



**Joget DX**

# Designing your first Form

 <http://facebook.com/jogetworkflow>

 <http://twitter.com/jogetworkflow>

# Prerequisites

---

- Appreciates the use of a form
- Understanding on ER diagram is highly recommended
- Created a Process

# Content

---

1. Introduction to Form Builder
2. Designing your First Forms
3. Controlling Process by using Form



# Chapter 1

## Introduction to Form Builder



### Apps

Progressive Web Apps  
Various Form Factors



Customer Relationship



Employee Services Portal



Expenses Claims App



Internal Service Request App



### Development

Simple, Visual,  
Drag and Drop.

Form Builder	List Builder	Userview Builder	Process Builder
<p>FORM BUILDER Customer</p> <p>DESIGN FORM</p> <p>PROPERTIES</p> <p>SOURCE</p> <p>DESIGN</p> <p>DESIGN USERVIEW</p> <p>SETTINGS</p> <p>PREVIEW</p>	<p>DATALIST BUILDER Customer</p> <p>DESIGN</p> <p>PROPERTIES</p> <p>SOURCE</p> <p>DESIGN USERVIEW</p> <p>SETTINGS</p> <p>PREVIEW</p>	<p>USERVIEW BUILDER Customer Relationship</p> <p>DESIGN</p> <p>PROPERTIES</p> <p>SOURCE</p> <p>DESIGN USERVIEW</p> <p>SETTINGS</p> <p>PREVIEW</p>	<p>PROCESS BUILDER - Customer Relationship Management</p> <p>DESIGN</p> <p>PROPERTIES</p> <p>SOURCE</p> <p>DESIGN USERVIEW</p> <p>SETTINGS</p> <p>PREVIEW</p>

**User Management**

- Organizations
- Groups
- Users

**Monitoring**

- Processes
- Application Performance
- Logging & Audit

**System**

- Workflow Engine
- Security and Directory
- Plugin Management

Plugin Architecture

**Add-on Builders**

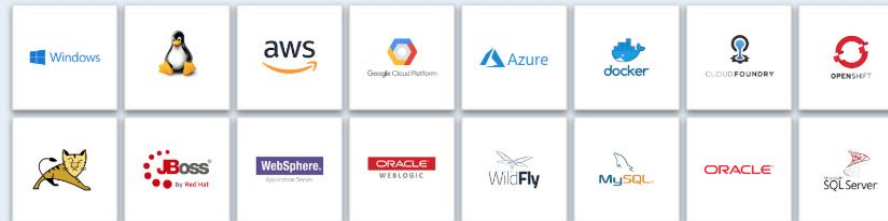
Integration API

- API JSON
- API Java
- API JavaScript



### Infrastructure

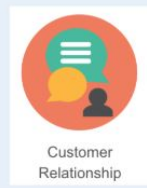
On-Premise, Containers,  
Private / Public / Hybrid  
Cloud. App Server,  
DB, OS Independent.



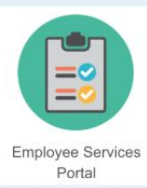
Icons by [icons8.com](https://icons8.com)



**Apps**  
Progressive Web Apps  
Various Form Factors



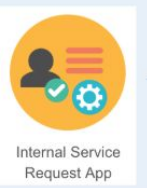
Customer Relationship



Employee Services Portal



Expenses Claims App



Internal Service Request App



**Development**  
Simple, Visual,  
Drag and Drop.

<b>Form Builder</b> <small>FORM BUILDER Customer</small> 	<b>List Builder</b> <small>DATALIST BUILDER Customer</small> 	<b>Userview Builder</b> <small>USERVIEW BUILDER Customer Relationship</small> 	<b>Process Builder</b> <small>PROCESS BUILDER - Customer Relationship Management</small> 
<b>User Management</b> Organizations Groups Users	<b>Monitoring</b> Processes Application Performance Logging & Audit	<b>System</b> Workflow Engine Security and Directory Plugin Management	<b>Add-on Builders</b> 

Plugin Architecture

Integration API

JSON

Java

JavaScript



**Platform**  
Open, Flexible and  
Extensible.



**Infrastructure**  
On-Premise, Containers,  
Private / Public / Hybrid  
Cloud. App Server,  
DB, OS Independent.



Icons by [icons8.com](https://icons8.com)

# Form Builder

---











- Used as a mean to collect information/interact with the end user.
- Drag and drop, and configure based form builder.
- Little/No programming knowledge needed to operate.
- Extensible functionalities through Joget's plugin architecture.

# Form Builder




FORM BUILDER Expenses Claim v2: Expense Claim New (Published)

DESIGN FORM
PROPERTIES
PREVIEW
SAVE

Basic

-  Hidden Field
-  Text Field
-  Password Field
-  Text Area
-  Select Box
-  Check Box
-  Radio
-  Date Picker
-  File Upload
-  Subform

Custom

-  Grid
-  Custom HTML
-  ID Generator Field

Expense Claim Details

Drag This Column

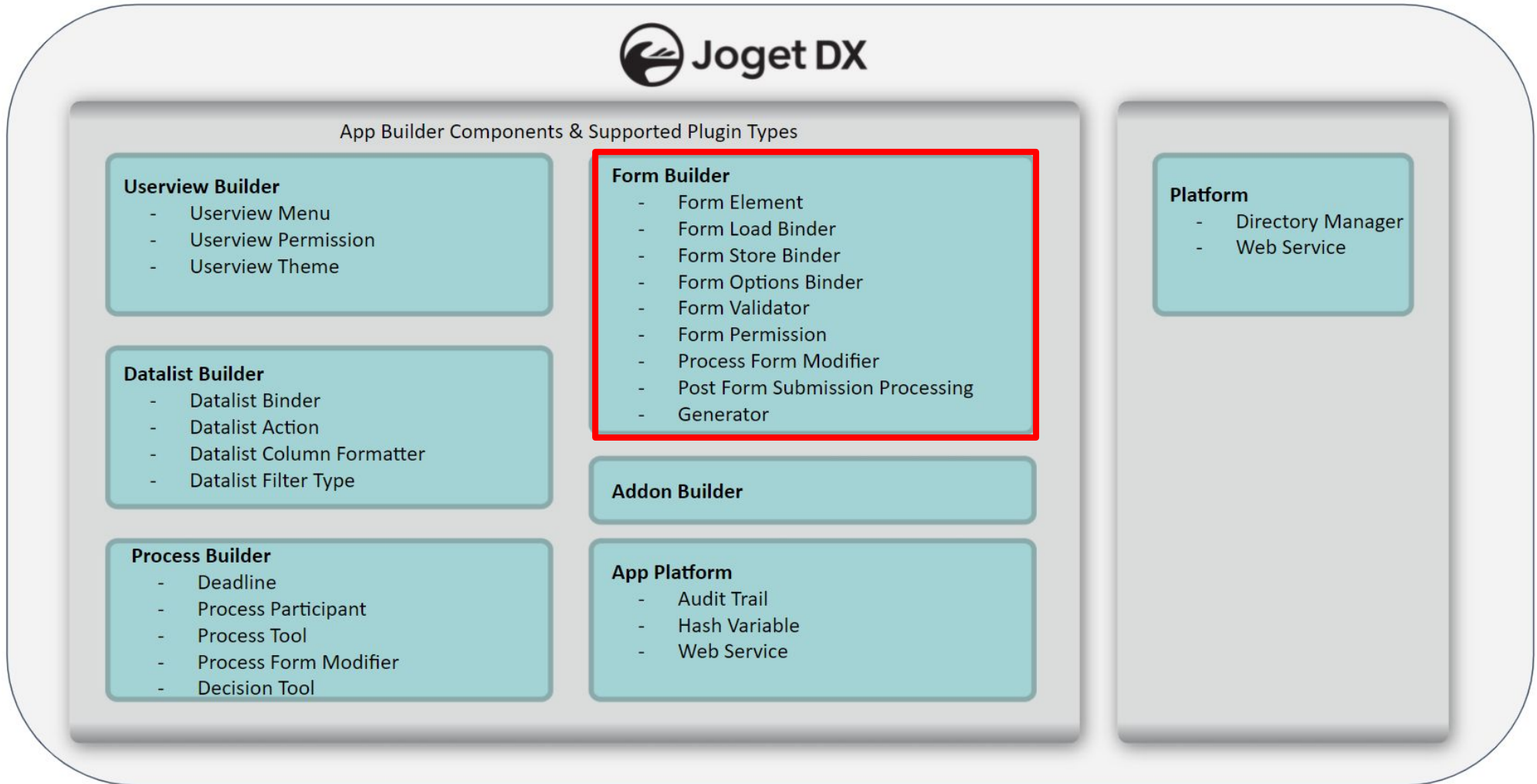
Title	ID Generator Field
(E	
Cl	
Re	
Se	
cla	HIDDEN FIELD
us	HIDDEN FIELD
sta	HIDDEN FIELD

- Form elements.
- Each element is a plugin
- Extensible through Joget's plugin architecture
- Drag-and-drop

© Joget DX - Joget Inc. All Rights Reserved.



# Plugin Types



# Form Builder

**ADVANCED TOOLS**

Tree Viewer    Permission    Usages    Table    I18N    Tooltip    Diff Checker    **JSON Definition**

```
1 {
2   "className": "org.joget.apps.form.model.Form",
3   "properties": {
4     "noPermissionMessage": "",
5     "loadBinder": {
6       "className": "org.joget.apps.form.lib.WorkflowFormBinder",
7       "properties": {}
8     },
9     "name": "Expense Claim New",
10    "description": "",
11    "postProcessorRunOn": "both",
12    "permission": {
13      "className": "",
14      "properties": {}
15    },
16    "id": "ExpensesClaimNew",
17    "postProcessor": {
18      "className": "",
19      "properties": {}
20    },
21    "storeBinder": {
22      "className": "org.joget.apps.form.lib.WorkflowFormBinder",
23      "properties": {}
24    },
25    "tableName": "j_expense_claim"
26  },
27  "elements": [
```

Update

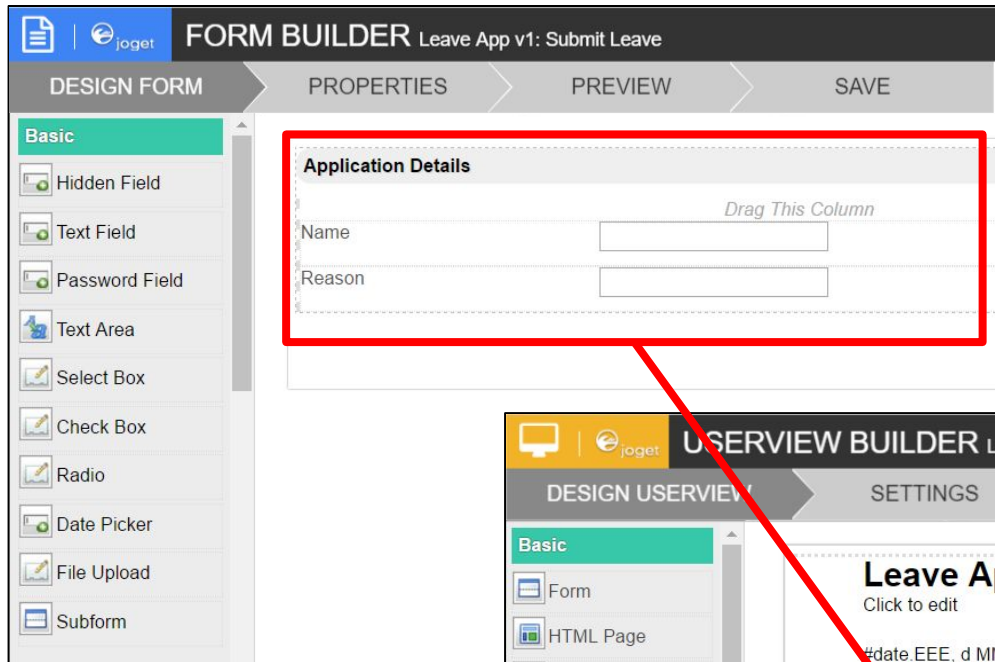
Advanced method for developer to edit the form design by changing its JSON definition

# Where and How Form is used?

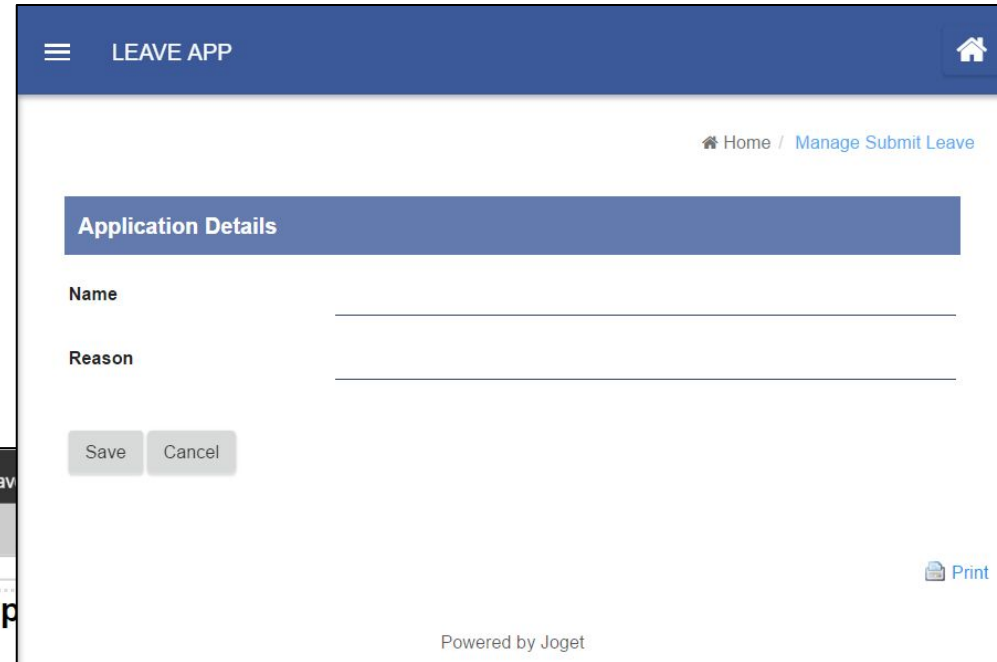
---

- By itself
  - Each form record is distinctive by itself.
  - Made accessible through the use of **Uerview**.

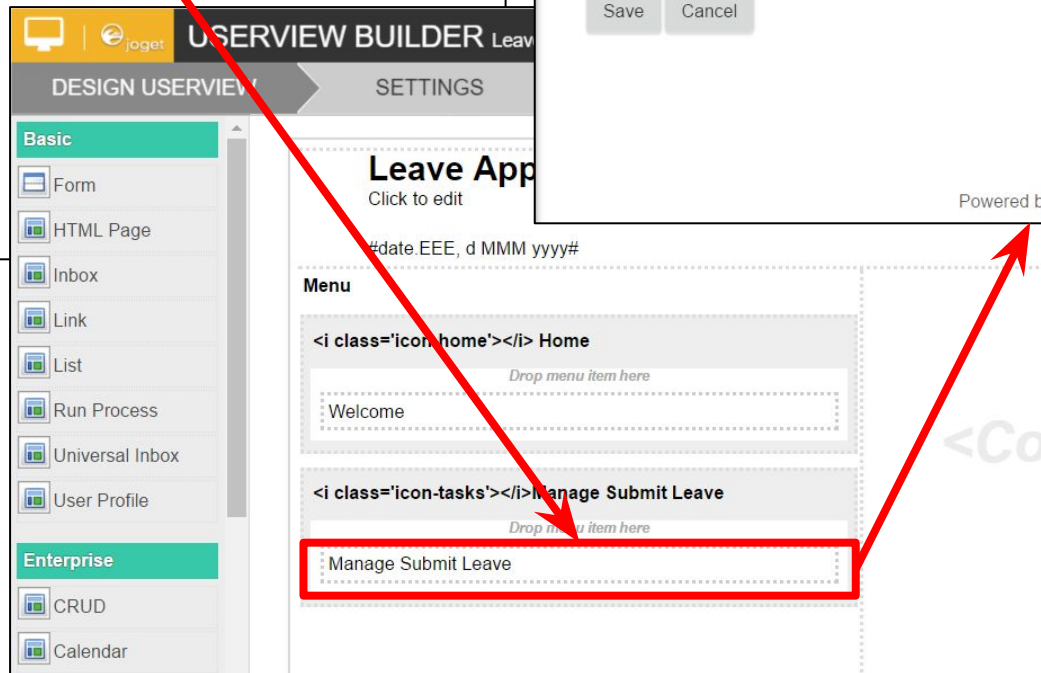
# Form or CRUD in Userview



The screenshot shows the 'FORM BUILDER' interface with tabs for 'DESIGN FORM', 'PROPERTIES', 'PREVIEW', and 'SAVE'. A sidebar on the left lists various form components under the 'Basic' category, including Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, File Upload, and Subform. The main workspace displays a form titled 'Application Details' with two input fields labeled 'Name' and 'Reason'. A red box highlights this form, and a red arrow points from it to the 'Userview Builder' interface.



The screenshot shows the 'LEAVE APP' user view. The header includes a home icon, the text 'LEAVE APP', and a home icon. Below the header, there is a breadcrumb trail: 'Home / Manage Submit Leave'. The main content area displays the 'Application Details' form, which is a blue header followed by two input fields labeled 'Name' and 'Reason', and two buttons labeled 'Save' and 'Cancel'. A 'Print' icon is visible in the bottom right corner. A red arrow points from the 'Form Builder' interface to this user view.



The screenshot shows the 'USERVIEW BUILDER' interface with tabs for 'DESIGN USERVIEW' and 'SETTINGS'. A sidebar on the left lists various user view components under the 'Basic' category, including Form, HTML Page, Inbox, Link, List, Run Process, Universal Inbox, User Profile, and under the 'Enterprise' category, including CRUD and Calendar. The main workspace displays a user view titled 'Leave App' with a 'Click to edit' prompt. Below the title, there is a date placeholder '#date.EEE, d MMM yyyy#' and a 'Menu' section. The menu contains two items: 'Home' and 'Manage Submit Leave'. A red box highlights the 'Manage Submit Leave' menu item, and a red arrow points from it to the 'Form Builder' interface.

# Where and How Form is Used?

---

- Part of a Process
  - Each form record relates to a process instance.
  - Made accessible through the process activity's assignment view in the Inbox.
    - Inbox is made accessible via the **App Center** or **Userview**.

# Form - Assignment - Inbox - Userview

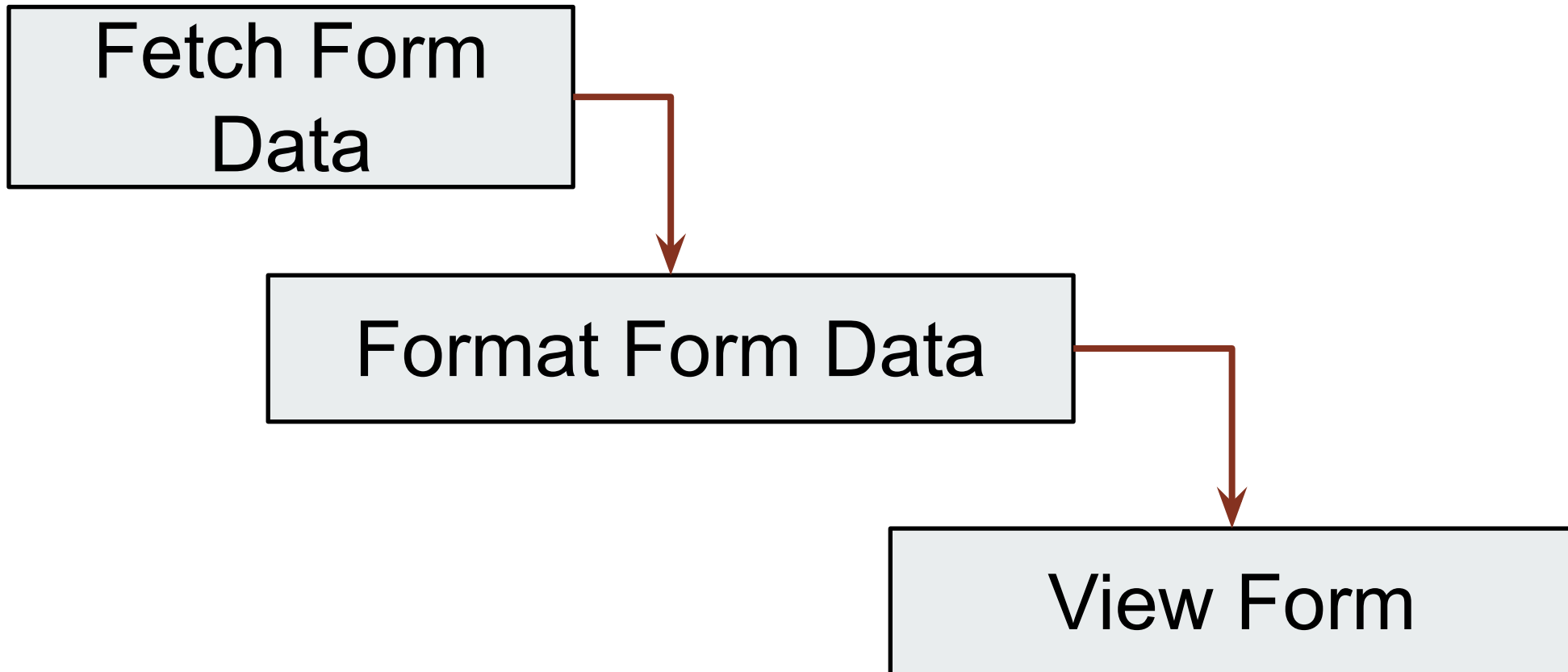
The image illustrates the development and deployment of a leave approval process using Joget. It is divided into three main sections:

- Workflow Diagram (Top Left):** Shows the process flow for an Applicant and an Approver. The Applicant starts with 'Email on Received', leading to 'Approval'. From 'Approval', the process can go to 'Clarification' (if 'status == "Clarification Required"') or 'Update Record Status'. 'Update Record Status' leads to 'Route 1', which then branches based on 'status ==' to either 'Approval' or 'Update Record Status'.
- Form Builder (Top Right):** The 'FORM BUILDER' interface for 'Leave App v1: Submit Leave Approval (Published)'. It shows the 'DESIGN FORM' tab with a visual form layout. The form includes sections for 'Application Details' (Name, Reason), 'Approval' (approval\_status, Status), and 'Remarks'. A toolbar with various form elements like Text Field, Password Field, Select Box, etc., is visible on the left.
- User View (Bottom):** The final user interface titled 'SUBMIT LEAVE APPROVAL PROCESS - APPROVAL'. It features a sidebar with navigation options: 'Welcome', 'Manage Submit Leave', 'Submit Leave Approval Process', 'Inbox', and 'Submit Submit Leave'. The main content area displays the form with 'Application Details' (Name: Cat Grant, Reason: Holiday) and 'Approval' (Status: ). A 'Remarks' section with a rich text editor is at the bottom.

Red arrows indicate the flow: from the 'Approval' step in the workflow to the 'Approval' section in the user view; from the 'Update Record Status' step in the workflow to the 'Remarks' section in the user view; and from the 'Route 1' decision diamond in the workflow to the 'Status' dropdown in the user view.

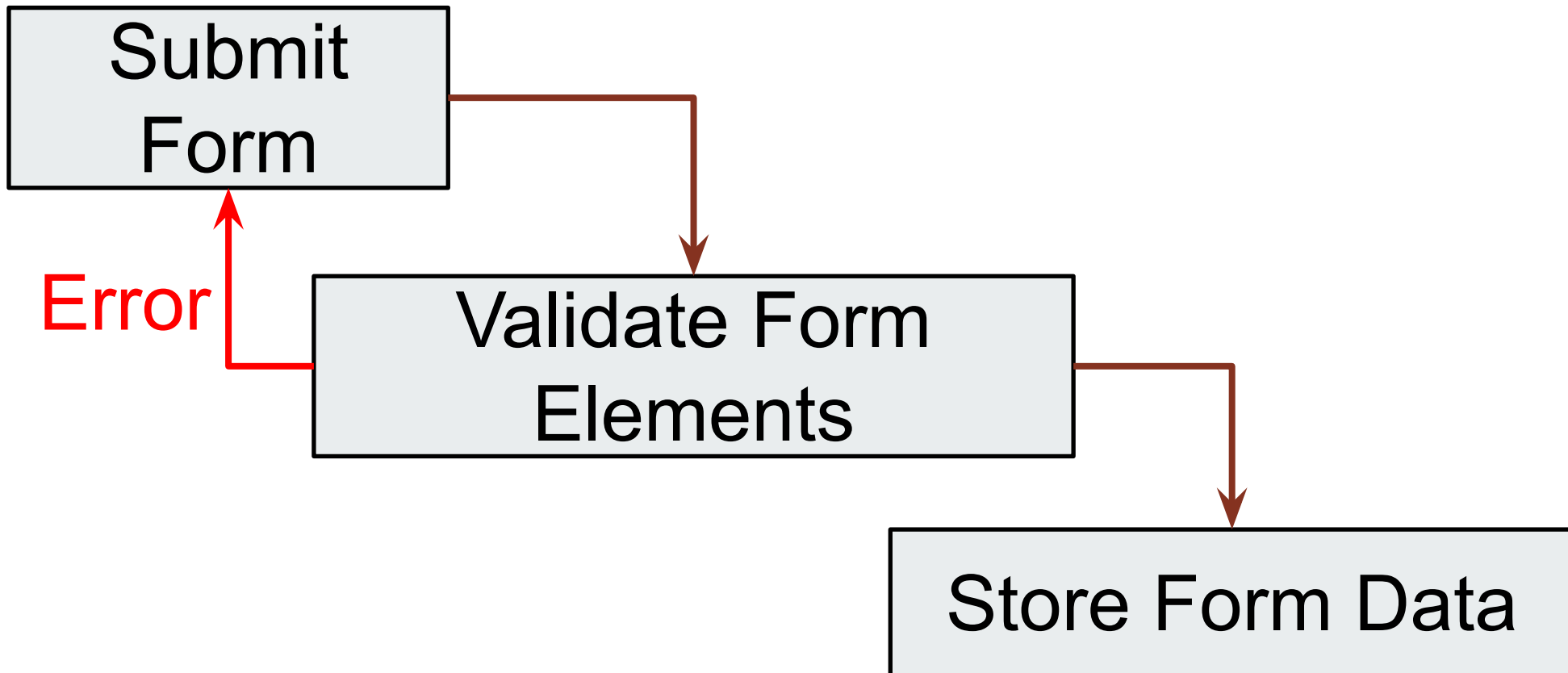
# Viewing a Form

- In general, this is how a Form is loaded.



# Submitting a Form

- In general, this is how a Form is handled when submitted.





# Good To Know

---

- It is not possible to run/view a Form without having Joget running as the server.

# Chapter Review

---

- General understanding on how Form and Form Builder works.

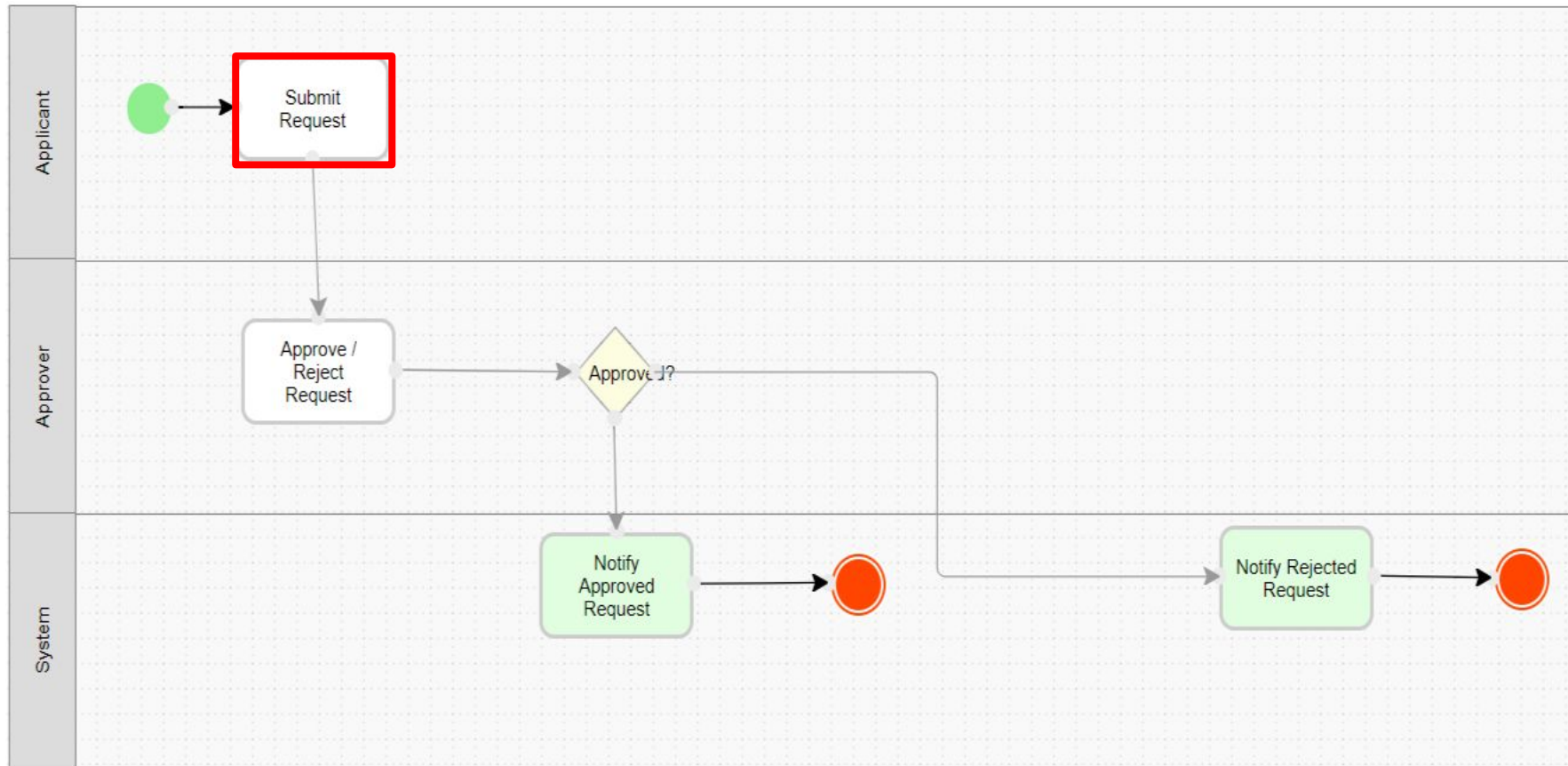


# Chapter 2

## Designing Your First Form

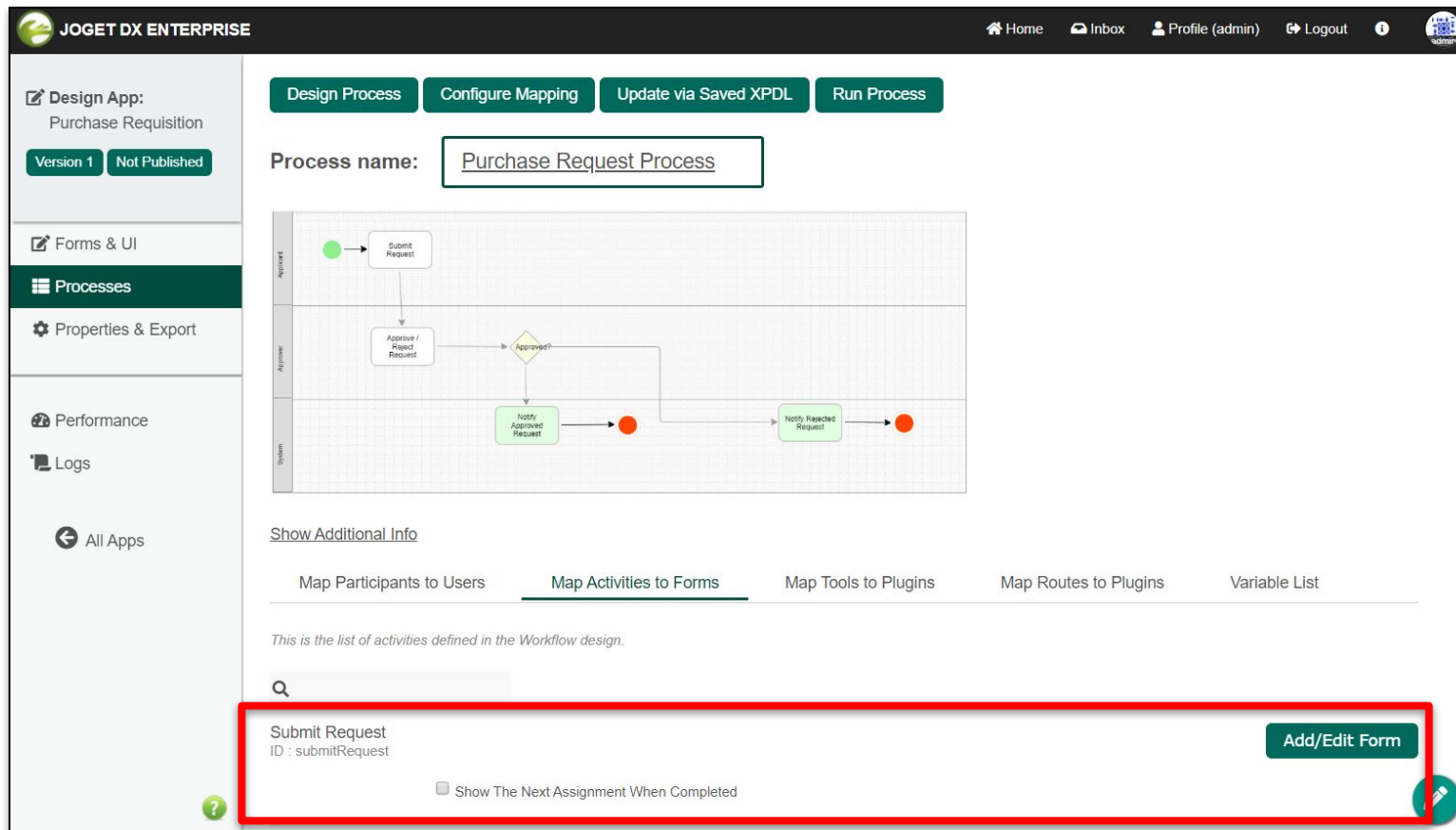
# Association With the Process Flow

- Designing the first form for the first activity in the flow.



# Map Activities to Form

- Go to App's **Processes** -> **Map Activities to Forms**.
- Click on "Add/Edit Form" on "Submit Request" activity.



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 (Version 1, Not Published)', 'Forms & UI', 'Processes' (highlighted), 'Properties & Export', 'Performance', 'Logs', and 'All Apps'. The main content area features buttons for 'Design Process', 'Configure Mapping', 'Update via Saved XPDL', and 'Run Process'. Below these is a 'Process name' field containing 'Purchase Request Process'. A workflow diagram is shown, with activities: 'Submit Request', 'Approve / Reject Request', 'Approved?' (decision), 'Notify Approved Request', and 'Notify Rejected Request'. At the bottom, there are tabs for 'Map Participants to Users', 'Map Activities to Forms' (selected), 'Map Tools to Plugins', 'Map Routes to Plugins', and 'Variable List'. A search bar is present above a list of activities. The 'Submit Request' activity is highlighted with a red box, and its 'Add/Edit Form' button is also highlighted.

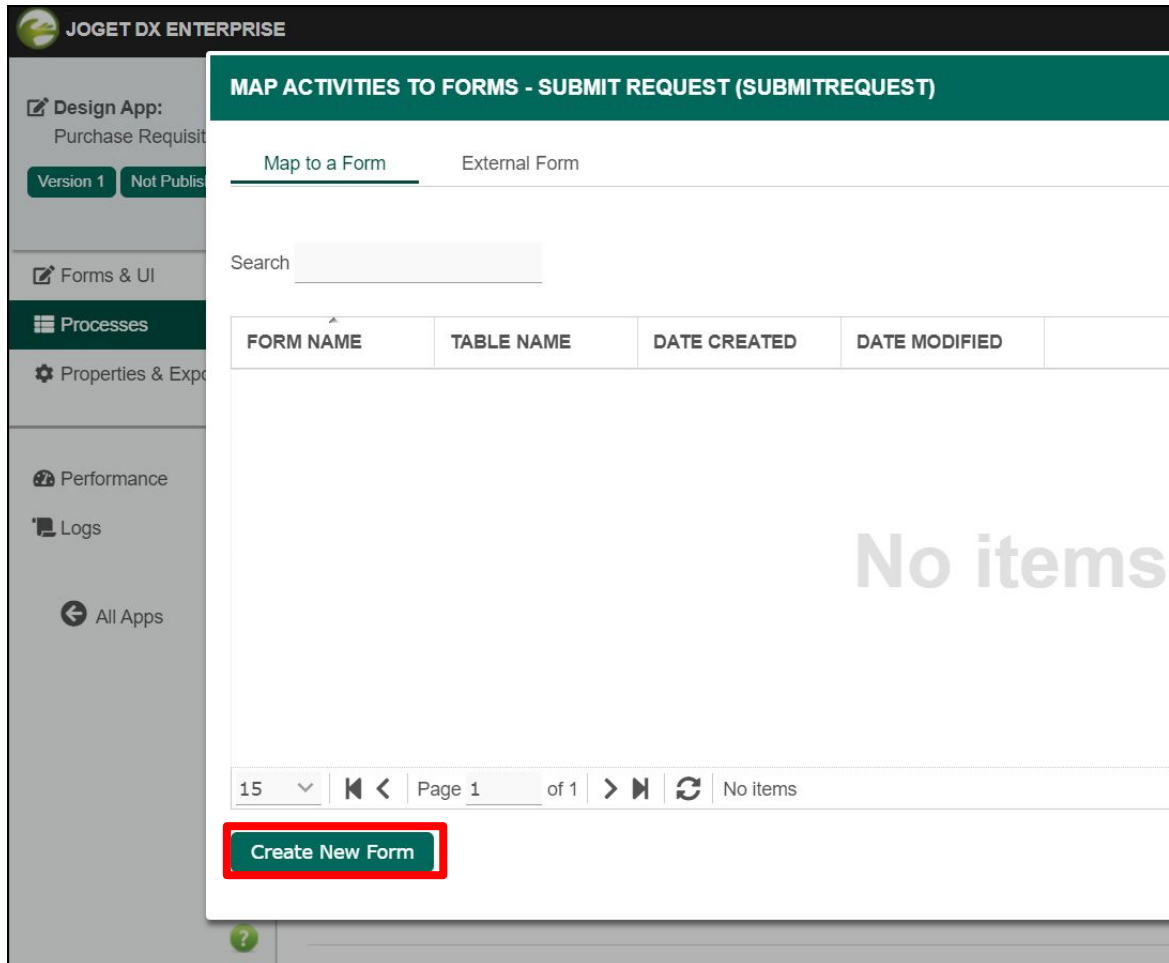
**Submit Request**  
ID : submitRequest

Show The Next Assignment When Completed

**Add/Edit Form**

# Add/Edit Form

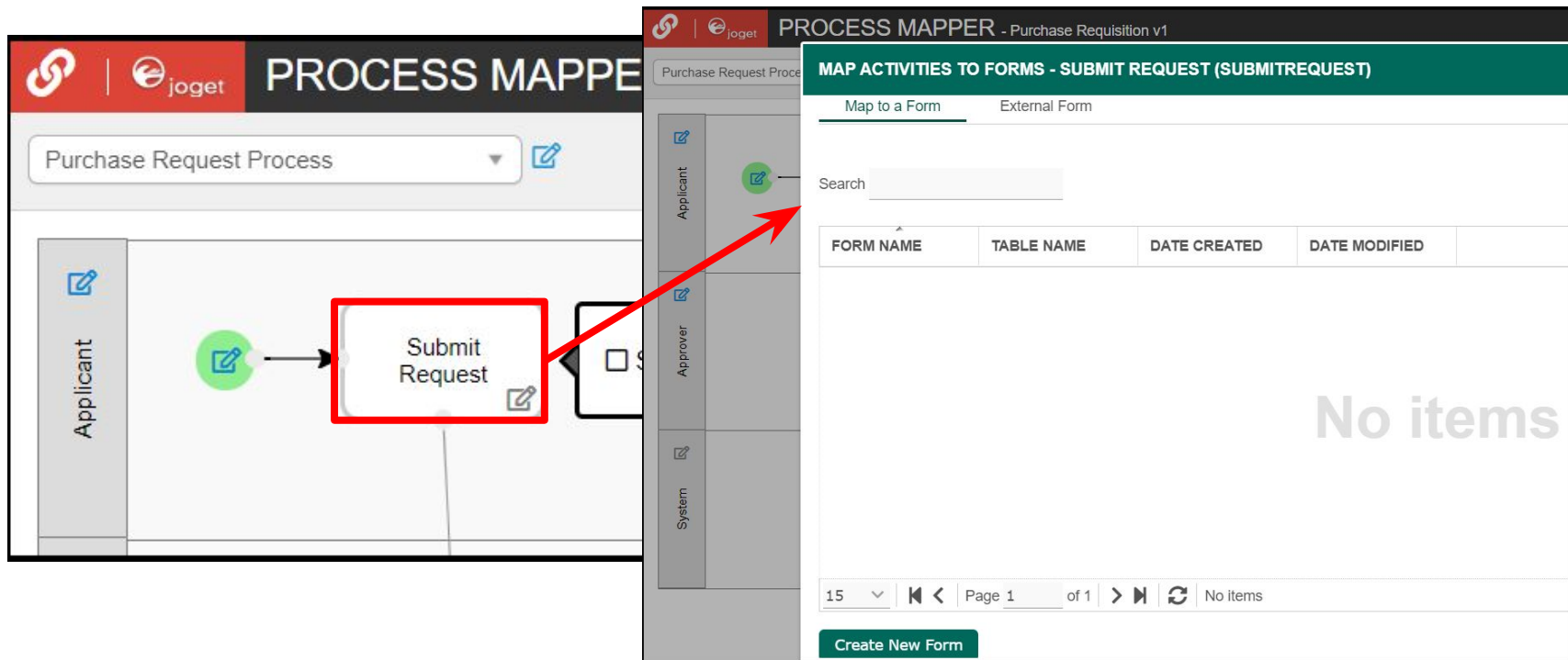
- Click on “Create New Form”



The screenshot shows the Joget DX Enterprise interface. The main content area is titled "MAP ACTIVITIES TO FORMS - SUBMIT REQUEST (SUBMITREQUEST)". It features two tabs: "Map to a Form" (selected) and "External Form". Below the tabs is a search input field. A table with the following headers is visible: "FORM NAME", "TABLE NAME", "DATE CREATED", and "DATE MODIFIED". The table is currently empty, displaying "No items". At the bottom of the table, there is a pagination control showing "Page 1 of 1" and "No items". A red box highlights the "Create New Form" button located at the bottom left of the main content area.

# Alternative Way of Creating New Form

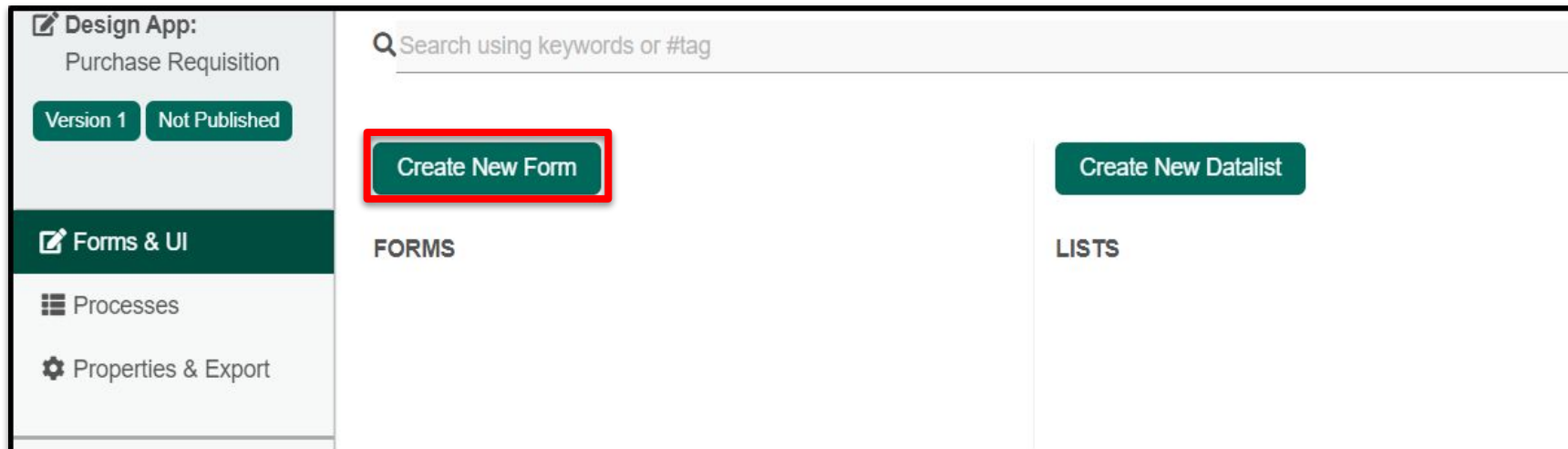
- One can also go to **Configure Mapping** button and click on the relevant **activity** to create a new form to said activity.



The screenshot displays the Joget Process Mapper interface for a 'Purchase Request Process'. The 'Submit Request' activity is highlighted with a red box. A red arrow points from this activity to the 'MAP ACTIVITIES TO FORMS - SUBMIT REQUEST (SUBMITREQUEST)' dialog box. The dialog box shows a search bar and a table with columns: FORM NAME, TABLE NAME, DATE CREATED, and DATE MODIFIED. The table is currently empty, displaying 'No items'. A 'Create New Form' button is visible at the bottom of the dialog.

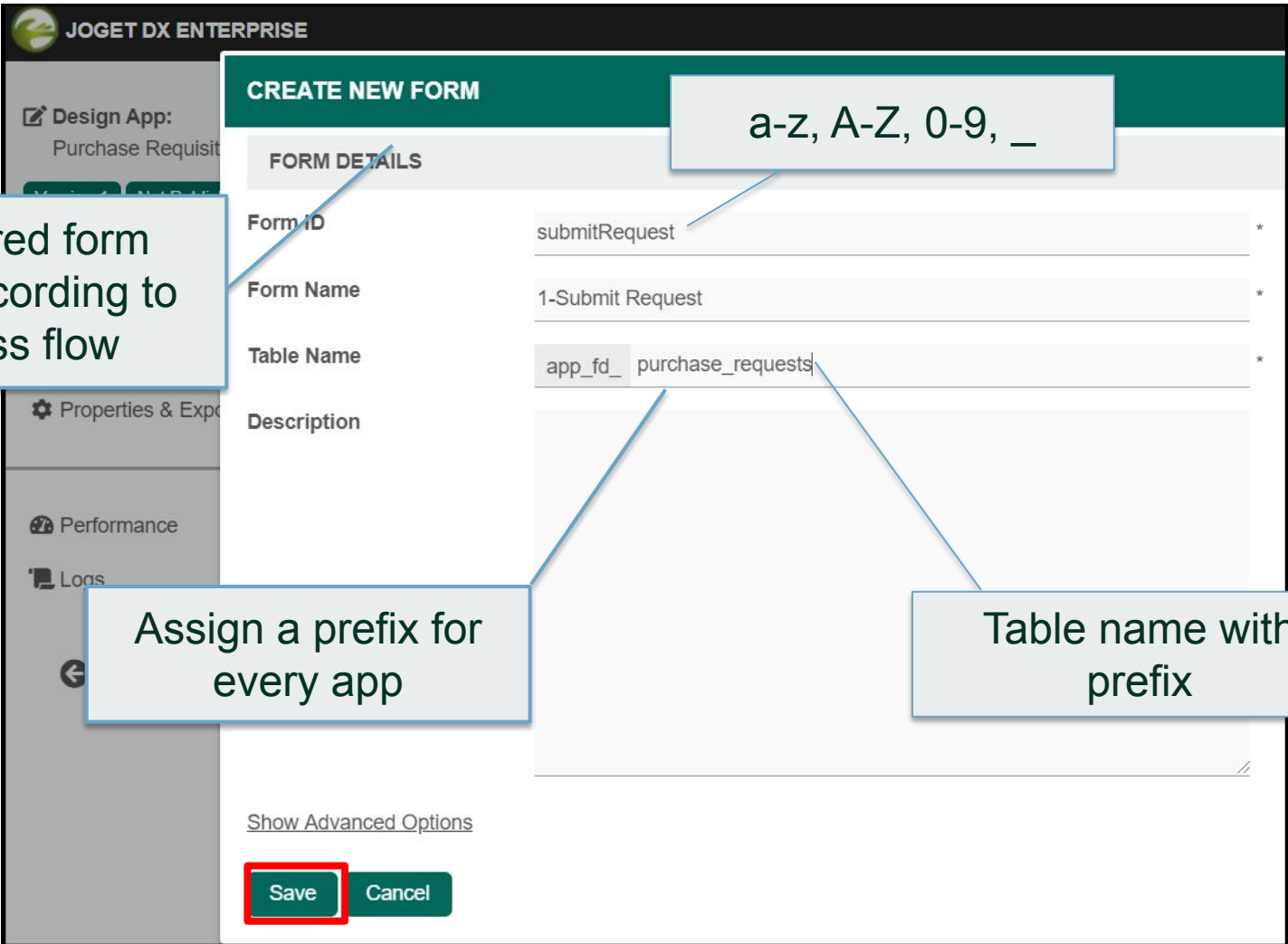
# Alternative Way of Creating New Form

- One can also go to the **Forms & UI** tab to create the new form and **add** the **newly created form** to the **activity mapping** later on.





# Create a New Form, for “Submit Request” Activity



**CREATE NEW FORM**

**FORM DETAILS**

Form ID	submitRequest *
Form Name	1-Submit Request *
Table Name	app_fd_ purchase_requests *
Description	

Numbered form name according to process flow

Assign a prefix for every app

Table name with prefix

a-z, A-Z, 0-9, \_

Show Advanced Options

Save Cancel

# App Prefix for Table Name

---

- By default, **ALL form data** will be stored into the **same database** that Joget is using.
- **All Apps** are storing into the **same database** as well.
- Therefore, it is very **important** to segregate the form data by App by declaring its own prefix before defining its designated table name.  
(e.g. purchase\_requests )

# 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.

# Naming Convention

---

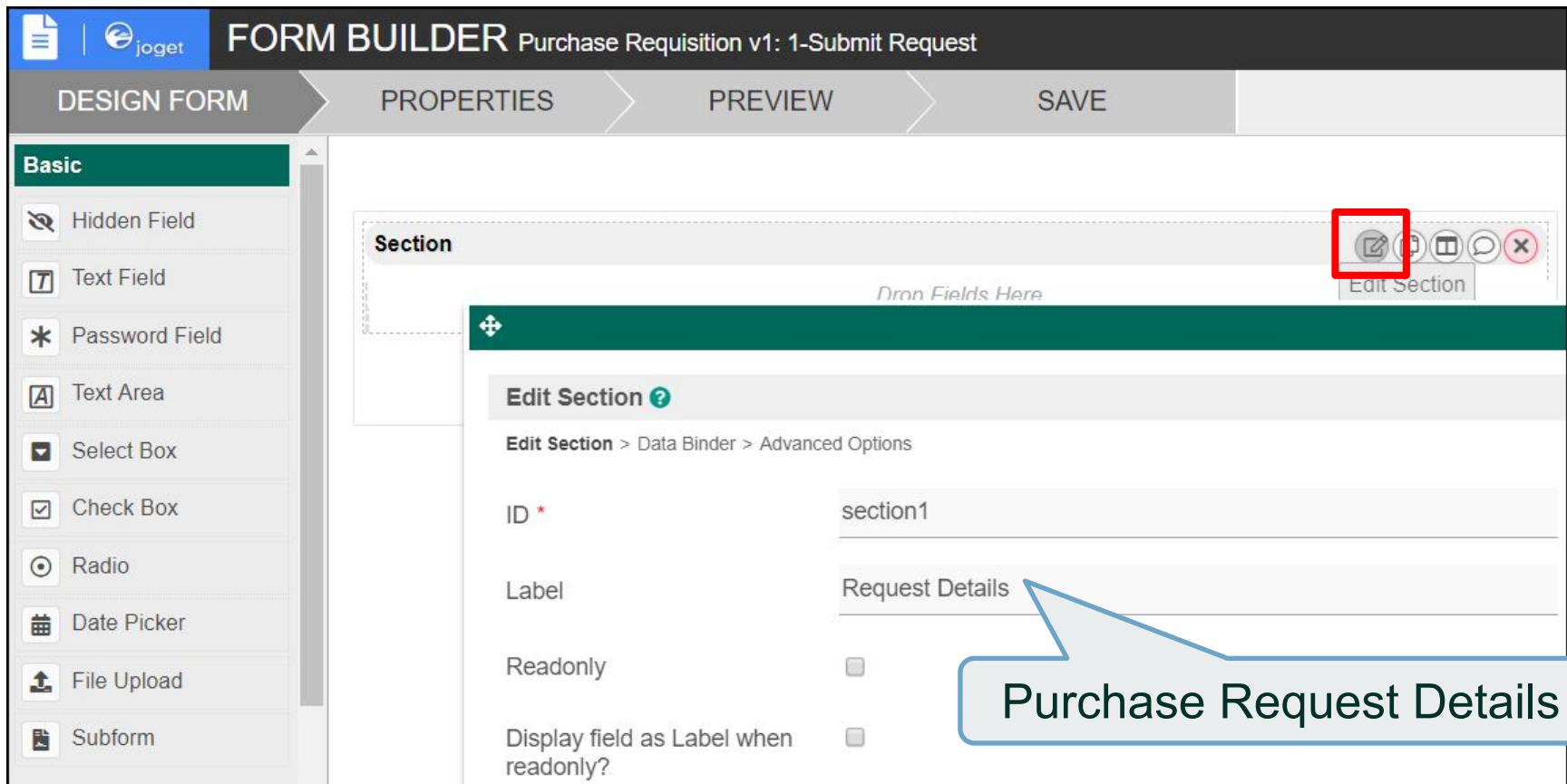
- Use **snake case** for anything related to **form data storage**.
- For example:-
  - purchase\_requests**
- Just search for “database naming convention” for more information.

E.g:

<http://stackoverflow.com/questions/7662/database-table-and-column-naming-conventions>

# Edit Section

- Form Section serves as a placeholder for form input elements. Rename it to make it meaningful.



The screenshot shows the Joget Form Builder interface for a form titled "Purchase Requisition v1: 1-Submit Request". The interface includes a top navigation bar with "DESIGN FORM", "PROPERTIES", "PREVIEW", and "SAVE" tabs. On the left, there is a "Basic" widget palette with options like Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, File Upload, and Subform. The main workspace shows a "Section" placeholder with a red box around the "Edit Section" icon. The "Edit Section" dialog is open, showing the following details:

- Section:** section1
- Label:** Request Details
- Readonly:**
- Display field as Label when readonly?:**

A callout bubble points to the "Request Details" label field with the text "Purchase Request Details".

# Insert Date Picker

- Insert a **Date Picker** form element by dragging it into the section placeholder.

The screenshot displays the Joget Form Builder interface for a form titled "Purchase Requisition v1: 1-Submit Request". The interface is divided into four main sections: "DESIGN FORM", "PROPERTIES", "PREVIEW", and "SAVE". The "DESIGN FORM" section is active, showing a list of form elements on the left and a preview of the form on the right.

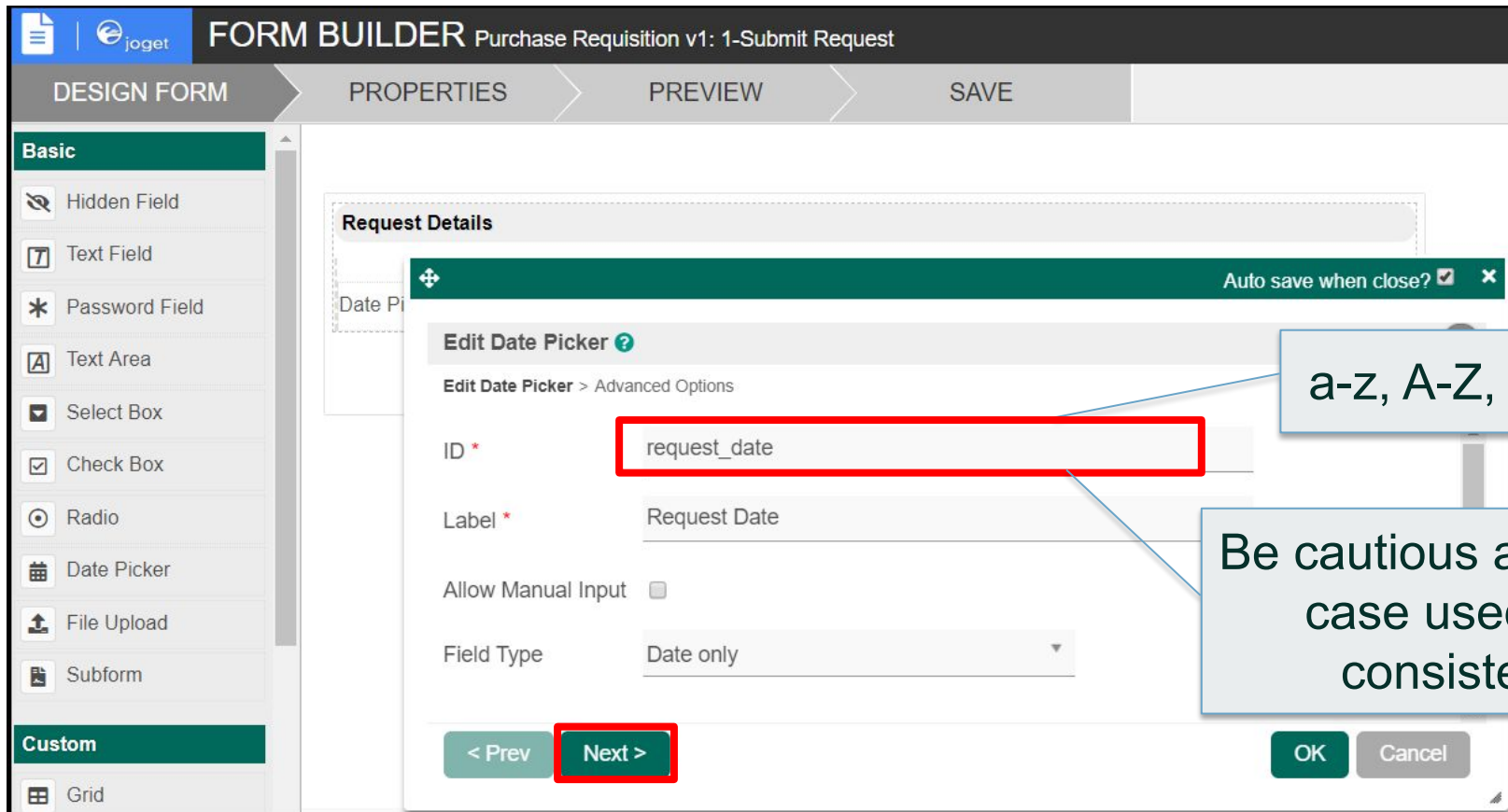
The "Basic" category of form elements is expanded, listing the following options:

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker** (highlighted with a red box)
- File Upload
- Subform

The "Request Details" section placeholder in the preview area contains a "Date Picker" element and a "Drop Fields Here" area. A red arrow points from the "Date Picker" element in the left-hand menu to the "Date Picker" element in the "Request Details" section placeholder.

# Edit The First Date Picker

- Set the **ID** according to database naming convention – use underscore instead of camel case to separate words.



The screenshot displays the Joget Form Builder interface for editing a date picker. The 'Edit Date Picker' dialog box is open, showing the following fields:

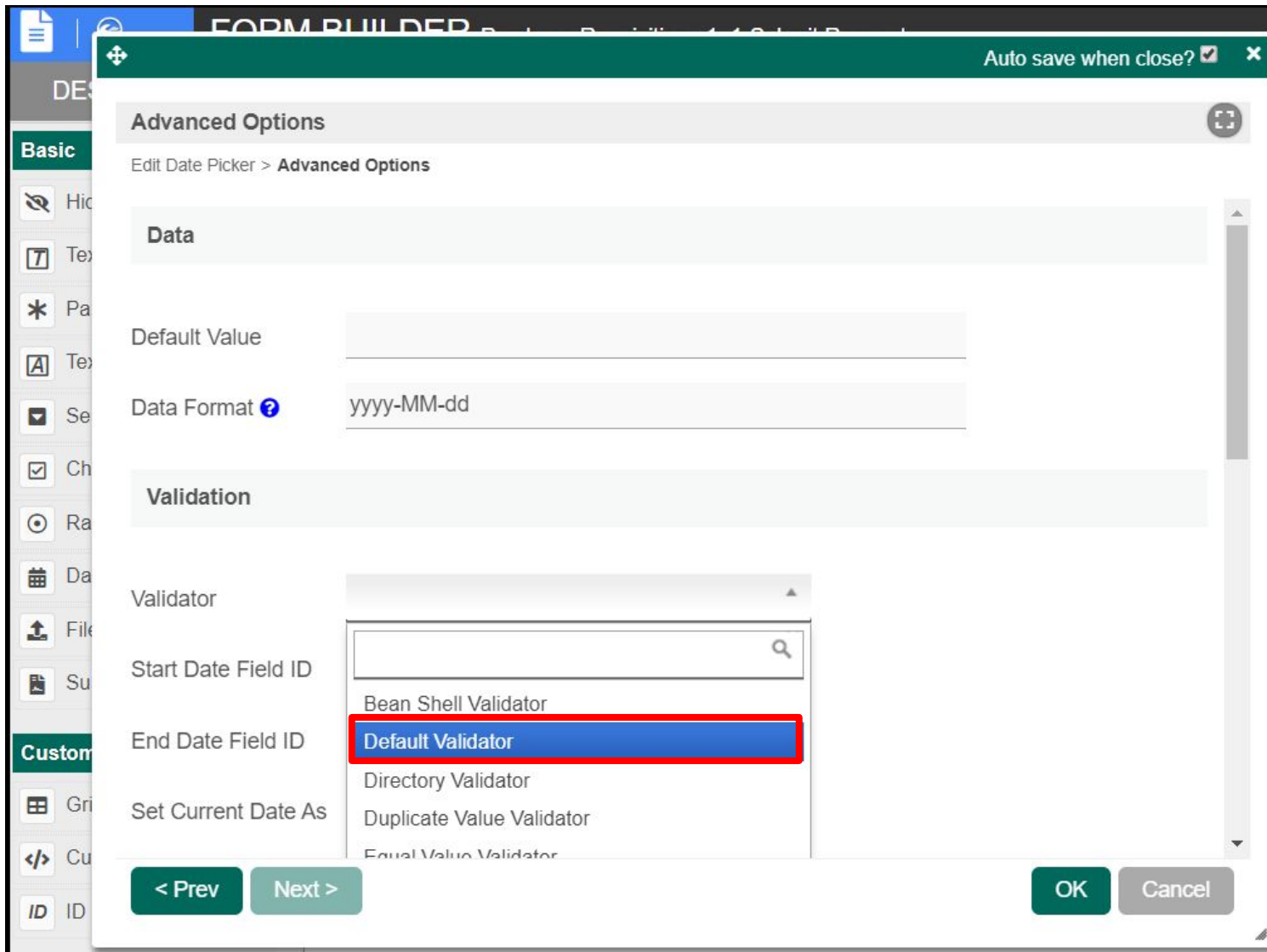
- ID \***: request\_date (highlighted with a red box)
- Label \***: Request Date
- Allow Manual Input**:
- Field Type**: Date only

Callout boxes provide additional instructions:

- A box pointing to the ID field contains the text: "a-z, A-Z, 0-9, \_"
- A box pointing to the 'Next >' button contains the text: "Be cautious about the case used, be consistent!"

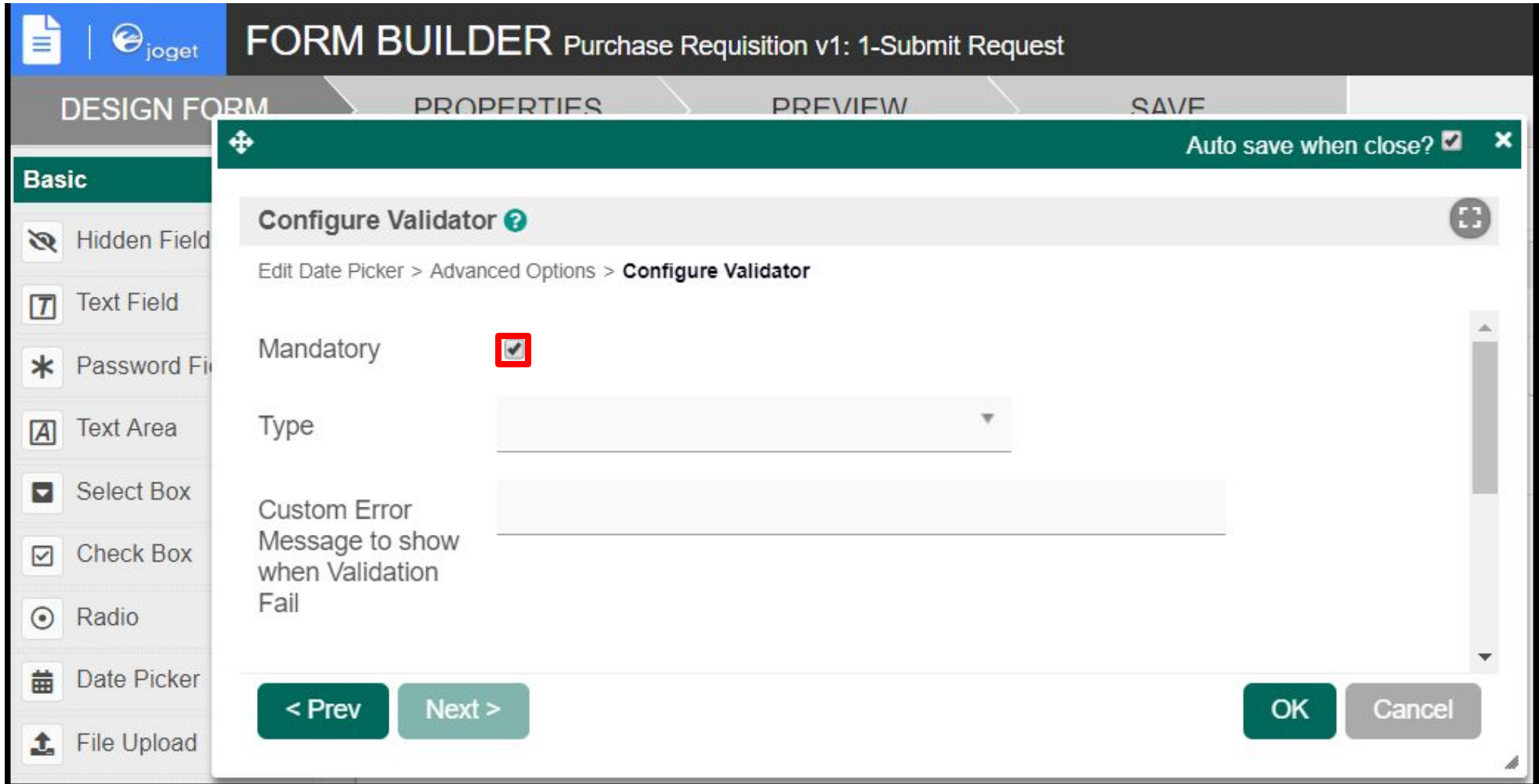
The interface also shows a 'Basic' sidebar with various form field options and a 'Request Details' section in the background.

# Set The Date Picker as Mandatory Field





# Set The Date Picker as Mandatory Field



The screenshot displays the Joget Form Builder interface for a form titled "Purchase Requisition v1: 1-Submit Request". The "DESIGN FORM" tab is active. A "Basic" sidebar on the left lists various field types, with "Date Picker" selected. A "Configure Validator" dialog is open, showing the "Mandatory" checkbox checked. The "Type" dropdown is empty, and the "Custom Error Message to show when Validation Fail" field is also empty. The dialog includes navigation buttons "< Prev" and "Next >" and "OK" and "Cancel" buttons.

FORM BUILDER Purchase Requisition v1: 1-Submit Request

DESIGN FORM PROPERTIES PREVIEW SAVE

Auto save when close?  X

Basic

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker
- File Upload

Configure Validator ?

Edit Date Picker > Advanced Options > Configure Validator

Mandatory

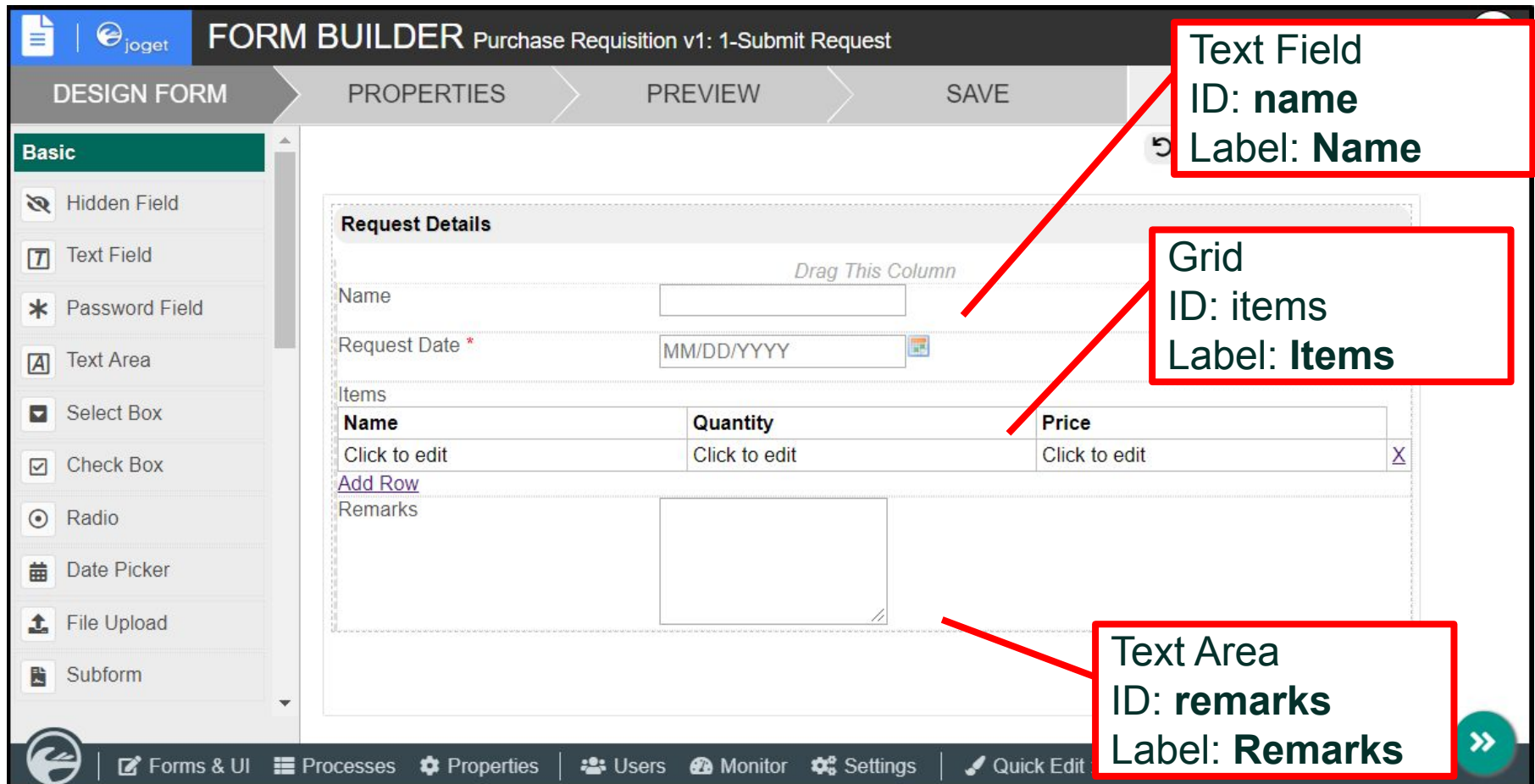
Type

Custom Error Message to show when Validation Fail

< Prev Next > OK Cancel

# Complete the Design of “Submit Request” Form

- Add text field, grid, and text area.



**FORM BUILDER** Purchase Requisition v1: 1-Submit Request

DESIGN FORM | PROPERTIES | PREVIEW | SAVE

**Basic**

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker
- File Upload
- Subform

**Request Details**

Name

Request Date \*

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

[Add Row](#)

Remarks

**Text Field**  
ID: name  
Label: Name

**Grid**  
ID: items  
Label: Items

**Text Area**  
ID: remarks  
Label: Remarks

Forms & UI | Processes | Properties | Users | Monitor | Settings | Quick Edit

# Complete the Design of “Submit Request” Form

- Preview and save.

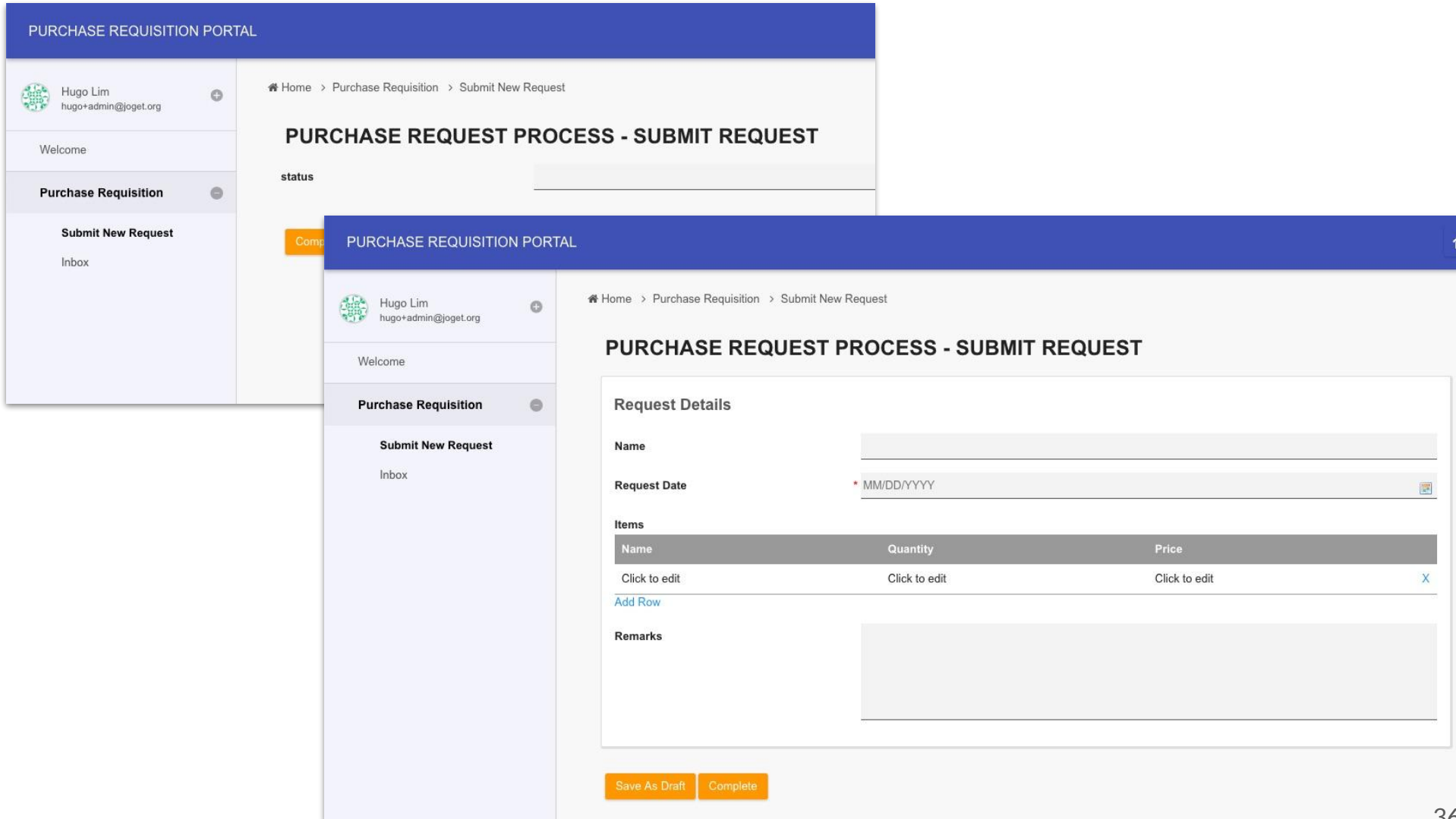
The screenshot displays the Joget Form Builder interface for a form titled "Purchase Requisition v1: 1-Submit Request". The interface is in the "PREVIEW" mode, as indicated by the selected tab. A red box highlights the "Preview and Save" button, which is a prominent feature in the preview area. The form preview shows a "Request Details" section with the following fields:

- Name:
- Request Date \*:
- Items table:

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

Below the table, there is a "Remarks" field with a text area. The interface also includes a sidebar with various form components like Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, File Upload, and Subform. The bottom navigation bar shows options like Forms & UI, Processes, Properties, Users, Monitor, Settings, and Quick Edit.

# How the Form Looks Like When You Launch “Run Process” Now?



The screenshot displays the 'PURCHASE REQUISITION PORTAL' interface. The top navigation bar is blue with the text 'PURCHASE REQUISITION PORTAL'. Below this, a breadcrumb trail shows 'Home > Purchase Requisition > Submit New Request'. The main content area is titled 'PURCHASE REQUEST PROCESS - SUBMIT REQUEST'. On the left, a sidebar menu includes 'Purchase Requisition' (expanded), 'Submit New Request', and 'Inbox'. The user profile for 'Hugo Lim' is visible at the top left of the main content area.

The form itself is titled 'PURCHASE REQUEST PROCESS - SUBMIT REQUEST' and contains the following sections:

- Request Details:**
  - Name:** A text input field.
  - Request Date:** A date picker field with a red asterisk and a calendar icon.
- Items:** A table with columns for Name, Quantity, and Price. Each cell contains a 'Click to edit' link. There is an 'Add Row' link below the table.
- Remarks:** A large text area for entering additional information.

At the bottom of the form, there are two orange buttons: 'Save As Draft' and 'Complete'.

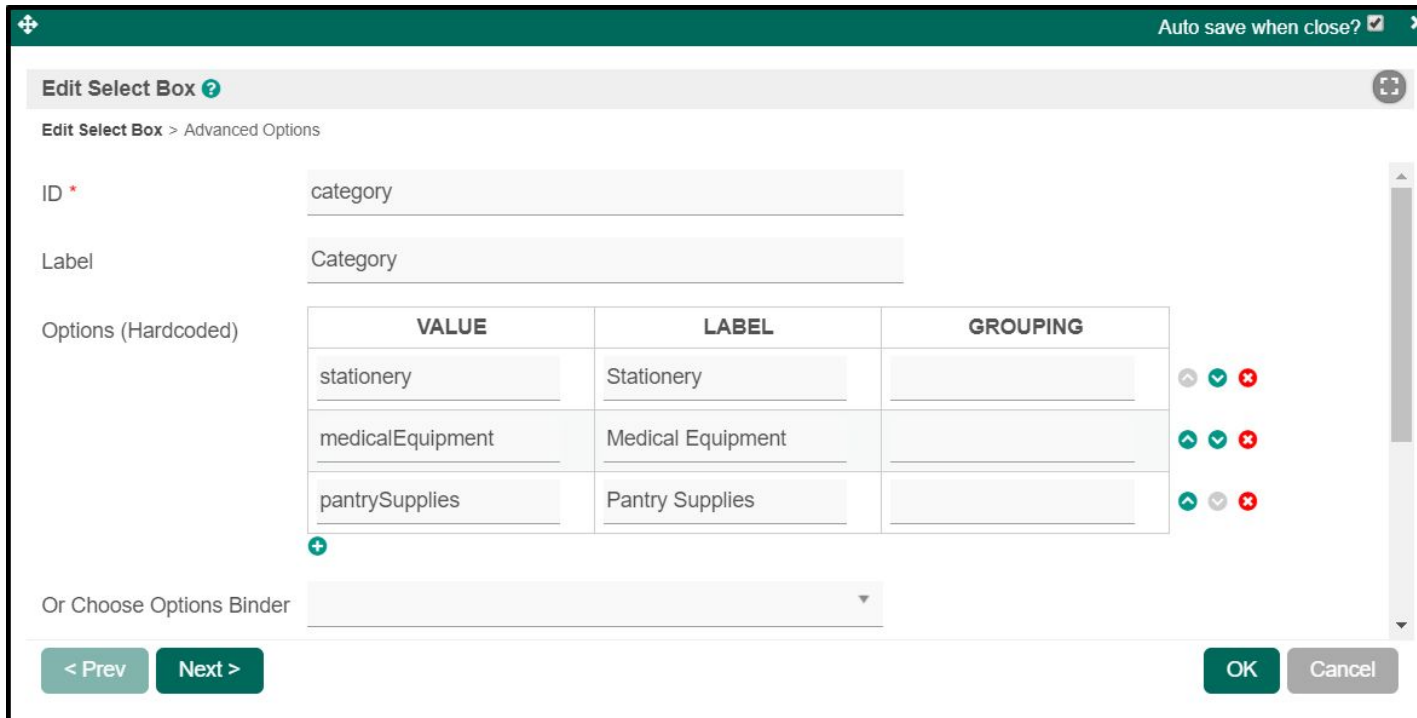
# Exercise 1

---

- Fill in the form, leave “Request Date” empty to test out form validation.
- Try to submit the form.
- Now fill up all the fields and complete the assignment.

# Exercise 2

- Add a new **Select Box** called “Category” with the following options:-
  - Stationery
  - Medical Equipment
  - Pantry Supplies



Auto save when close?

**Edit Select Box** ?

Edit Select Box > Advanced Options

ID \*

Label

Options (Hardcoded)

VALUE	LABEL	GROUPING
<input type="text" value="stationery"/>	<input type="text" value="Stationery"/>	<input type="text"/>
<input type="text" value="medicalEquipment"/>	<input type="text" value="Medical Equipment"/>	<input type="text"/>
<input type="text" value="pantrySupplies"/>	<input type="text" value="Pantry Supplies"/>	<input type="text"/>

Or Choose Options Binder

< Prev    Next >    OK    Cancel

# Exercise 3 - Optional


- Customize the look and feel of how the form is rendered by modifying its CSS.
  - Add a **Custom HTML** form element into the bottom of the form.
  - Edit it, add the following code into **Custom HTML** property.

```
<style type="text/css">
.form-cell .label,
.subform-cell .label{
    width: 100%;
}
</style>
```

**Request Details**

*Drag This Column*

Name

Request Date \*  

Category  ▼

Items

Name	Quantity	Price
Click to edit	Click to edit	Click to edit
<a href="#">Add Row</a> <span style="float: right;">✕</span>		

Remarks

Custom HTML CUSTOM HTML

# Discussion

---

- Do you know that you can still modify forms even after (your app has been published) and users started using them?
- What will happen to the submitted data if you delete form elements that is previously added in the form?



# Chapter Review

---

- Associating activity with form
- Creating Form and understanding the basic elements and validations.
- Accessing assignment with a Form associated.

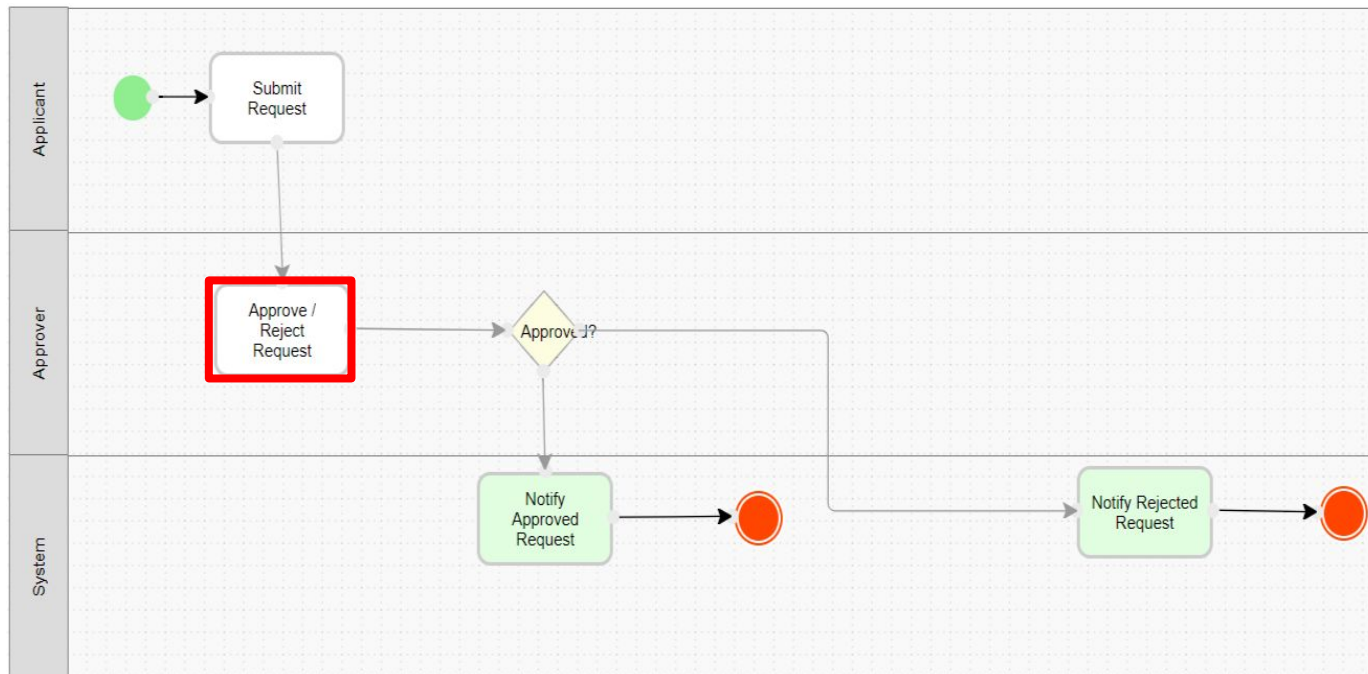


# Chapter 3

## Controlling Process by Using Form

# Association With the Process

- Designing the second form for the second activity in the flow.
- Form controls the next course of action in the process flow.



# Create a New Form for “Approve / Reject Request” Activity

- Click on “Add/Edit Form” on “Approve / Reject Request” activity.  
Fill in the following details.

CREATE NEW FORM

FORM DETAILS

Form ID	<input type="text" value="approvalRequest"/>
Form Name	<input type="text" value="2-Approve / Reject Request"/>
Table Name	<input type="text" value="app_fd_ purchase_requests"/>
Description	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>

[Show Advanced Options](#)

Save
Cancel

# Using Subform

- **Subform** allows one to make **reference** to existing form without duplicating form design again.

Using "1-Submit Request" as Subform

## Approve / Reject Request Form

**Request Details**

*Drag This Column*

Name

Request Date \*

Category  ▼

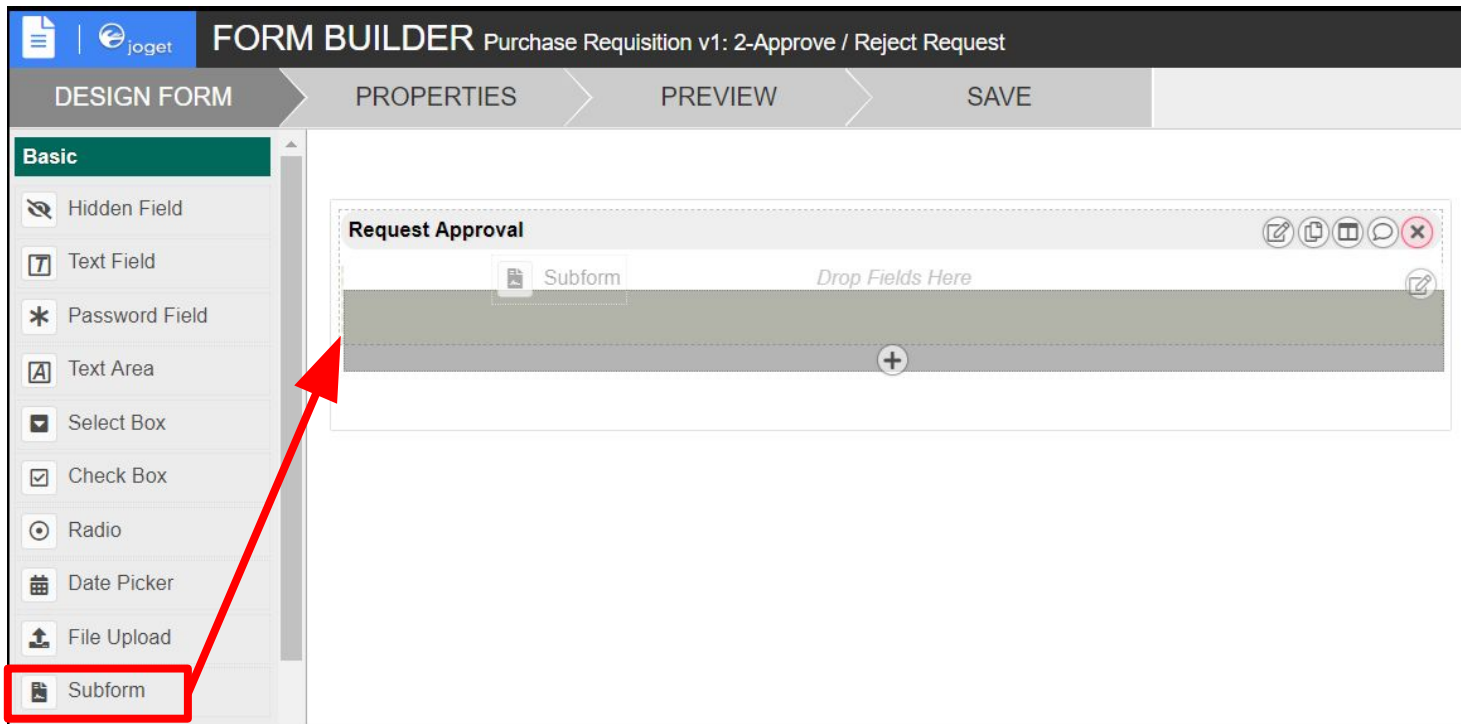
Items		
Name	Quantity	Price
Click to edit	Click to edit	Click to edit
<a href="#">Add Row</a>		

Remarks

Custom HTML CUSTOM HTML

# Using Subform

- Rename the section and add in **Subform**.
- Edit the **Subform** to point to the first form.
- Set it to read only.



# Using "1-Submit Request" As Subform

FORM BUILDER Purchase Requisition v1: 2-Approve / Reject Request

DESIGN FORM    PROPERTIES    PREVIEW    SAVE

**Basic**

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker
- File Upload
- Subform

**Custom**

- Grid
- Custom HTML
- ID Generator Field

**Request Approval**

*Drag This Column*

**Request Details**

Name

Request Date \*

Category

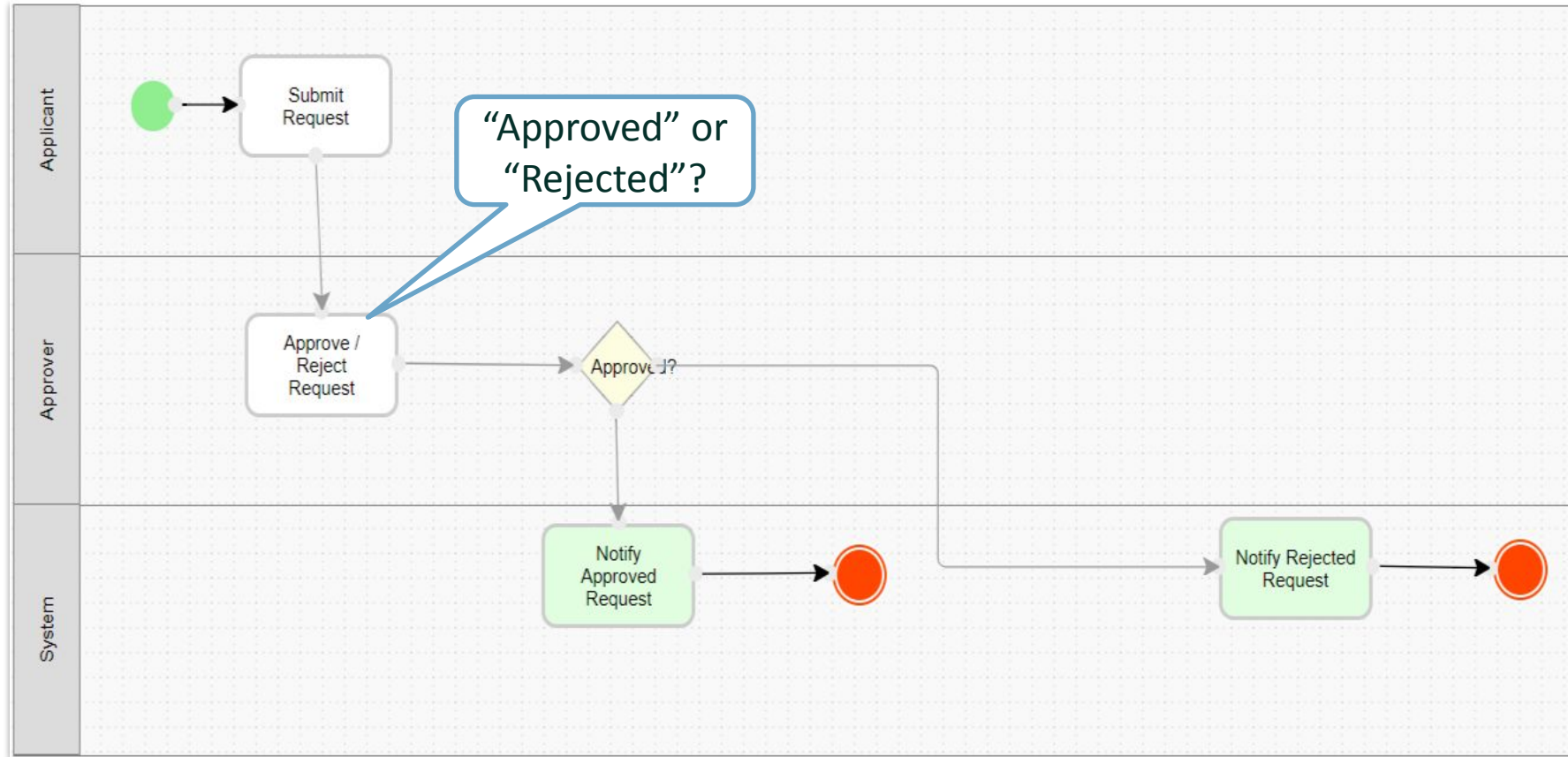
Items

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

[Add Row](#)

Remarks

# What About the Approval Status?





# Insert Radio Button

- Insert a Radio into the bottom of the form.
- Configure accordingly.

**Edit Radio** ?

Edit Radio > Advanced Options

ID \*

Label

Options (Hardcoded)

VALUE	LABEL	GROUPING
<input type="text" value="Approved"/>	<input type="text" value="Approved"/>	<input type="text"/>
<input type="text" value="Rejected"/>	<input type="text" value="Rejected"/>	<input type="text"/>

⬆️ ✓ ✖️  
⬆️ ✓ ✖️

+

Or Choose Options Binder

< Prev
Next >
OK
Cancel

Note: Value is **CASE-SENSITIVE**. Check your process design **transition conditions** to confirm.

# Radio Button Options

- Go to the next tab and set the **Workflow Variable** to “status”.
- Upon submission of this form, value defined here will also set the respective workflow variable.

The screenshot shows a dialog box titled "Advanced Options" for editing a radio button. The dialog is divided into two main sections: "UI" and "Workflow".

**UI Section:**

- Readonly:** A checkbox that is currently unchecked.
- Display field as Label when readonly?:** A checkbox that is currently unchecked.

**Workflow Section:**

- Workflow Variable:** A text input field containing the value "status". This field is highlighted with a red rectangular border.

At the bottom of the dialog, there are navigation buttons: "< Prev", "Next >", "OK", and "Cancel".

# Process Review

- Recall what we have declared and set in the route's conditions.

PLUGIN CONFIGURATION - APPROVED? (ROUTE1)
✕

Configure Simple Rules Decision
⛶

Rules \*

**IF** 🗑️ | ⋮

And ▾ + Add Condition + Add Group

! status Equal To ▾ Approved 🗑️

**THEN** + Add Action

Transition ▾ transition3 (Notify Approved Request) ▾ 🗑️

+ Add Rule

**ELSE THEN** + Add Action

Transition ▾ transition4 (Notify Rejected Request) ▾ 🗑️

Submit
Change Plugin

### Edit Radio

Edit Radio > Advanced Options

ID \*

Label

Options (Hardcoded)

Value	Label
<input type="text" value="Approved"/>	<input type="text" value="Approved"/>
<input type="text" value="Rejected"/>	<input type="text" value="Rejected"/>

Or Choose Options Binder

### Options

Display field as Label when readonly?

Workflow

Workflow Variable

### Configure Simple Rules Decision

Rules \*

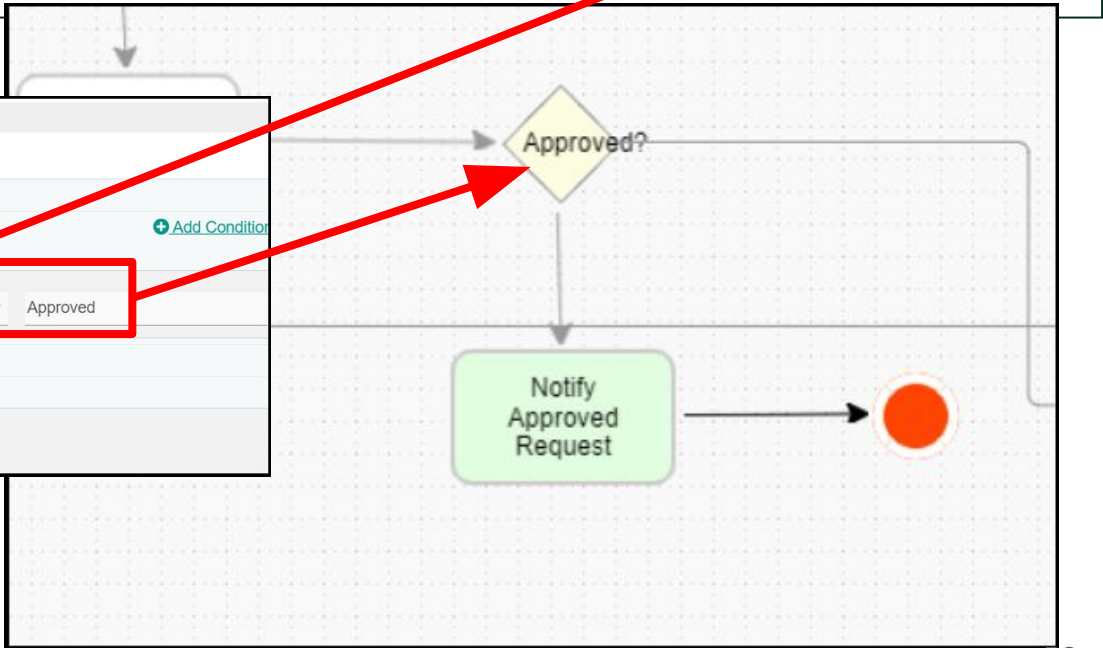
IF

And

<input type="text" value="status"/>	Equal To	<input type="text" value="Approved"/>
-------------------------------------	----------	---------------------------------------

THEN

Transition



# Request Approval Form

FORM BUILDER Purchase Requisition v1: 2-Approve / Reject Request

DESIGN FORM
PROPERTIES
PREVIEW
SAVE

**Basic**

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker
- File Upload
- Subform

**Custom**

- Grid
- Custom HTML
- ID Generator Field

**Request Approval**

*Drag This Column*

**Request Details**

Name

Request Date \*

Category

Name	Quantity	Price
Click to edit	Click to edit	Click to edit
<a href="#">Add Row</a>		

Remarks

Approval
 Approved
 Rejected

# Continue With the Assignment

PURCHASE REQUISITION PORTAL

Hugo Lim  
hugo+admin@joget.org

Welcome

Purchase Requisition

Submit New Request

**Inbox**

Home > Purchase Requisition > Inbox

### PURCHASE REQUEST PROCESS - APPROVE REQUEST

#### Request Approval

##### Request Details

Name: John Smith

Request Date: \* 01/23/2020

Category: Stationery

##### Items

Name	Quantity	Price
Pencil	5	10

Remarks

Approval  Approved  Rejected

Save As Draft **Complete** Cancel

# Exercise

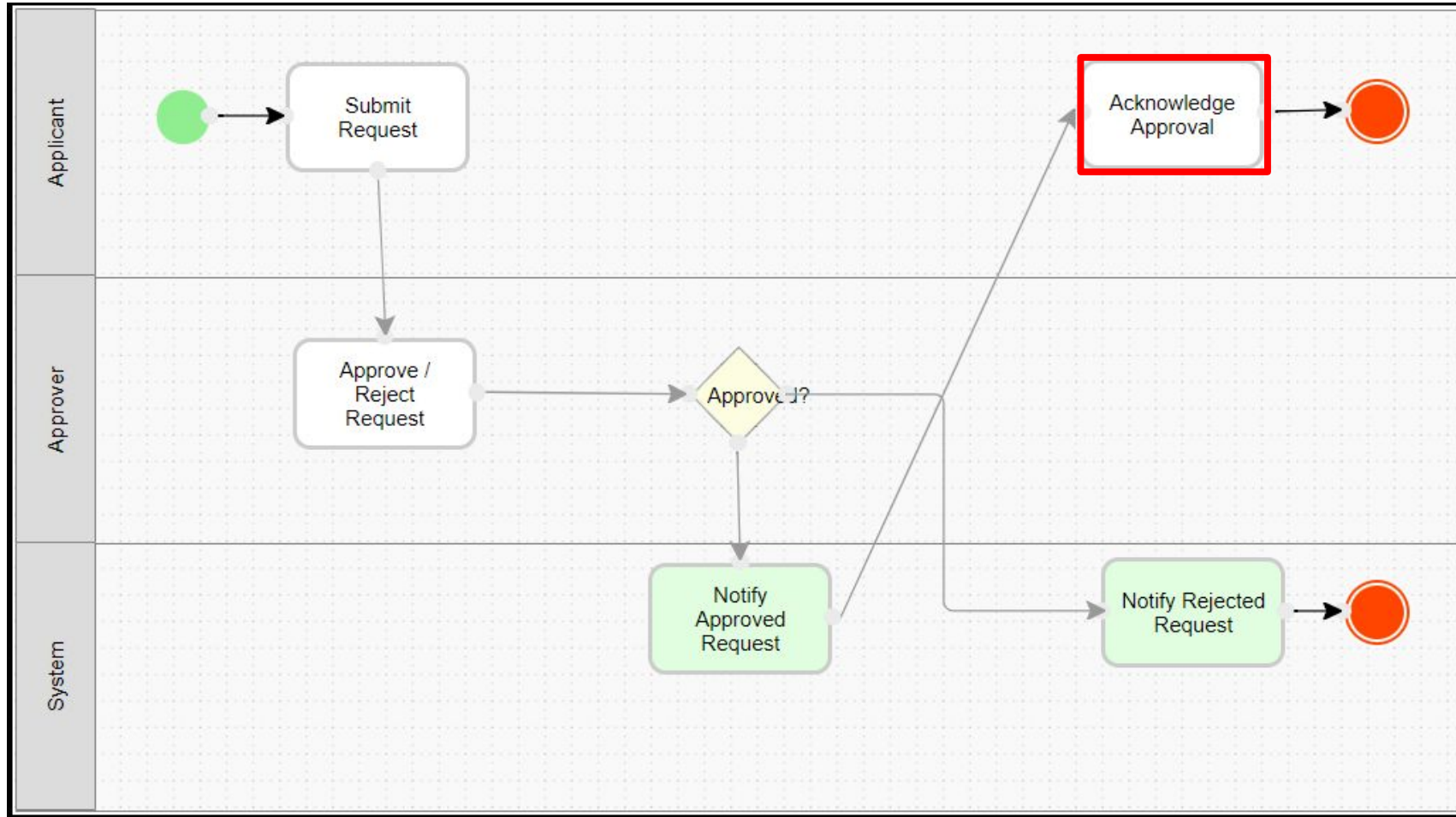
---

- Refresh your memory on module 3 - designing process!

Design Form for “**Acknowledge Approved**” activity

- Add an activity to your purchase request process, after the email tool in the “approved” lane.
- Create a new Form named “**3-Acknowledge Approved**” form
- Map the form to “**Acknowledge Approved**” activity.
- Point it back to the **same** table used before.
- In the form design, add a “Subform” and refer it to the “**2-Approve / Reject Request**” form.

# “Acknowledge Approved” Activity Added in Process



Note: Pay attention to **WHERE** you are placing the “Acknowledge Approved” activity.



# Acknowledge Approved Form

joget
**FORM BUILDER** Purchase Requisition v1: 3-Acknowledge Approved
admin

DESIGN FORM
PROPERTIES
PREVIEW
SAVE
GENERATE APP

**Basic**

- Hidden Field
- Text Field
- Password Field
- Text Area
- Select Box
- Check Box
- Radio
- Date Picker
- File Upload
- Subform

**Custom**

- Grid
- Custom HTML
- ID Generator Field

**Approval Acknowledgement**

*Drag This Column*

**Request Approval**

**Request Details**

Name

Request Date \*

Category

Name	Quantity	Price
Click to edit	Click to edit	Click to edit
<a href="#">Add Row</a>		

Remarks

Approval
 Approved
 Rejected


# Complete the Process Cycle

---

- Go back to your Inbox.
- Open “**Acknowledge Approved**” assignment.  
Verify the form created and complete the process.

# Complete the Process Cycle

PURCHASE REQUISITION PORTAL

 Hugo Lim  
hugo+admin@joget.org

Welcome

**Purchase Requisition**

Submit New Request

**Inbox**

Home > Purchase Requisition > Inbox

## PURCHASE REQUEST PROCESS - ACKNOWLEDGE APPROVAL

### Approval Acknowledgement

#### Request Approval

##### Request Details

**Name**

**Request Date**

**Category**

Name	Quantity	Price
Pencil	5	10

**Remarks**

**Approval**  Approved  Rejected

Save As Draft
Complete
Cancel

# Discussion

---

- Curious on where the form data is actually stored?  
Use your preferred SQL Client and try to look for the form data submitted earlier.

# Chapter Review

---

- Learn about Subform.
- Control flow of process using **Workflow Variable** mapping via the **Form Field**.

# Module Review

---

We have learnt to:

1. Create Form and assign it to Process Activity with basic Form Elements.
2. Understand the Subform concept.
3. Map Form Field to Workflow Variable for routing purpose.
4. Verify Process with the Form attached.

# Recommended Further Learning

---

- Improving your Form design and Presentation.
- Creating a Datalist based on the Forms designed.

# 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)