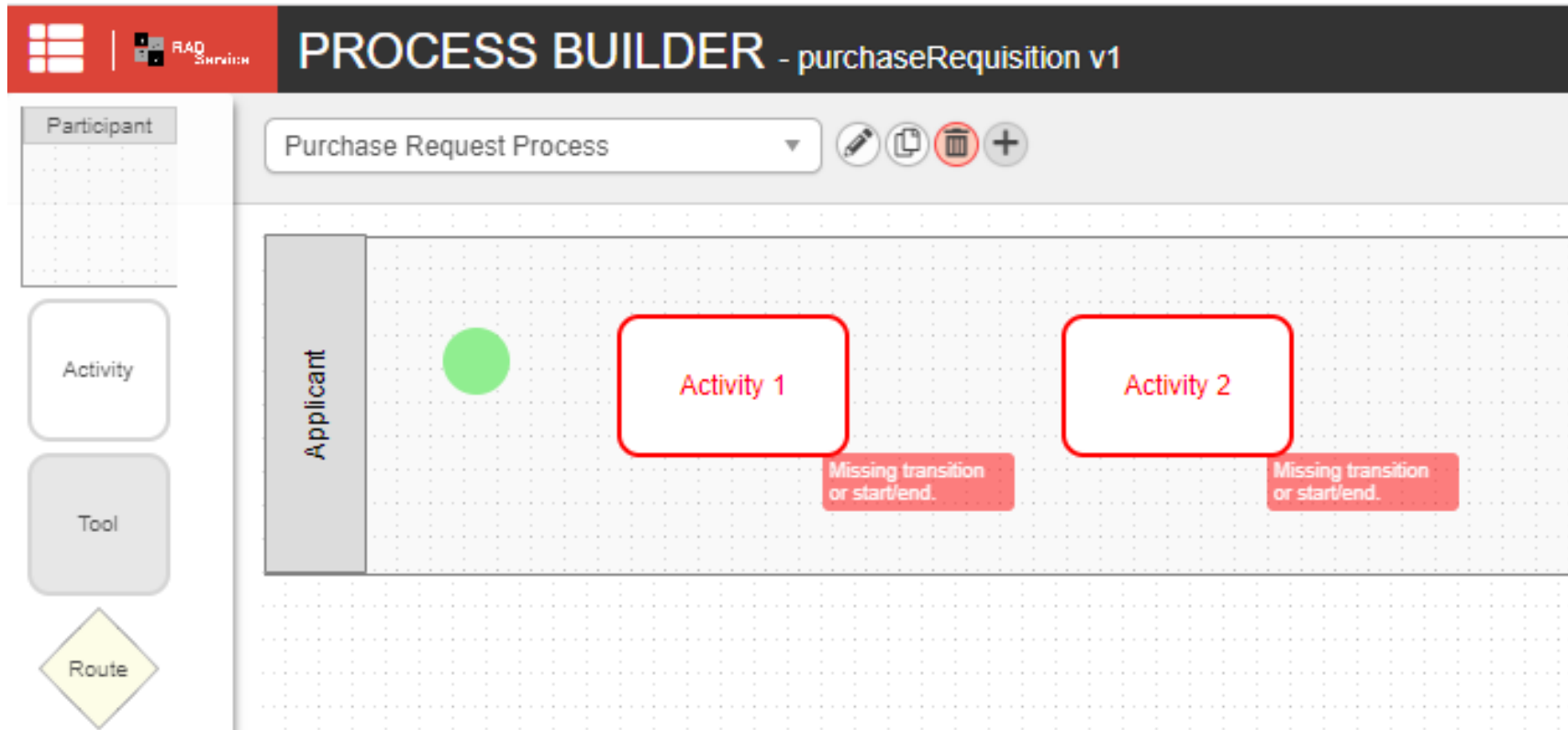
- Rename the existing participant to **Applicant**.



The screenshot shows a software interface for defining a participant. On the left, a vertical panel labeled "Participant 1" contains a red 'X' icon and a '+' icon. A red arrow points from this panel to the "ID" field in the "Participant Properties" dialog box. The dialog box has a title bar with a question mark icon and the text "Participant Properties ?". It contains two input fields: "ID *" with the value "applicant" and "Name *" with the value "Applicant".

Smart Guide

- The Process Builder will highlights logical errors as you design.

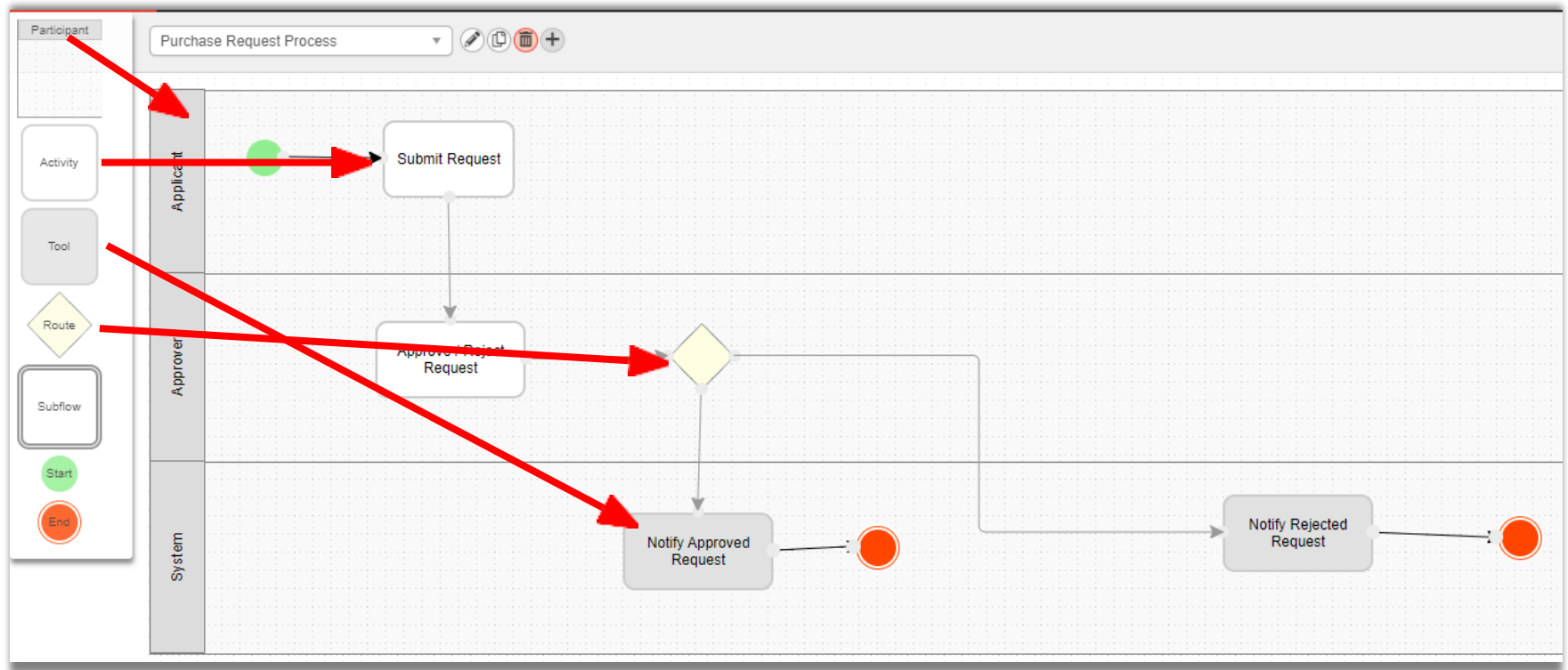


About Smart Guide

- Advises only on possible design errors.
- Will **NOT** advise on flawed business logic, design inefficiency.

Insert Activities and Transitions

- Drag and drop, select the appropriate nodes to build the Process.



Rename Activity ID and Name

- Rename the activity node to make it meaningful.
- Use **camelCase** naming convention for ID (submitRequest).

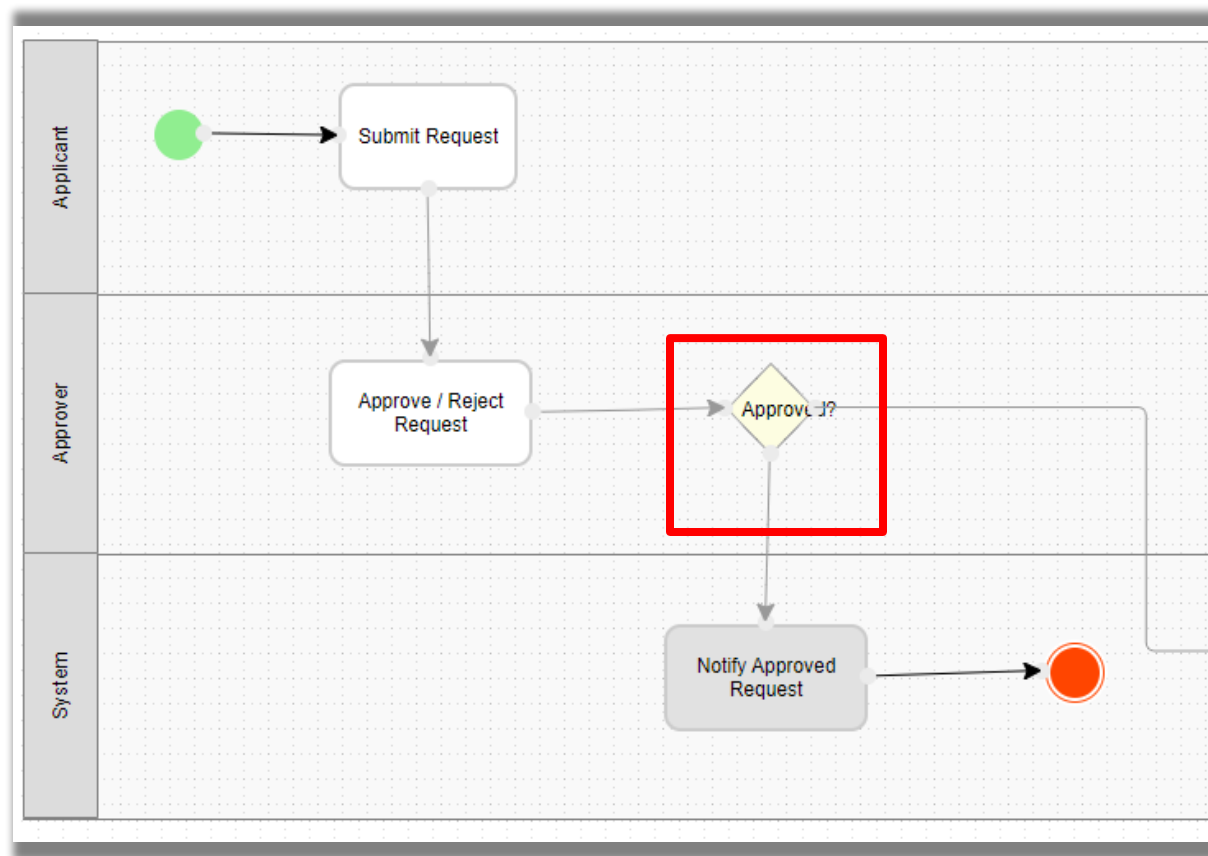
The screenshot shows the RAD IDE interface. On the left, a diagram shows a green circle connected to a rounded rectangle labeled "Submit Request". The left sidebar has a tab labeled "Applicant" and two buttons, a red "X" and a green "+". On the right, the "Activity Properties" panel is open, showing the breadcrumb "Activity Properties > Deadlines > SLA Options". It has two input fields: "ID *" with the value "submitRequest" and "Name *" with the value "Submit Request". A red box highlights the "Naming convention: Use Camel Case for better legibility for App ID, Process ID, Activity ID" text, with a red arrow pointing from it to the "ID" field.

Naming convention:
Use Camel Case for better
legibility for App ID, Process
ID, Activity ID

Activity Properties ?
Activity Properties > Deadlines > SLA Options
ID * submitRequest
Name * Submit Request

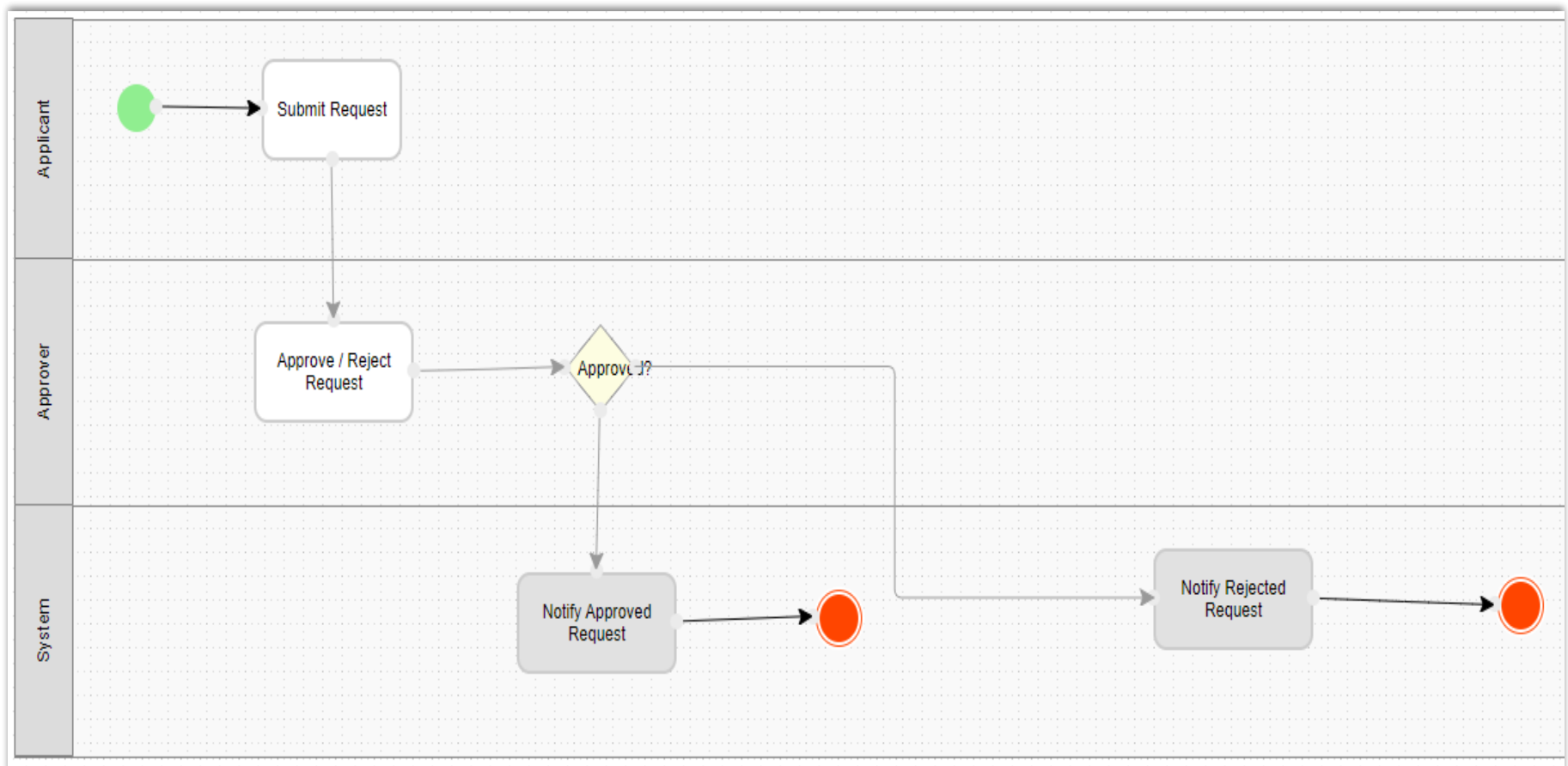
Implementing the Business Rule

- Route: Diamond-shaped node – Used to implement business rules.



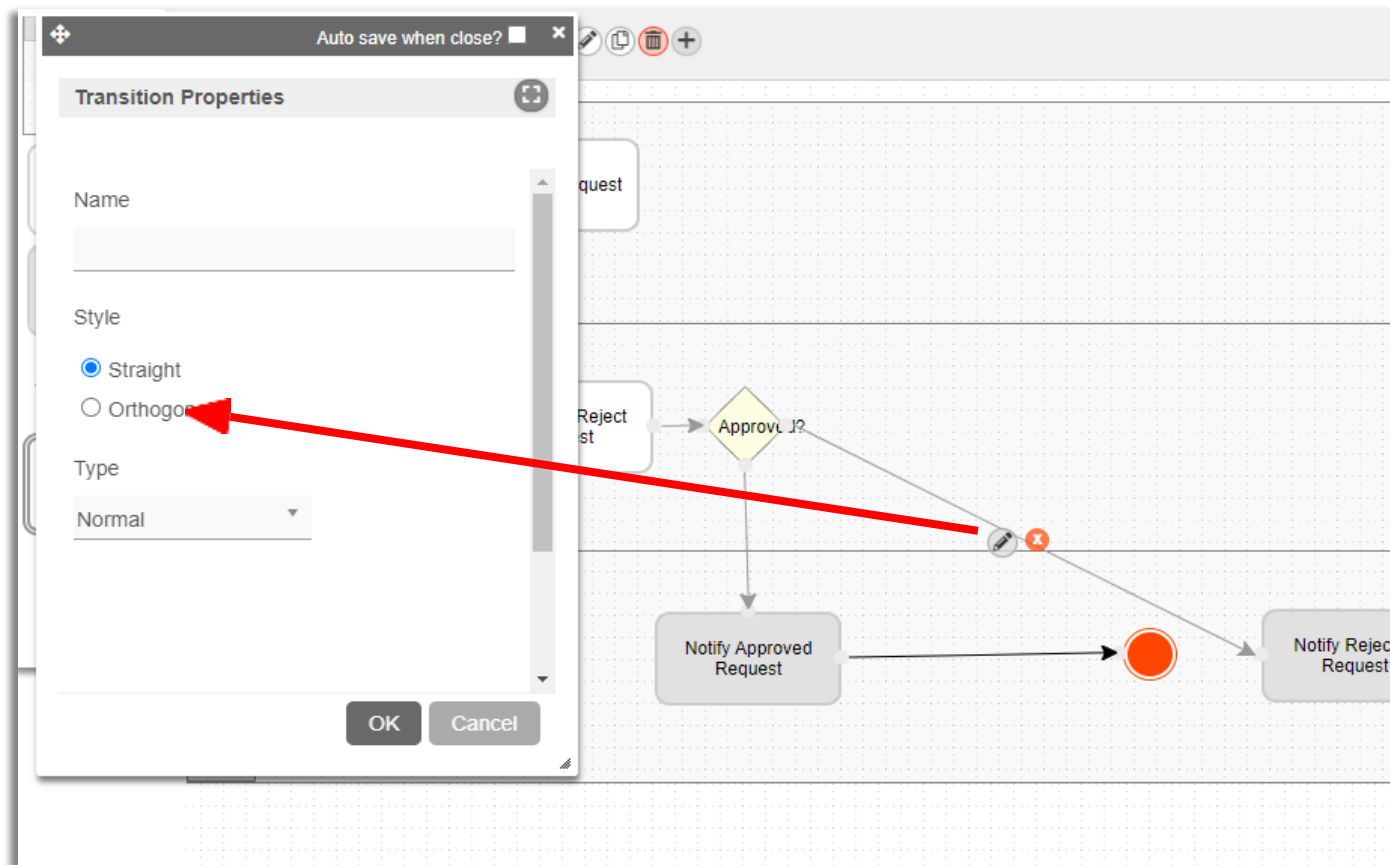
Completing the Design

- Insert the rest of the activities.



Managing the Transition Style

- Set shape to Straight / Orthogonal to shape your transition.



Declaring the Workflow Variable

- Variables used in the Process must be declared in the Process's properties.

The screenshot shows the RAD Service interface for editing a process. The main window is titled 'Purchase Request Process'. On the left, there is a sidebar with a 'Participant' table, an 'Activity' button, a 'Tool' button, a 'Route' button, a 'Subflow' button, and 'Start' and 'End' buttons. The main area is divided into three sections: 'Applicant', 'Approver', and 'System'. The 'Process Properties' dialog is open, showing the following fields:

- ID ***: purchaseRequestProcess
- Name ***: Purchase Request Process

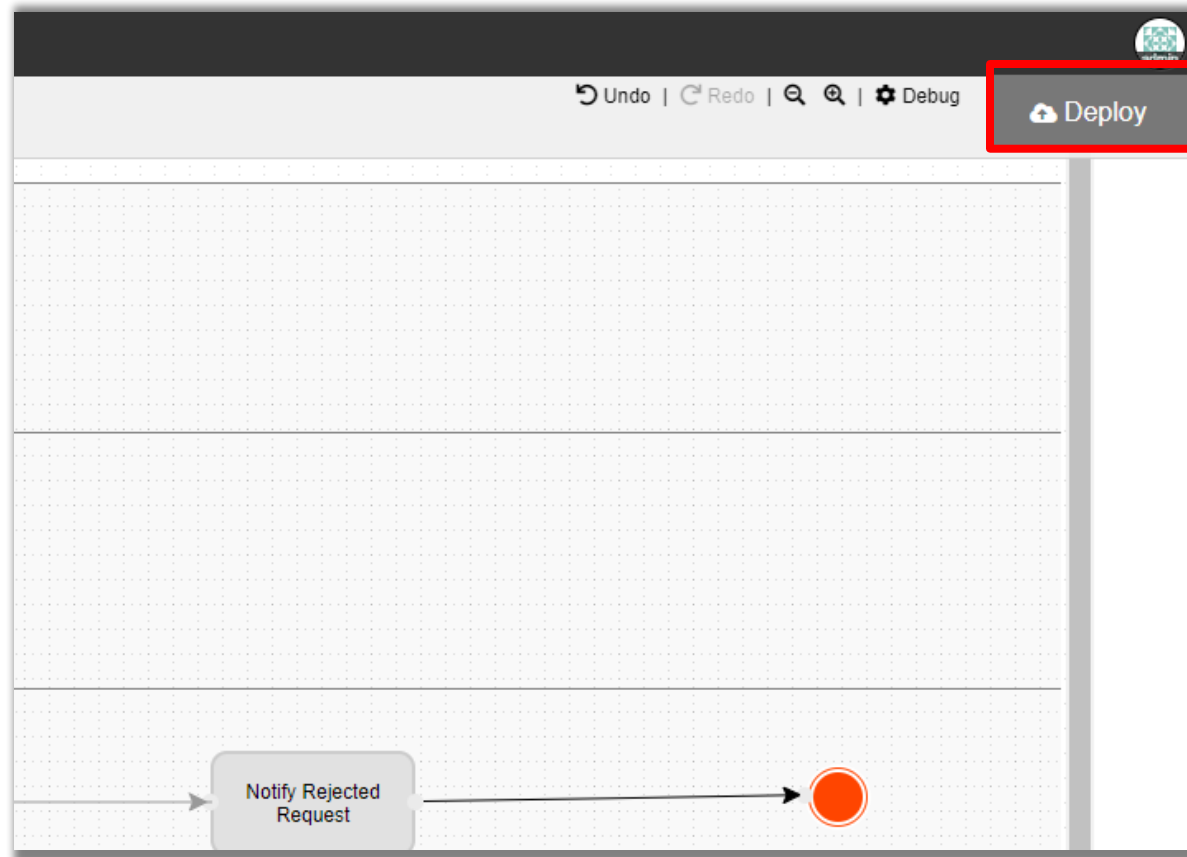
The 'Workflow Variables' section is highlighted with a red box. It contains a table with the following structure:

VARIABLE ID
status

At the bottom of the dialog, there are buttons for '< Prev', 'Next >', 'OK', and 'Cancel'.

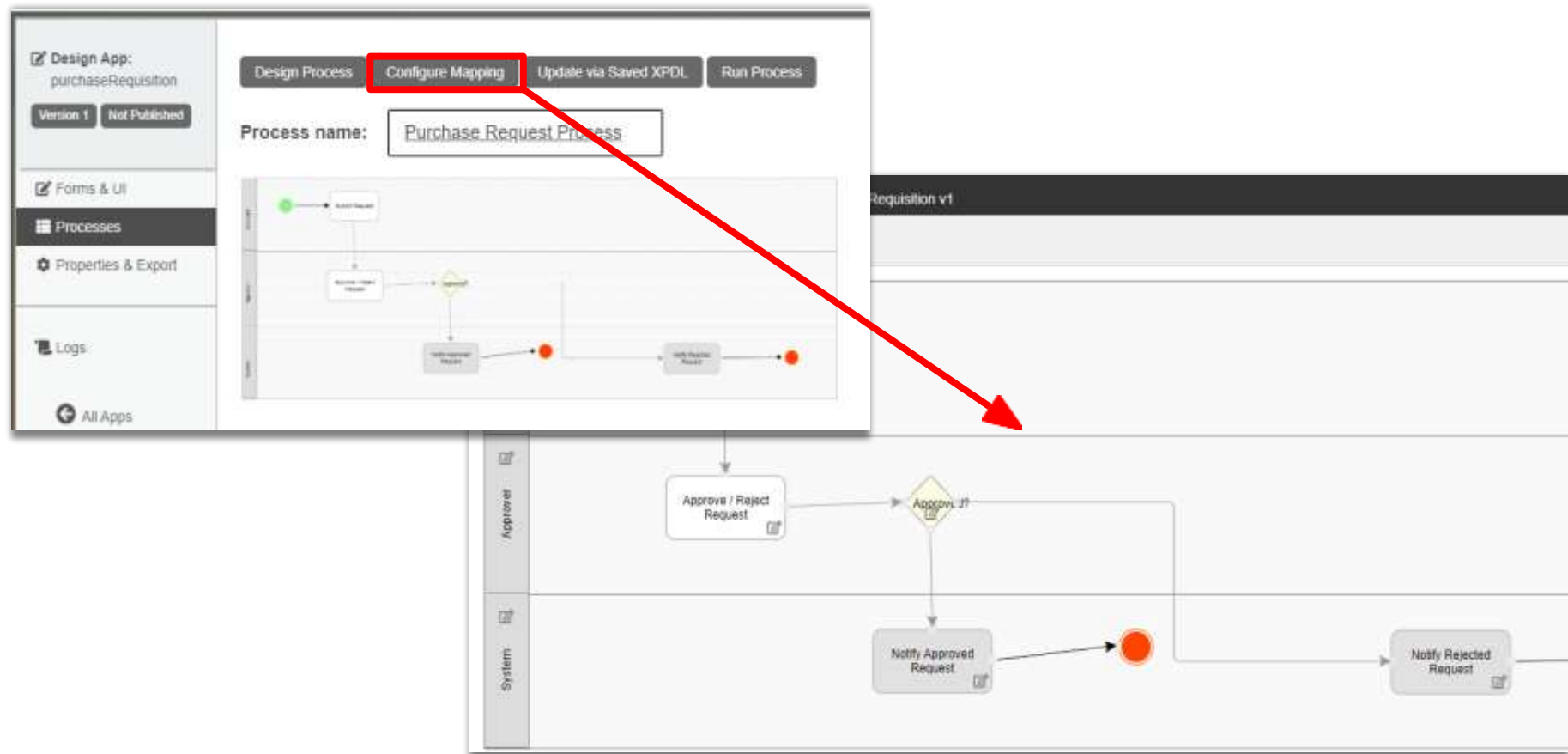
Deploying Your Process Design

- On the top right corner of the designer, click on “Deploy” to push your finished design back into your RADS App.



Implementing the Business Rules

- Launch Process Mapper.
App Design > Processes > Configure Mapping



Implementing the Business Rules

- Route: Diamond-shaped node – Used to implement business rules.

PROCESS MAPPER - purchaseRequisition v1

Purchase Request Process

Applicant: Submit Request

Approver: Approve / Reject Request

System: Notify Approved Request

MAP ROUTES TO PLUGINS - APPROVED? (ROUTE1)

Search

PLUGIN NAME	PLUGIN DESCRIPTION	PLUGIN VERSION
Bean Shell Decision	Executes standard Java syntax	7.0.0
Simple Rules Decision	Using rules to decide the next transitions for process flow.	7.0.0
Simple TensorFlow AI Decision	Using TensorFlow outputs to construct the rules for process flow next transitions decision.	7.0.0

15 | Page 1 of 1 | Displaying 1 to 3 of 3 items

Implementing the Business Rules

- Choose Simple Rules Decision.

MAP ROUTES TO PLUGINS - APPROVED? (ROUTE1)

Search

PLUGIN NAME	PLUGIN DESCRIPTION
Bean Shell Decision	Executes standard Java syntax
Simple Rules Decision	Using rules to decide the next transitions for process flow
Simple TensorFlow AI Decision	Using TensorFlow outputs to construct the rules for process flow next transitions decision.

15 Page 1 of 1 Displaying 1 to 3 of 3

PLUGIN CONFIGURATION - APPROVED? (ROUTE1)

Configure Simple Rules Decision

Rules *

IF

And

status Equal To

THEN

Transition transition4 (Notify Approved Request)

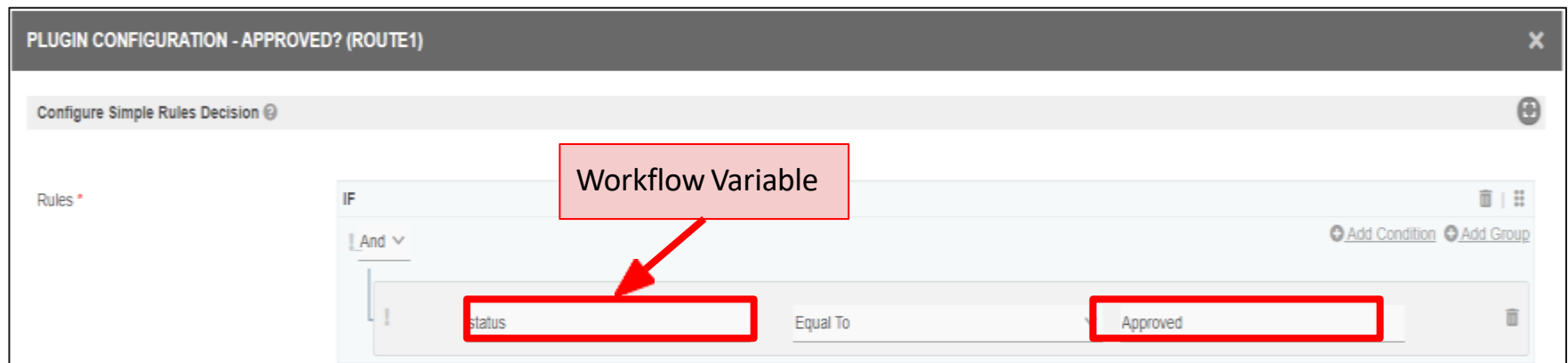
Add Rule

ELSE THEN

Transition transition5 (Notify Rejected Request)

Simple Rules Decision Plugin

- **Hash variable** is supported in input fields as well for complex use cases.



Note: This means you don't even need workflow variables to control the route!

e.g.: using Form Hash variable to read form field value to route

Simple Rules Decision Plugin

- **Hash variable** is supported in input fields as well for complex use cases.



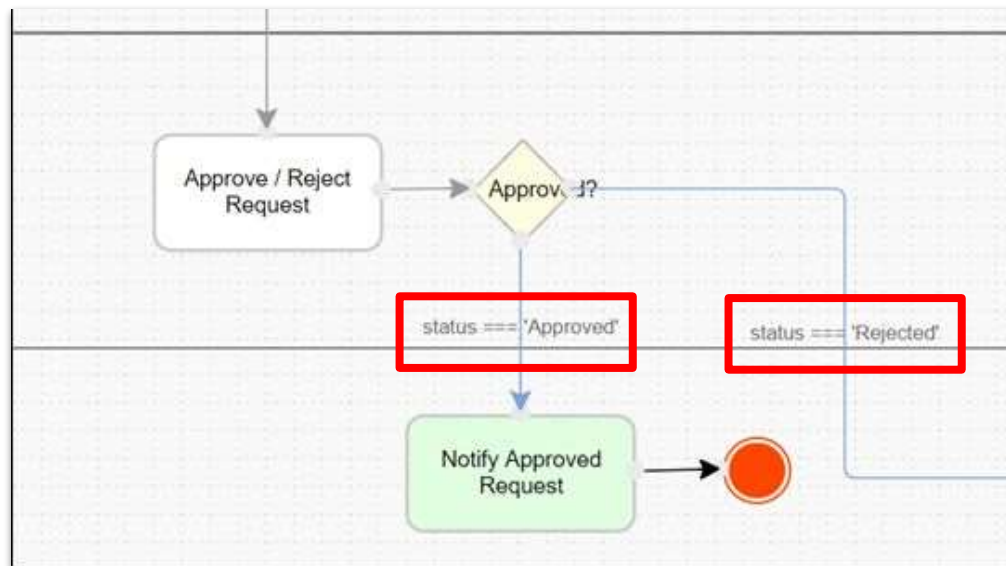
e.g.: using *Form Hash Variable* to read form field value to route
#form.purchase_requests.approval#

If and otherwise...

- Does this resemble the typical programming if-else condition block?
- Is this the best design?
- Why not if-else-if-else everything? (In other words, explicitly set all possible conditions)

Alternative Business Rule Implementation

- Prior to DX, business rules implementation must be done in the Process Builder.
- This approach is **no longer recommended** as it is difficult to change business logic without redeploying the whole process design of the entire app.

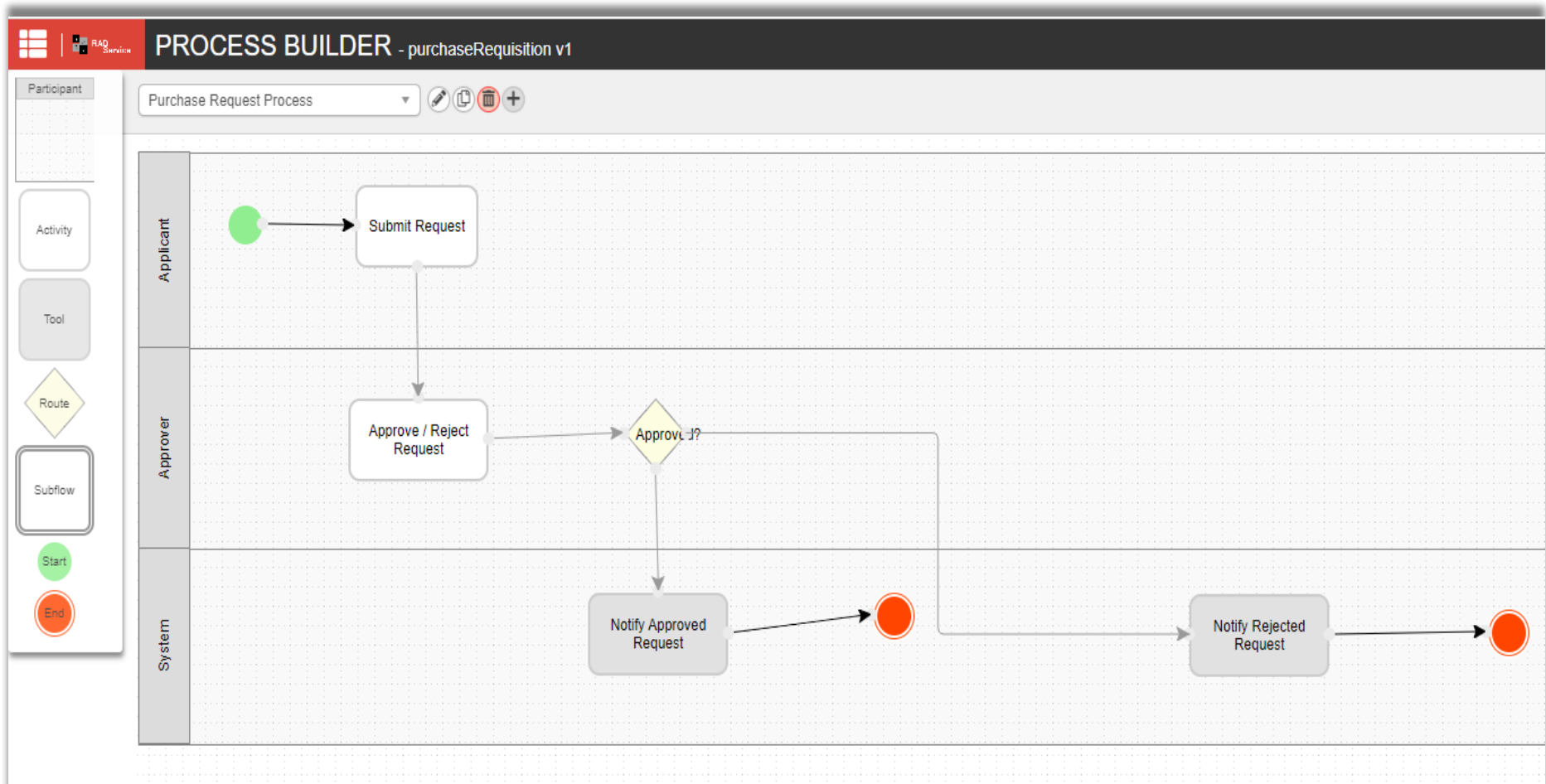


Advantages of using Process Mapper

- No need to constantly deploy the Process Builder when changing the business logic.
- Decision Tool Plugin takes precedence over Process Builder conditional setup.

Alternative Business Rule Implementation

- App Design > Processes > Design Process.



Alternative - Defining the Transition Condition

- When the **status** variable is met with certain condition(s), route accordingly.

The screenshot displays the RAD Service interface. On the left, a BPMN diagram shows a yellow decision diamond labeled "Approved?". One path leads to a rounded rectangle task labeled "Notify Approved Request", which then connects to an orange circular event. Another path from the diamond leads to a red rectangular box containing a pencil icon and a red 'X' icon. A red arrow points from this box to the "Conditions" section of the "Transition Properties" panel on the right.

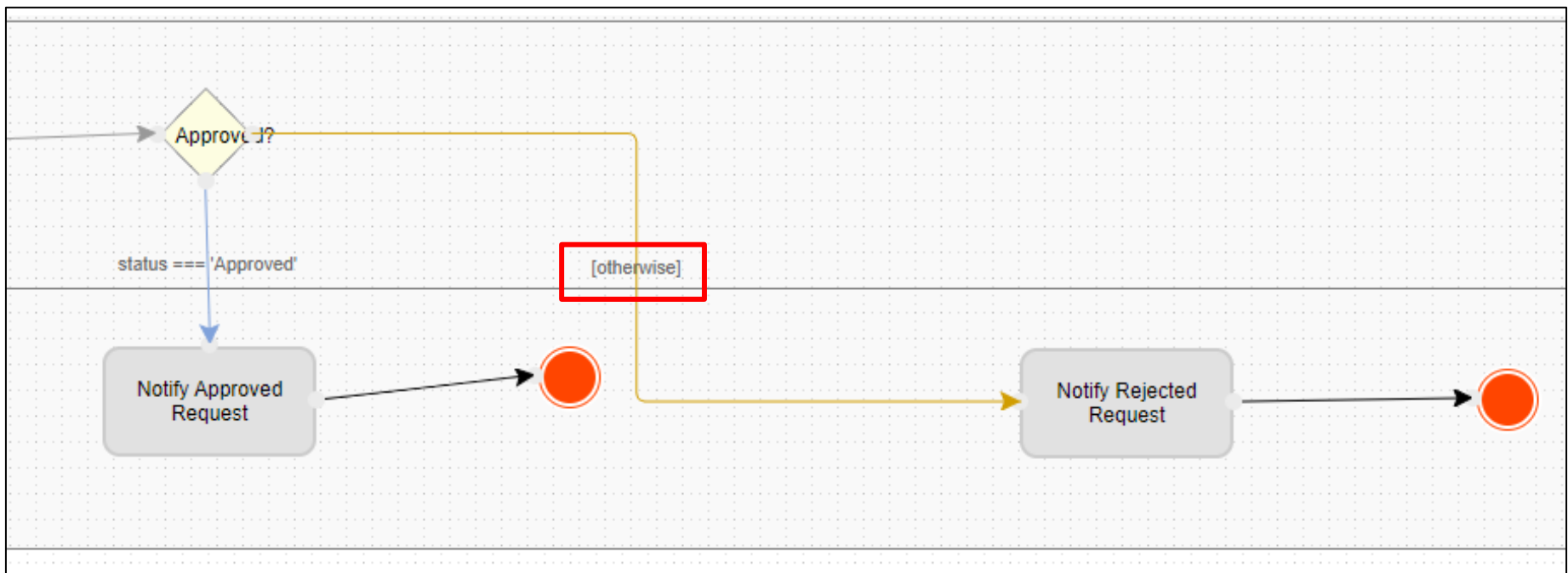
The "Transition Properties" panel includes the following fields:

- Name:
- Style: ☐ Straight, ☒ Orthogonal
- Type: Condition
- Use Condition Helper: Yes
- Conditions:

JOIN	VARIABLE	OPERATION	VALUE
AND	status	Equal To	Rejected

Alternative - Anything else...

- Handling the “otherwise”, if any.



Note: In this case, “otherwise” is also correct. All conditions are set explicitly for educational purposes.

Good to Know

- Expressions in “Condition” are evaluated in JavaScript.
- Workflow variable are stored as data type **string**.

Good to Know

- Native Javascript functions are accepted. For example...
 - status == "Approved" && parseInt(balance) > 0
- Make sure of debugger tool such as **Firebug / Developer Tools** to test the expression before deploying.



Chapter 2 Review

- Create a new App.
- Design and deploy Process design into the App.
- Understand the naming convention.
- Define various kind of process design elements.
- Define workflow variable.
- Deploying process design to RADS.



Chapter 3

Running your First Automated Process

How to Test?

- There are 2 ways in general.
 - Make use of the “**Run Process**” button in App Design > Processes (only for admin)
 - Make use of **Userveview** (for all users)

How to Test?

- There are 2 ways in general.
 - Make use of the “**Run Process**” button in App Design > Processes (only for admin)
 - Make use of **Userview** (for all users)

Verify the Process Using Run Process

- Run through the Process to verify.

The screenshot displays the RAD Service interface for managing a process. On the left sidebar, the 'Processes' section is selected. The main area shows the 'Purchase Request Process' configuration. At the top, there are four buttons: 'Design Process', 'Configure Mapping', 'Update via Saved XPDL', and 'Run Process' (highlighted with a red border). Below these buttons, the 'Process name' is set to 'Purchase Request Process'. The process flowchart is visible, showing the following steps:

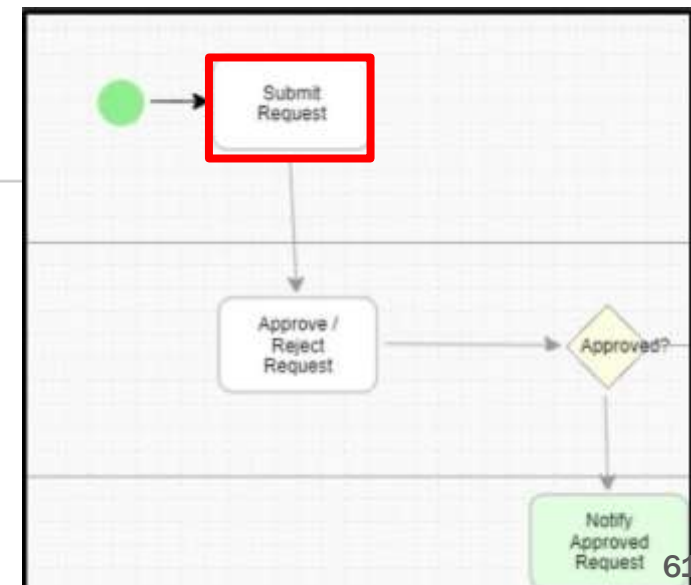
```
graph TD; Start(( )) --> Submit[Submit Request]; Submit --> Approve[Approve / Reject Request]; Approve --> Approved{Approved?}; Approved --> NotifyApproved[Notify Approved Request]; Approved --> NotifyRejected[Notify Rejected Request]; NotifyApproved --> End1(( )); NotifyRejected --> End2(( ))
```

Verify the Process

- Complete the assignment to proceed.

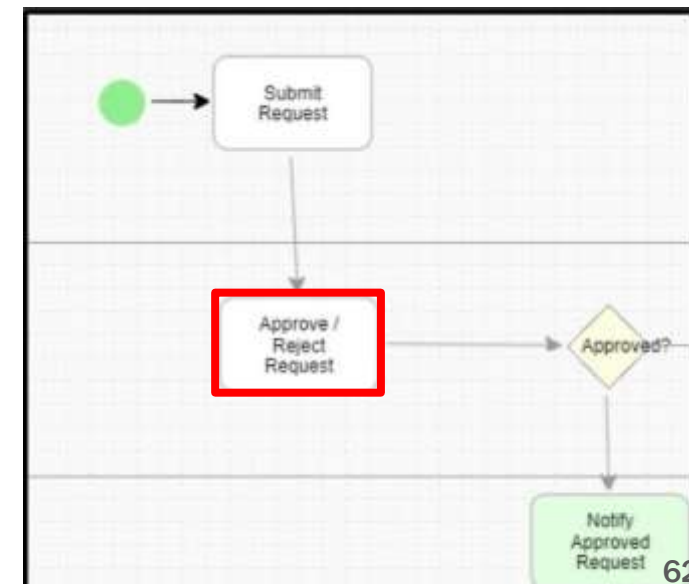
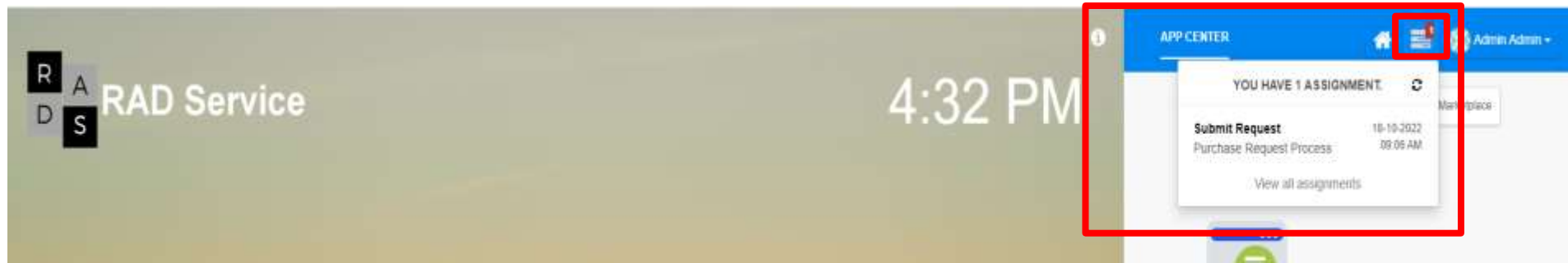
HINT: Activity name is configured in Process Builder

The screenshot shows the RAD Service interface. On the left, a sidebar contains the following menu items: 'Design App: purchaseRequisition' (with a pencil icon), 'Version 1' and 'Published' buttons, 'Forms & UI' (with a pencil icon), 'Processes' (highlighted with a grid icon), and 'Properties & Export' (with a gear icon). The main area displays the title 'PURCHASE REQUEST PROCESS - SUBMIT REQUEST' in a red-bordered box. Below this, there is a 'status' label and a 'Complete' button, also highlighted with a red border. A blue speech bubble points from the 'Complete' button to the hint text.



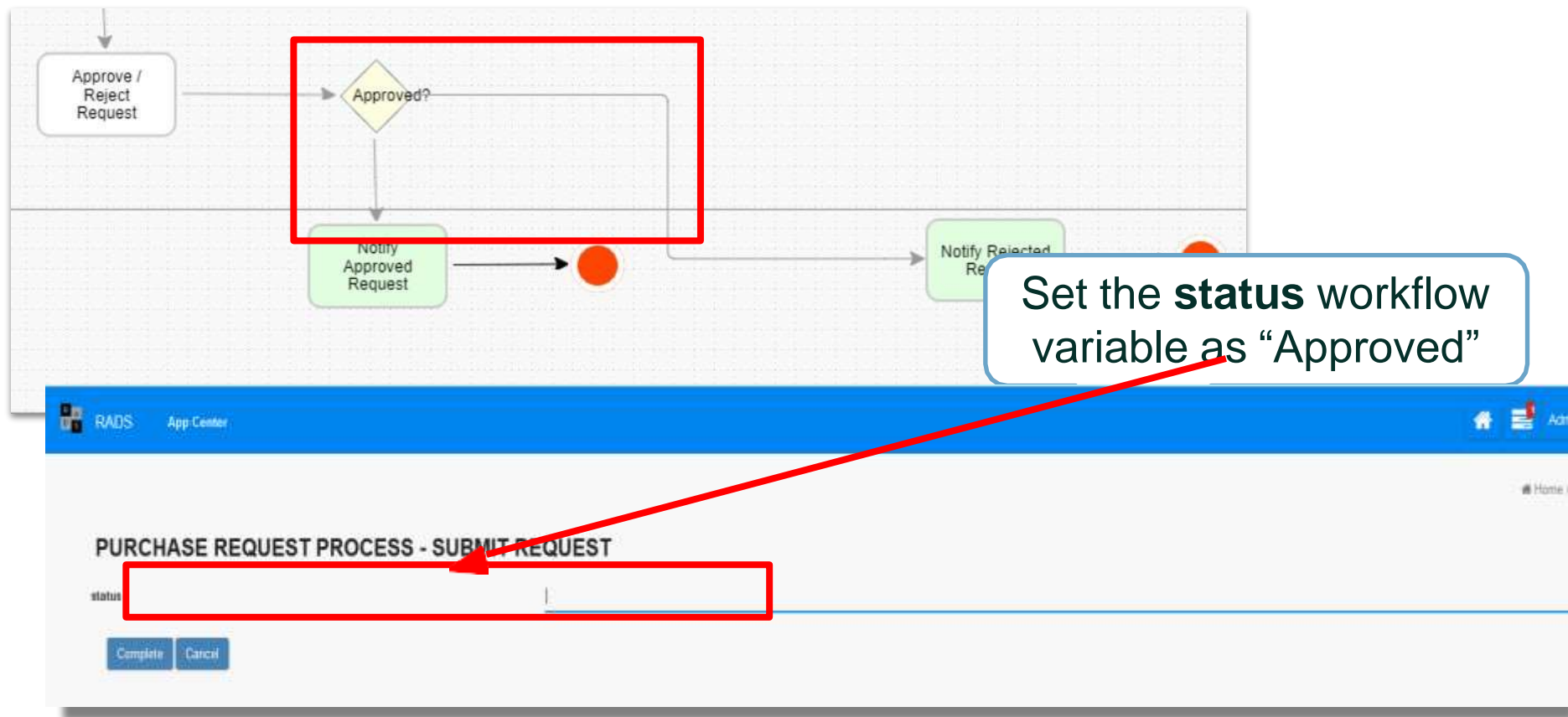
Accessing the Assignments

- App Center -> Universal Inbox



Determining the Flow Using Route

- Manipulate the Process using Workflow Variable and complete the process.



Exercise

- Run through the process from the start again to test out the “Rejected” case scenario.
- The last chapter will inform you how to be *exactly* sure if your process flows properly.

Good to know

- At this point of time, the App is NOT published yet, therefore end user would not be able to run it.
- “Run Process” button at the App Designer’s Processes tab is a convenient way to verify your process flow.

How to Test?

- There are 2 ways in general.
 - Make use of the “**Run Process**” button in App Design > Processes (only for admin)
 - Make use of **Usserview** (for all users)

Verify the Process Using Userview

- Create a new userview.
- Add **Run Process** and **Inbox** element.
- **Publish** the App.
- Try running the whole process again by accessing the userview from the app center.

Create New Userview

The screenshot shows the RAD Service interface. On the left is a sidebar with navigation options: Design App: purchaseRequisition (with Version 1 and Published buttons), Forms & UI (with Processes and Properties & Export), Logs, and All Apps. The main area has a search bar and three buttons: Create New Form, Create New Datalist, and Create New Userview (highlighted with a red box). A red arrow points from the highlighted button to a modal form titled 'CREATE NEW USERVIEW'.

CREATE NEW USERVIEW

USERVIEW DETAILS

ID	purchaseRequisitionPortal
Name	Purchase Requisition Portal
Description	

Userview Category

- Items are organized by Category.
- Add 1 new category – Purchase Requisition.

The screenshot displays the 'USERVIEW BUILDER' interface for 'purchaseRequisition v1: Purchase Requisition Portal (Published)'. The top navigation bar includes 'DESIGN USERVIEW', 'SETTINGS', 'PREVIEW', and 'SAVE'. The left sidebar, under the 'Basic' category, lists various components: Form, HTML Page, Inbox, Link, List, Run Process, Universal Inbox, and User Profile. The main workspace shows the 'Purchase Requisition Portal' title with a 'Click to edit' prompt and a date format '#date.EEE, d MMM yyyy#'. Below this is a 'Menu' section with a 'Home' item and a 'Drop menu item here' placeholder. A red box highlights a '+ Add Category' button in the top right corner of the menu area.

<Cont

Add Run Process Into Userview

- Drag-n-drop **Run Process** into **Purchase Requisition** category.

The screenshot displays the 'USERVIEW BUILDER' interface for 'purchaseRequisition v1: Purchase Requisition Portal (Published)'. The interface includes a top navigation bar with 'DESIGN USERVIEW', 'SETTINGS', 'PREVIEW', and 'SAVE' tabs. A left sidebar lists various widget categories: 'Basic', 'Form', 'HTML Page', 'Inbox', 'Link', 'List', 'Run Process' (highlighted with a red box and an arrow pointing to the main canvas), 'Universal Inbox', and 'User Profile'. The main canvas shows a 'Purchase Requisition Portal' layout with a 'Menu' section containing 'Home' and 'Welcome' items. Below the menu, a 'Purchase Requisition' section is highlighted with a red box, showing a 'Run Process' widget being added. The footer indicates the page is 'Powered by RADS'.

Integrate Run Process Into Userview

- Set the appropriate label.
- Select Purchase Request Process in Process.

The screenshot displays the 'USERVIEW BUILDER' interface for a 'Purchase Requisition Portal'. The top navigation bar includes 'DESIGN USERVIEW', 'SETTINGS', 'PREVIEW', and 'SAVE'. The left sidebar lists various components: Basic, Form, HTML Page, Inbox, Link, List, Run Process, Universal Inbox, and User Profile. The main content area is titled 'Purchase Requisition Portal' and includes a 'Menu' section with 'Home', 'Welcome', 'Purchase Requisition', and 'Run Process'. The 'Edit Run Process' configuration panel is open, showing the following details:

- Edit Run Process** (with a help icon)
- Edit Run Process > Redirection > Advanced > Performance & Offline**
- Id**: 220E6D0376464F118E8282E8B96274EF
- Custom ID**: (empty field)
- Label ***: Submit New Request (highlighted with a red box)
- Process ***: Purchase Request Process (purchaseRequest) (highlighted with a red box)
- Run Process Without Confirmation Screen?**: ☒
- Show in Popup Dialog?**: ☐
- Submit Button Label**: (empty field)

The 'Run Process' component is selected in the menu, and the 'Label' and 'Process' fields are highlighted with a red box, indicating the steps to set the appropriate label and select the Purchase Request Process in Process.

Add Inbox Into Userview

- Add **Inbox** into **Purchase Requisition** category.
- Configure it to show **All Assignments**.

Publish the App

- With the Userview created, it is now ready for prime time.
- End users can now access them through this interface when the App is made published.

Note: Only **ONE** App version can be made published at any point of time.

Publish the App

- Click on **Version/Publish** button, select the version to publish.

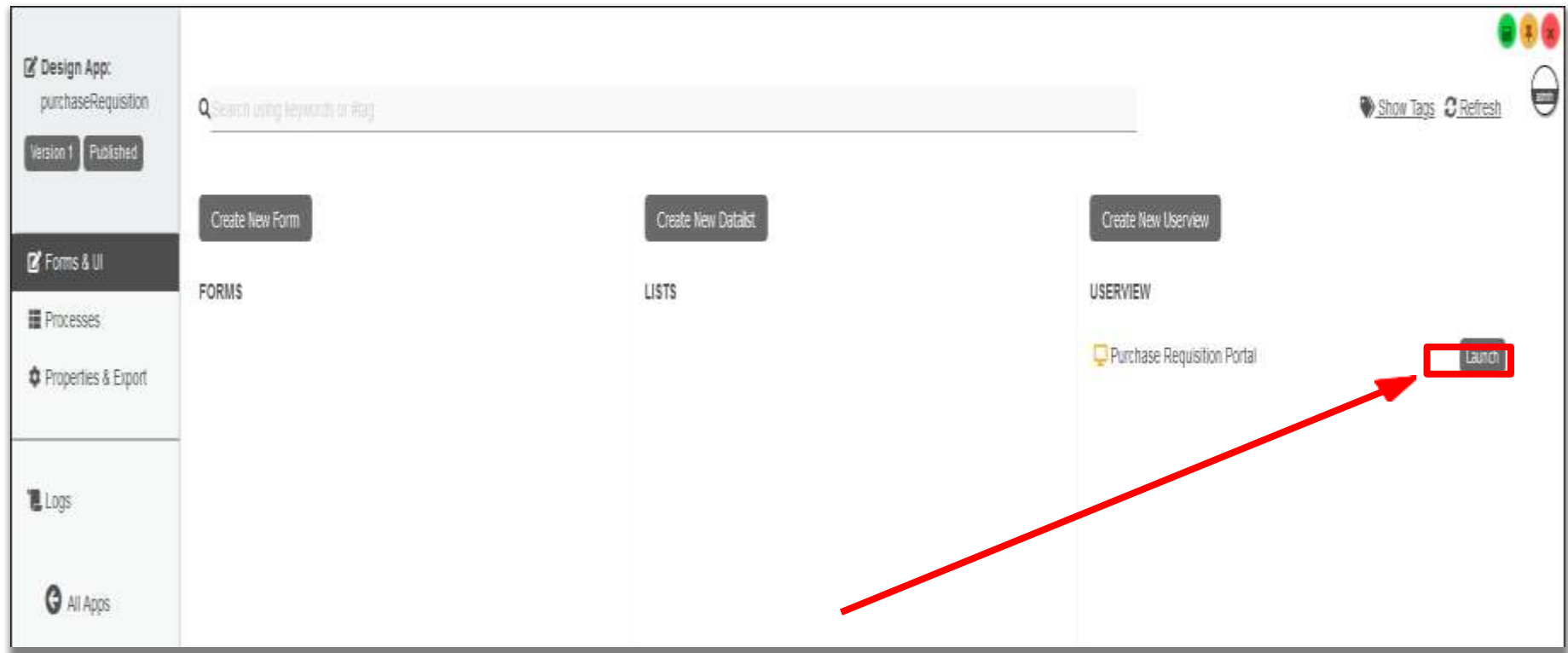
The screenshot shows the RAD Service interface. On the left sidebar, under 'Design App: purchaseRequisition', the 'Version | Published' button is highlighted with a red box. A red arrow points from this button to the 'APP CONFIGURATION MANAGEMENT' section on the right. In this section, the 'Manage App Version' tab is active, displaying a table with the following data:

VERSION	PUBLISHED	NOTES	DATE CREATED
1	✓		18-10-2022 05:25 AM

At the bottom of the interface, the 'Publish' button is also highlighted with a red box. The pagination bar at the bottom indicates 'Page 1 of 1' and 'Displaying 1 to 1 of 1 items'.

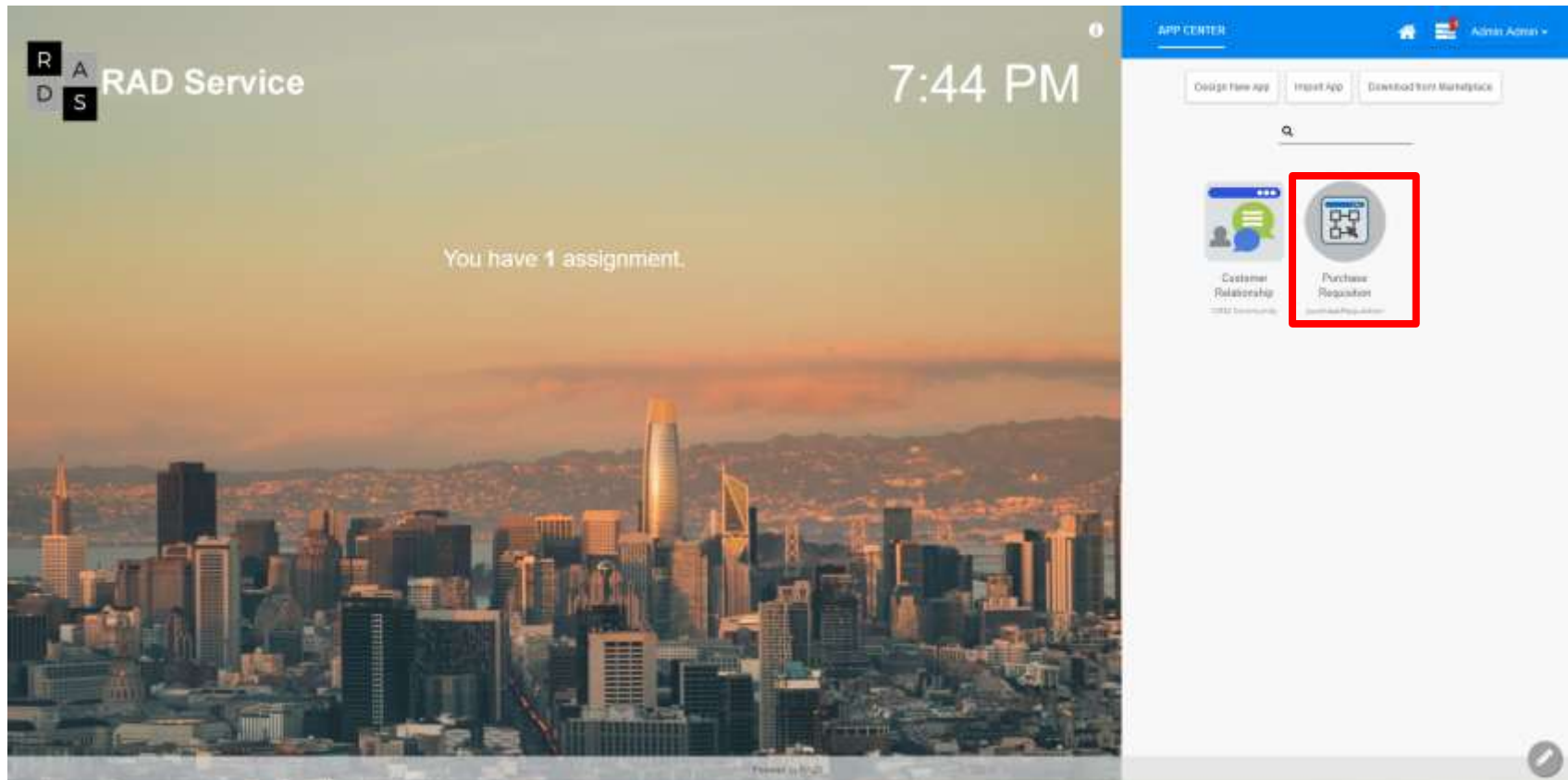
Launching the Published App

- Click on **Launch** under Userview pane.



Launching the Published App

- One can also access it through the **App Center** itself.



Run Through the Process

- With the Userview created, run through the entire **Purchase Request Process** again by using the userview.

The screenshot displays the 'PURCHASE REQUISITION PORTAL' interface. On the left, a sidebar contains the user name 'Admin Admin', a 'Welcome' message, and a 'Submit New Request' button which is highlighted with a red rectangular border. The main content area shows a breadcrumb trail 'Home > Submit New Request' and a title 'PURCHASE REQUEST PROCESS - SUBMIT REQUEST'. Below the title, there is a 'status' label and a text input field. At the bottom of the main area, there is an orange 'Complete' button.

Run through the Process

- Access subsequent assignment through the Inbox menu.

The screenshot displays the 'PURCHASE REQUISITION PORTAL' interface. On the left sidebar, the 'Inbox' menu item is highlighted with a red rectangle. The main content area shows a table of requests with the following data:

ACTIVITY NAME	PROCESS NAME	DATE CREATED	SERVICE LEVEL MONITOR	DUE DATE
Approve / Reject Request	Purchase Request Process	23-10-2022 12:52 PM	-	-
Submit Request	Purchase Request Process	23-10-2022 12:46 PM	-	-
Submit Request	Purchase Request Process	18-10-2022 09:06 AM	-	-

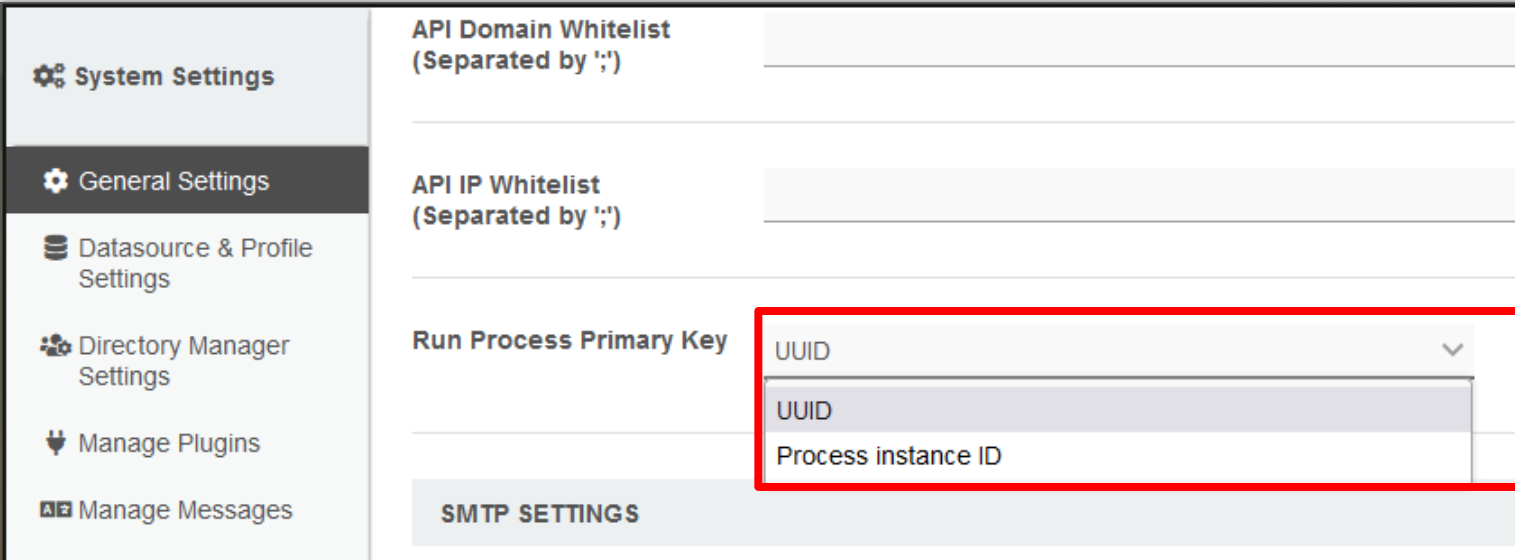
At the bottom right of the table, it states '3 items found, displaying all items.' and provides links for 'CSV | Excel | XML | PDF'. The footer of the page indicates 'Powered by RADS'.

Good To Know

- At this point of time, the App is published, therefore end user would be able to access it.

Important Note!

- In DX, by **default**, Run Process will generate **UUID** as the Primary Key instead of **Process Instance ID**.
- You can **change** it back at Admin Bar > Settings > General Settings > System Administration Settings > Run Process primary key.



The screenshot shows the 'System Settings' page in the RAD Service Admin Bar. The left sidebar contains the following menu items: System Settings, General Settings (highlighted), Datasource & Profile Settings, Directory Manager Settings, Manage Plugins, and Manage Messages. The main content area is divided into sections: API Domain Whitelist (Separated by ';'), API IP Whitelist (Separated by ';'), Run Process Primary Key, and SMTP SETTINGS. The 'Run Process Primary Key' section features a dropdown menu that is currently open, displaying 'UUID' as the selected option and 'Process instance ID' as an available option. The dropdown menu is highlighted with a red rectangle.

System Settings	API Domain Whitelist (Separated by ';')
General Settings	API IP Whitelist (Separated by ';')
Datasource & Profile Settings	Run Process Primary Key
Directory Manager Settings	SMTP SETTINGS
Manage Plugins	
Manage Messages	

Discussion

- Who is Applicant? Who is Approver?
Have we actually defined who they really are?

Chapter 3 Review

- Evaluate and verify the process by using:-
 - Run Process button available to only the App Designer/Admin.
 - Userview which is accessible by end users.

A short horizontal bar with a black left half and an orange right half.

Chapter 4

Participant Mapping

Participant Mapping

- User Mapping Categories.
 - Map to User or Group
 - Map to Org Chart
 - Map to Workflow Variable
 - Map to Plugin

Where to map participants?

ALL participants must be mapped, **except** "System"

The screenshot displays the RAD Service interface for configuring a process. The top navigation bar includes tabs for 'Design Process', 'Configure Mapping', 'Update via Saved XPDL', and 'Run Process'. The 'Process name' is set to 'Purchase Request Process'. Below this, a workflow diagram is visible, showing a sequence of activities: 'Submit Request', 'Review Request', 'Approve Request', and 'Reject Request'. The 'Map Participants to Users' tab is selected, showing a list of participants defined in the workflow design. The list includes 'Applicant', 'Approver', 'System', and 'Process Start White List'. Each participant has an 'Add/Edit Mapping' button next to it. The 'System' participant is highlighted in red, indicating it is the one that should not be mapped.

Design App: purchaseRequisition
Version 1 Published

Forms & UI
Processes
Properties & Export
Logs
All Apps

Design Process Configure Mapping Update via Saved XPDL Run Process

Process name: Purchase Request Process

Show Additional Info

Map Participants to Users Map Activities to Forms Map Tools to Plugins Map Routes to Plugins Variable List

This is the list of participants defined in the Workflow design:

Q

Applicant ID: applicant	Add/Edit Mapping
Approver ID: approver	Add/Edit Mapping
System ID: system	Add/Edit Mapping
Process Start White List ID: processStartWhiteList	Add/Edit Mapping

Participant Mapping

Let's assume that, applicant's HOD is responsible to approve any purchase requests made by the applicant.

The screenshot shows the 'Map Participants to Users' tab in a workflow design tool. It lists three participants defined in the workflow design:

- Applicant** (ID : applicant) with an 'Add/Edit Mapping' button. A red box labeled 'All users' is connected to this participant by a red line.
- Approver** (ID : approver) with an 'Add/Edit Mapping' button. A red box labeled 'Applicant's HOD' is connected to this participant by a red line.
- System** (ID : system) with an 'Add/Edit Mapping' button.

A search bar is located at the top left of the participant list.

Approver: Performer's HOD

Design App: purchaseRequisition
Version 1 Published

Design Process | Configure Mapping | Update via Saved XPDL | Run Process

MAP PARTICIPANTS TO USERS - APPLICANT (APPLICANT)

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

Map to the Performer | Map to a Head of Department (HOD) | Map to an entire Department

☐ Performer
☒ **Performer's HOD**
☐ Performer's HOD (Ignore Report To)
☐ Performer's Subordinates
☐ Performer's Department

Where the performer executed: **Previous Activity**

Approver is Performer's HOD where the performer executed Previous Activity

Process Participant Mapping

Map Participants to Users Map Activities to Forms Map Tools to Plugins Map Routes to Plugins Variable List

This is the list of participants defined in the Workflow design.

Q

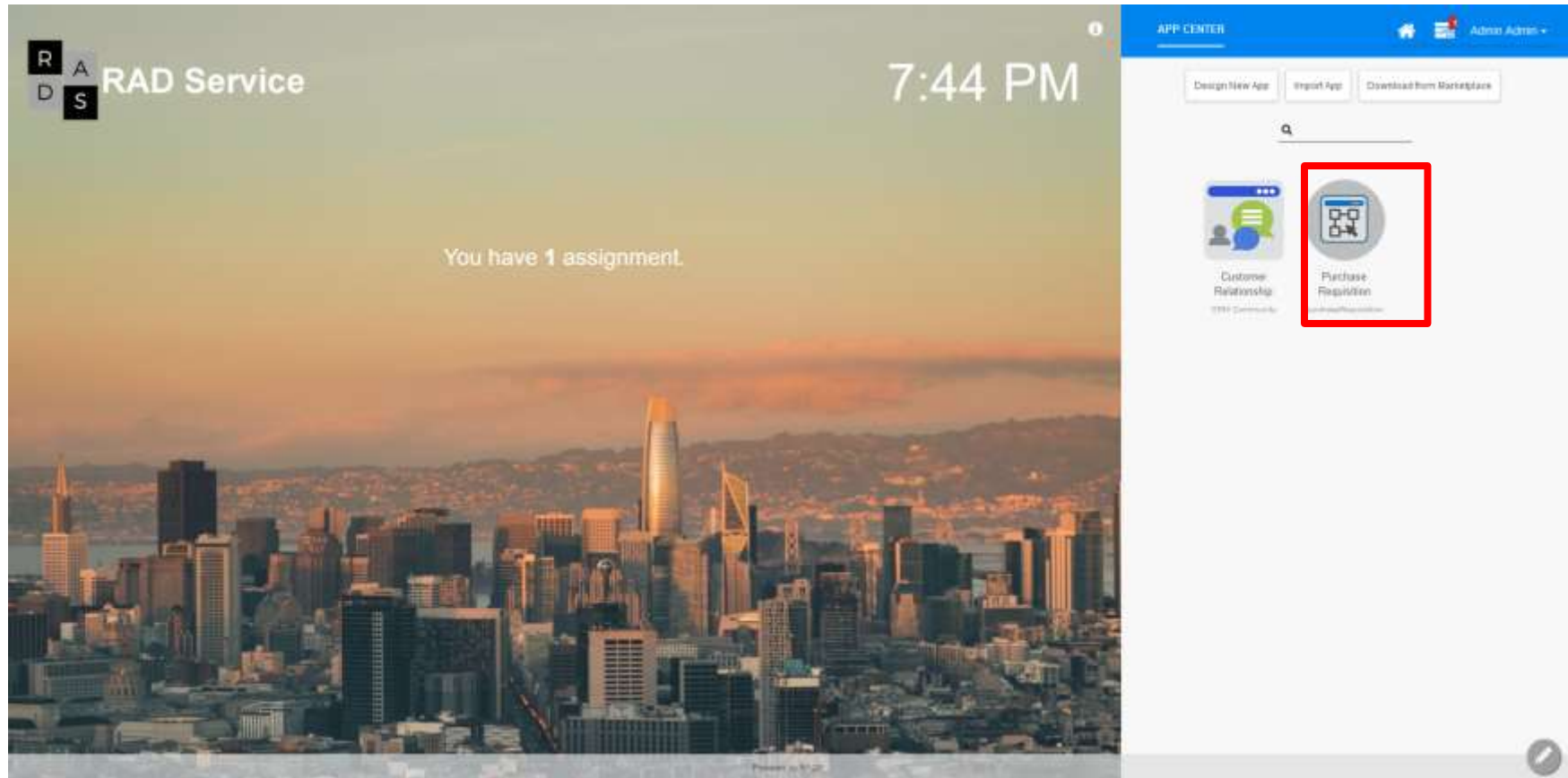
Applicant ID : applicant		Add/Edit Mapping
Type	Performer	
Value	Activity Definition ID : runProcess	
	Remove Mapping	
Approver ID : approver		Add/Edit Mapping
Type	Performer's HOD	
Value	Previous Activity	
	Remove Mapping	

Test Run the Process

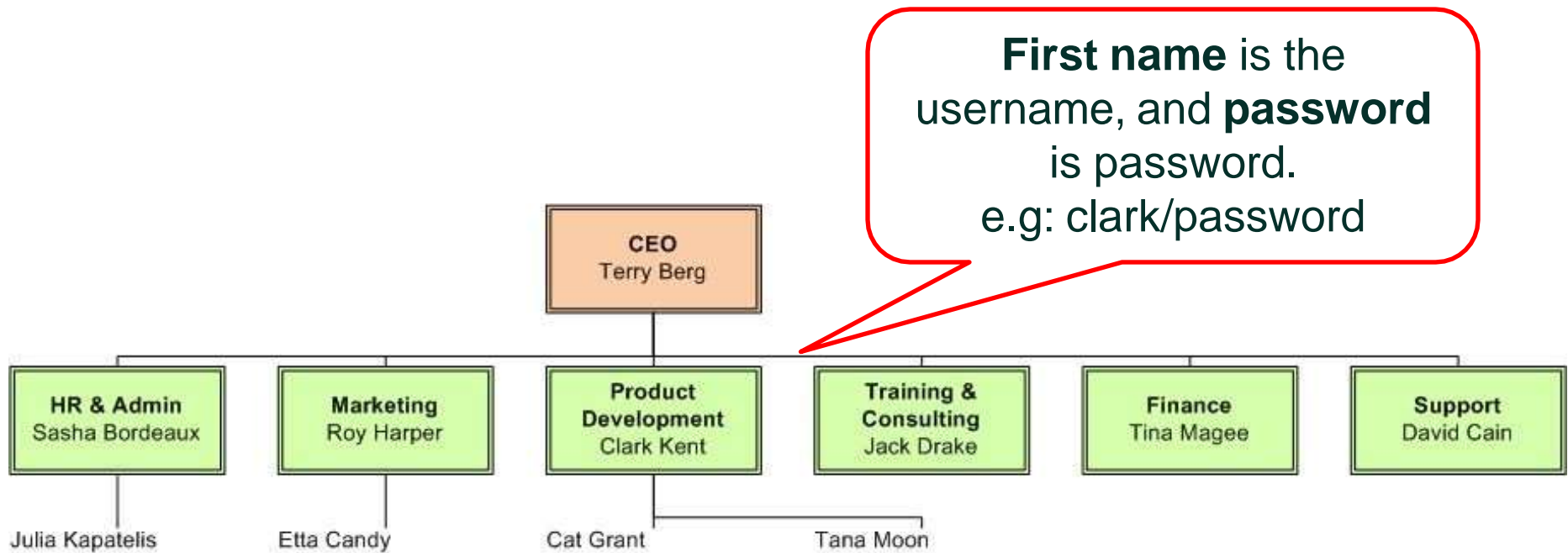
- With the participant mapping set, run the process again as the user “cat”, through the **Purchase Requisition Portal**.

Login As cat/ password

- Open up App Center to access Purchase Requisition portal.



Who Is Cat Grant's HOD?



Note: This is the sample organization chart provided upon initial installation, for your testing convenience. The 3 users: *cat*, *clark*, and *admin*, is free by default.

Important Notes

- All participants MUST be mapped.
- Do NOT choose activity from the same swimlane as the participant that is being mapped unless it is intended so.
- Choose only activity that has been completed.
- If NO user is found, it will be defaulted to the last performer. (Possibly, the current logged in user who had just completed the prior activity OR the process requester)
- If mapped to an inactive user, do delegate, re-assign, or change mapping and re-evaluate the assignment.

Chapter 4 Review

We have learnt to:

1. Understand various Participant Mapping options available.
2. Understand the important rules in Participant Mapping.
3. Able to relate and manage participant mapping with the administrative process monitoring module.

A short horizontal bar with a black left half and an orange right half.

Chapter 5

Process Monitoring

Monitoring Processes

- Allows you to monitor the status of processes, both **running** and **completed**.
- Allows you to view an **audit trail** of process actions.
- Captures process data to allow for the generation of reports.

Monitor Running Processes

- Click on Monitor from the Admin Bar to access.

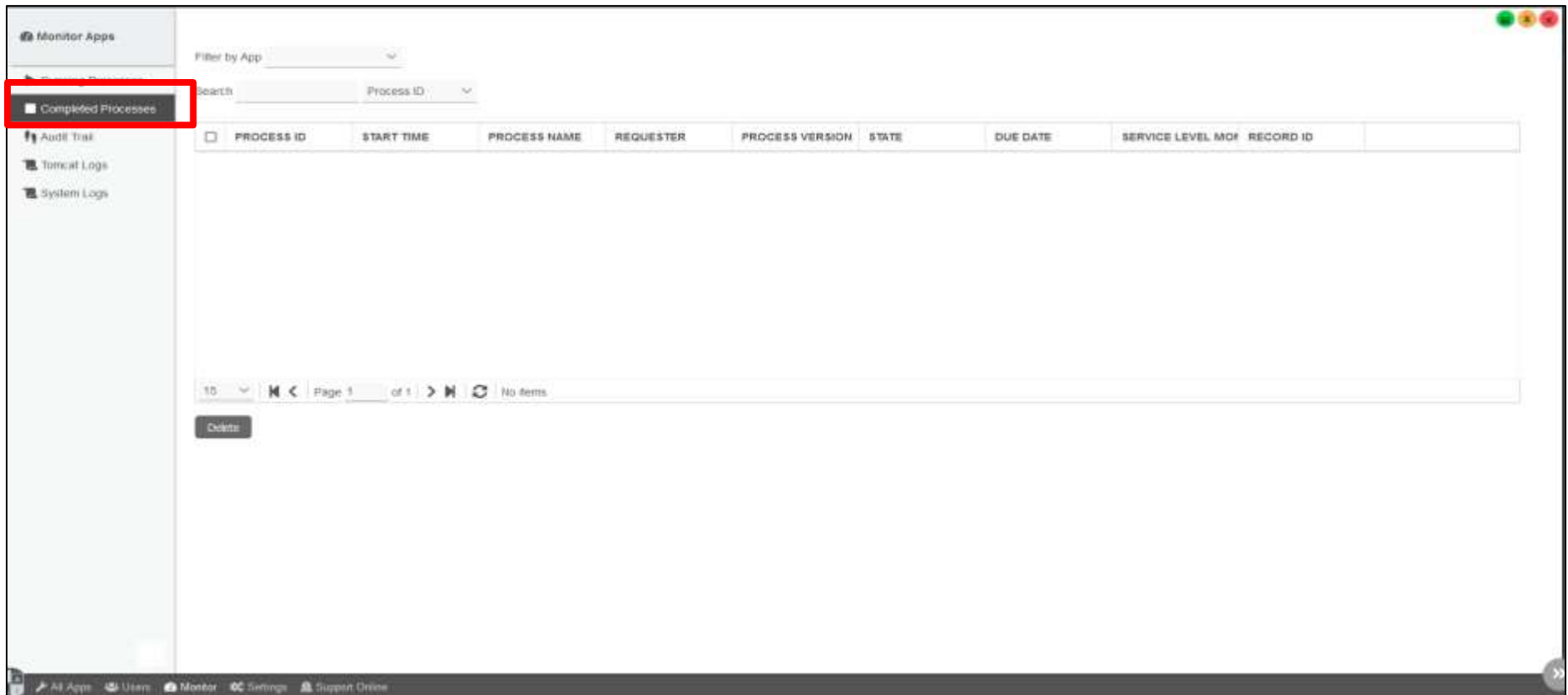
The screenshot displays the 'Monitor Apps' interface. On the left, a sidebar contains navigation options: 'Monitor Apps', 'Running Processes' (selected), 'Completed Processes', 'Audit Trail', 'Tomcat Logs', and 'System Logs'. The main area shows a table of running processes with columns: PROCESS ID, START TIME, PROCESS NAME, REQUESTER, PROCESS VERSION, STATE, DUE DATE, SERVICE LEVEL MOF, and RECORD ID. Three process entries are visible, all with a state of 'open.running'. A red callout box points to the 'PROCESS ID' column, containing the text: 'Process Instance ID Format: #_AppId_ProcessId'. At the bottom, the 'Monitor' button in the Admin Bar is highlighted with a red box.

PROCESS ID	START TIME	PROCESS NAME	REQUESTER	PROCESS VERSION	STATE	DUE DATE	SERVICE LEVEL MOF	RECORD ID
1002_purchaseReques	23-10-2022 12:47 PM	Purchase Request Process	admin	4	open.running	-	-	bcdd727-fa90-44b0-b908-8d053360a3c8
1001_purchaseReques	23-10-2022 12:46 PM	Purchase Request Process	admin	4	open.running	-	-	cb2b60a3-09ee-4cab-a001-ce75e05e9637
902_purchaseReques	23-10-2022 09:06 AM	Purchase Request Process	admin	4	open.running	-	-	26785d5b-c3ed-4150-8f08-8f3cd

Process Instance ID Format:
#_AppId_ProcessId

Monitor Completed Processes

- Click on “Completed Processes” to see completed process instances.



View the Details of a Running Process Instance

- From the list of running processes, **click into any instance.**
- **Details on the current state** of the selected process instance will be shown.

View the Details of a Running Process Instance

The screenshot displays the 'Monitor Apps' interface. On the left, a sidebar contains navigation links: 'Monitor Apps', 'Running Processes' (selected), 'Completed Processes', 'Audit Trail', 'Tomcat Logs', and 'System Logs'. The main area shows details for a running process instance of 'purchaseRequisition.x1'. Above the details are buttons for 'View Graph', 'Abort Instance', 'Remove Instance', and 'Re-evaluate'. The process details include: App (purchaseRequisition.x1), Process name (Purchase Request Process), Process ID (1002_purchaseRequisition_purchaseRequestProcess), Record ID (bcedd727-fa90-44b0-bf98-8f953380a3c8), Process Definition ID (purchaseRequisition#4#purchaseRequestProcess), Process Version (4), State (open.running), Service Level Monitor (-), Requester (admin), and Start Time (23-10-2022 12:47 PM). Below this, a table titled 'ACTIVITY LOG' shows the sequence of activities. A red callout box points to the 'ACTIVITY ID' column, explaining its format. Another red callout box points to the 'STATE' column, asking which activities are completed and which is active now.

Activity Instance ID format:
#_ProcessInstanceId_ActivityId

Which activities are completed?
Which activity is active now?

ACTIVITY ID	ACTIVITY NAME	STATE	CREATE TIME	SERVICE LEVEL MON
1003_1002_purchase	Approve / Reject Request	open.not_running.not	23-10-2022 12:52 PM	-
1002_1002_purchase	Submit Request	closed.completed	23-10-2022 12:47 PM	-

View the Details of a Running Process Instance

- These are actions that can be taken on a **process instance**.
 - **View Graph**
 - Displays the process diagram, where the current activities are highlighted in yellow.
 - **Abort Instance**
 - Terminates the process instance, leaving the process data intact
 - **Remove Instance**
 - Permanently delete the process instance, process data will be removed as well
 - **Re-evaluate**
 - In cases where participant mappings are changed before the process is completed, this action will allow pending assignments to be reassigned to the newly mapped participants.



View the Details of an Activity

Monitor Apps

Running Processes

Completed Processes

Audit Trail

Tomcat Logs

System Logs

Re-evaluate

Re-evaluate Assignment(s) For User

Reassign User

Complete

Process Instance

1002_purchaseRequisition_purchaseRequestProcess

Activity ID

1003_1002_purchaseRequisition_purchaseRequestProcess_approveRejectRequest

Activity Definition ID

approveRejectRequest

Activity Name

Approve / Reject Request

State

open.not_running.not_started

Service Level Monitor

-

List of Pending

admin

Create time

Date limit

Due date

Delay

Finish time

System Date Created

Workflow variable value

ACTIVITY LIST

ACTIVITY ID	ACTIVITY NAME	STATE	CREATE TIME	SERVICE LEVEL MOI
913_172_purchaseRequisition_purchaseRequestProcess_approveRejectRequest	Approve / Reject Request	open.not_running.not_	10-12-2019 01:23 AM	-
912_172_purchaseRequisition_purchaseRequestProcess_submitRequest	Submit Request	closed.completed	10-12-2019 01:23 AM	-

VARIABLE LIST

status

Approved

Set

View the Details of a Running Activity Instance

- These are the actions that can be taken on an **activity instance**.
 - **Re-evaluate**
 - Re-evaluate the participant mapping for the current activity instance.
 - **Re-evaluate assignment(s) for user**
 - Re-evaluate all assignments of a particular user.
 - **Reassign user**
 - Reassign one of the assignee(s) of this activity instance to another user.
 - **Complete**
 - Complete the activity instance as the current logged in user.

Monitor Apps

Running Processes

Completed Processes

Re-evaluate Re-evaluate Assignment(s) For User Reassign User Complete

Process Instance	1002_purchaseRequisition_purchaseRequestProcess
Activity ID	1003_1002_purchaseRequisition_purchaseRequestProcess_approveRejectRequest
Activity Definition ID	approveRejectRequest

Discussion

- What if a person who is assigned to an assignment has left the organization? What can we do?

Chapter 4 Review

We have learned to:

1. Monitor running and completed processes.
2. Drill down into activity details of each process that is running or completed.
3. Debug processes by examining workflow variable value.
4. Appreciate the “Re-evaluate” feature in process monitoring.

Module Review

1. Business Process Design (BPE)
2. Designing your first Process with Process Builder
3. Running your first Automated Process
4. Participant Mapping
5. Process Monitoring

Recommended Further Learning

- Designing Forms for the activities created in your Process Flow.
- Configure Process Tool plugin for the tool created in your Process Flow.

Stay Connected with RADS

- rads.purwana.net
- <https://github.com/radservice/rad-community>