

RADS

Designing your first Form

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

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 I

Introduction to Form Builder

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 RADS's plugin architecture.

Form Builder

FORM BUILDER CRM Community v1: Account Form (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

Account Details

Account ID *

Drag This Column

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

© RADS

Form Builder

ADVANCED TOOLS

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

```
1 {
2   "className": "net.purwana.rads.apps.form.model.Form",
3   "properties": {
4     "loadBinder": {
5       "className": "net.purwana.rads.apps.form.lib.WorkflowFormBinder",
6       "properties": {}
7     },
8     "name": "Account Form",
9     "description": "",
10    "id": "crm_account",
11    "storeBinder": {
12      "className": "net.purwana.rads.apps.form.lib.WorkflowFormBinder",
13      "properties": {}
14    },
15    "tableName": "crm_account"
16  },
17  "elements": [
18    {
19      "elements": [
20        {
21          "elements": [
22            {
23              "className": "net.purwana.rads.apps.form.lib.TextField",
24              "properties": {
25                "size": "",
26                "readonly": "",
27                "validator": {
28                  "className": "net.purwana.rads.apps.form.lib.DefaultValidator"
```

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 **Userveview**.

Form or CRUD in Userview

The image illustrates the workflow for creating a form or CRUD in Userview, showing the design phase, the final application view, and the user view configuration.

Form Builder (Top Left): The 'FORM BUILDER' interface for 'Leave App v1: Submit Leave'. It shows a design form with sections for 'Account Details' and 'Address Details'. A red box highlights the 'Account Details' section, which includes fields for 'Account ID' and 'Account Name'.

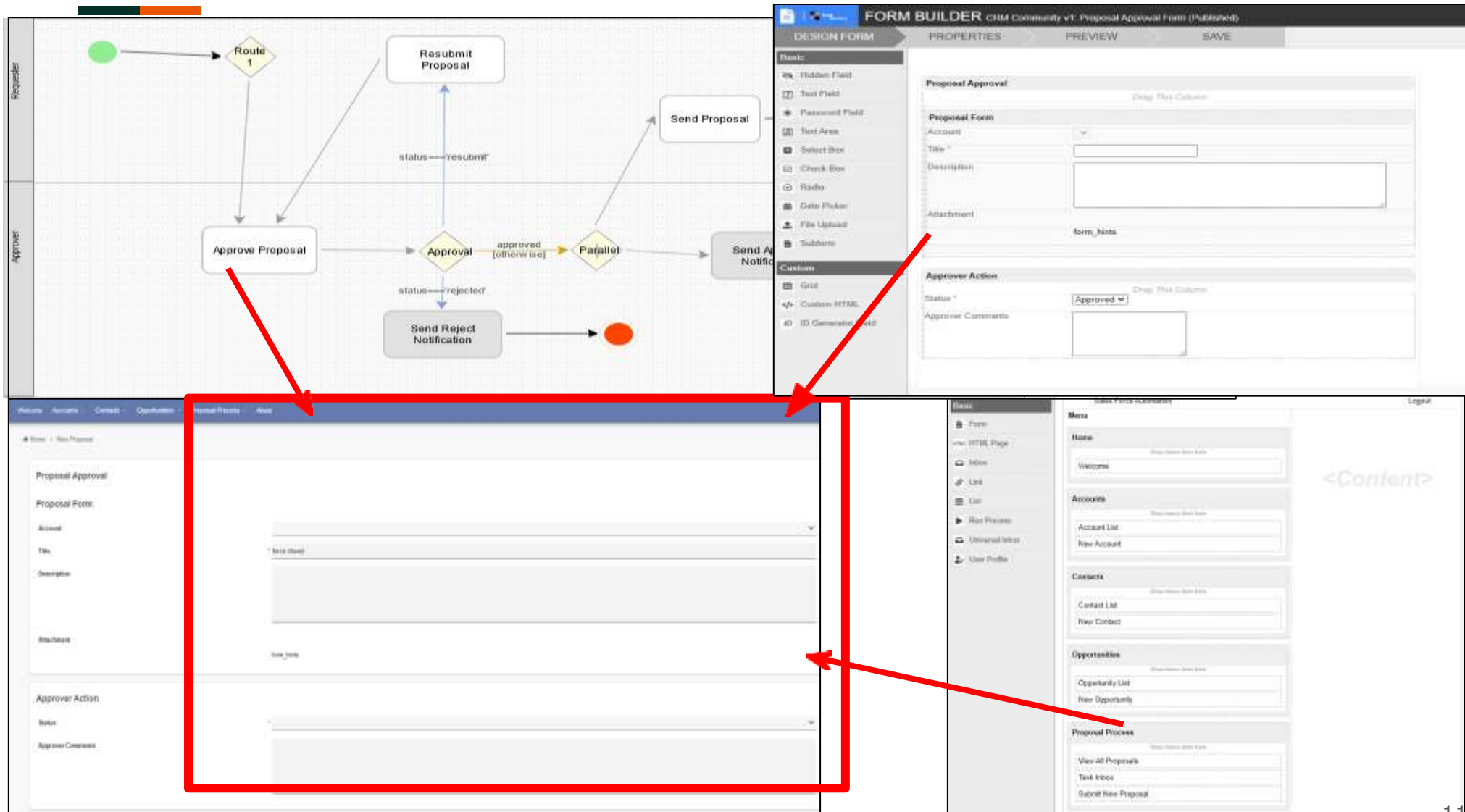
User View Builder (Bottom Center): The 'USER VIEW BUILDER' interface for 'CRM Community'. It shows a menu structure with 'Home', 'Accounts', and 'Contacts'. A red box highlights the 'Accounts' menu item, and a red arrow points from the 'Account List' sub-item to the live application view.

Application View (Top Right): The final application view for 'APPLICATIONNAME'. It shows the 'Account Details' and 'Address Details' form, which is a direct result of the design phase.

Where and How Form is Used?

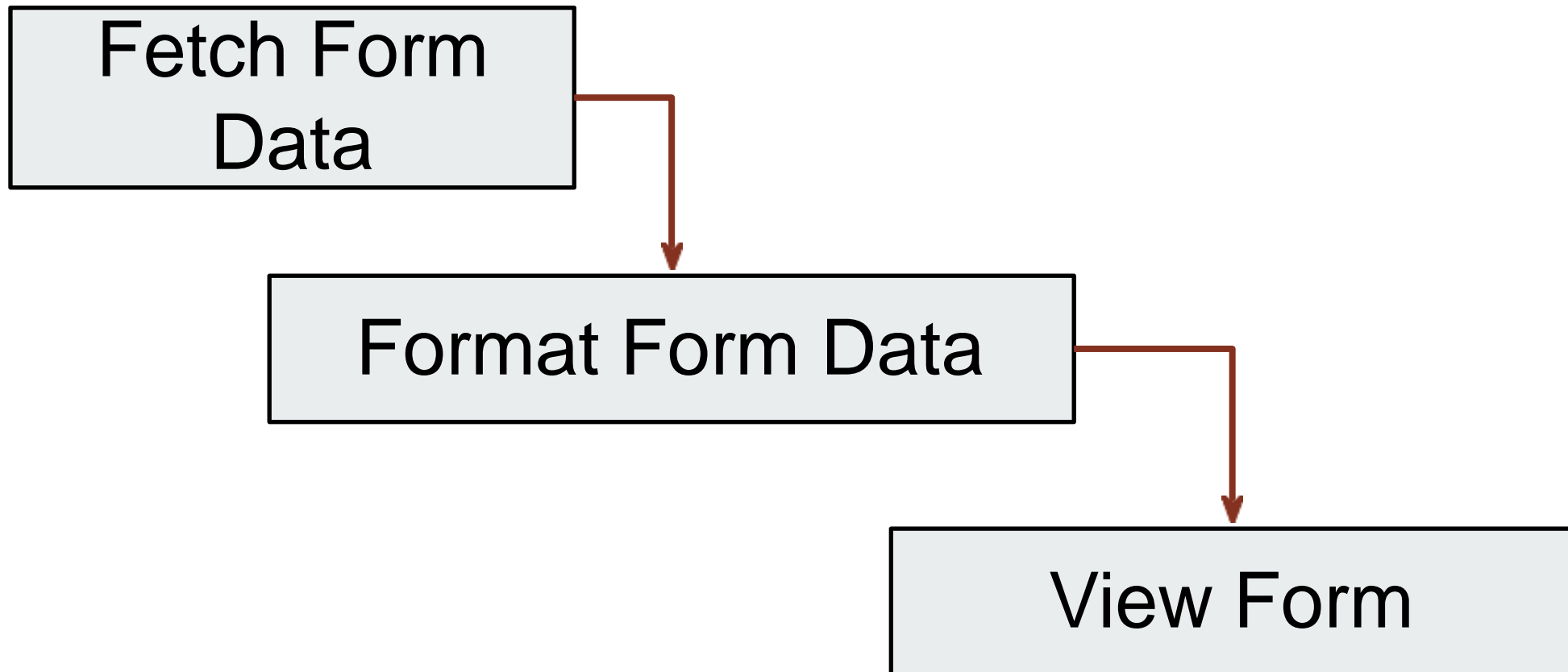
- Part of a Process
 - Each form record relates to a process instance.
 - Made accessible through the process activity's assignment view in the Inbox.
 - Inbox is made accessible via the **App Center** or **Userview**.

Form - Assignment - Inbox - Userview



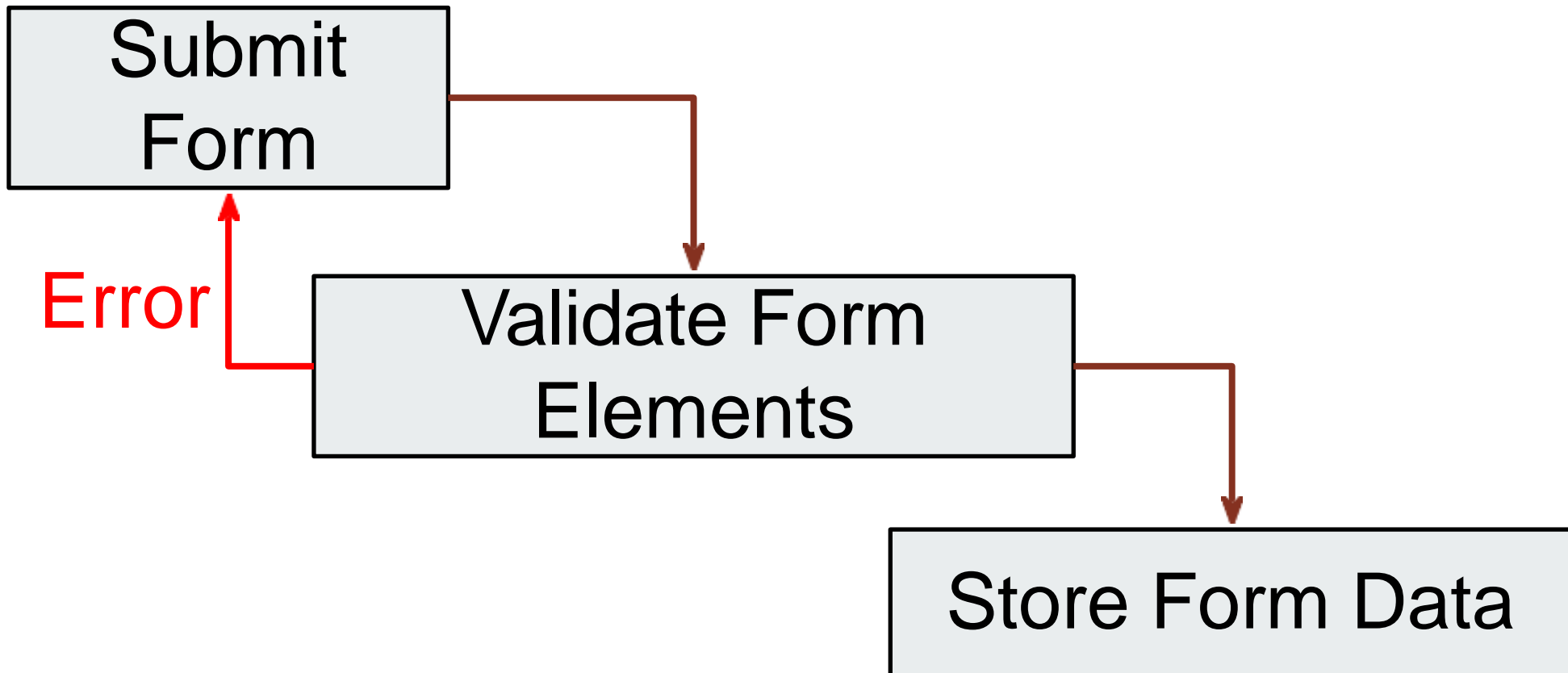
Viewing a Form

- 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 RADS running as the server.

Chapter Review

- General understanding on how Form and Form Builder works.

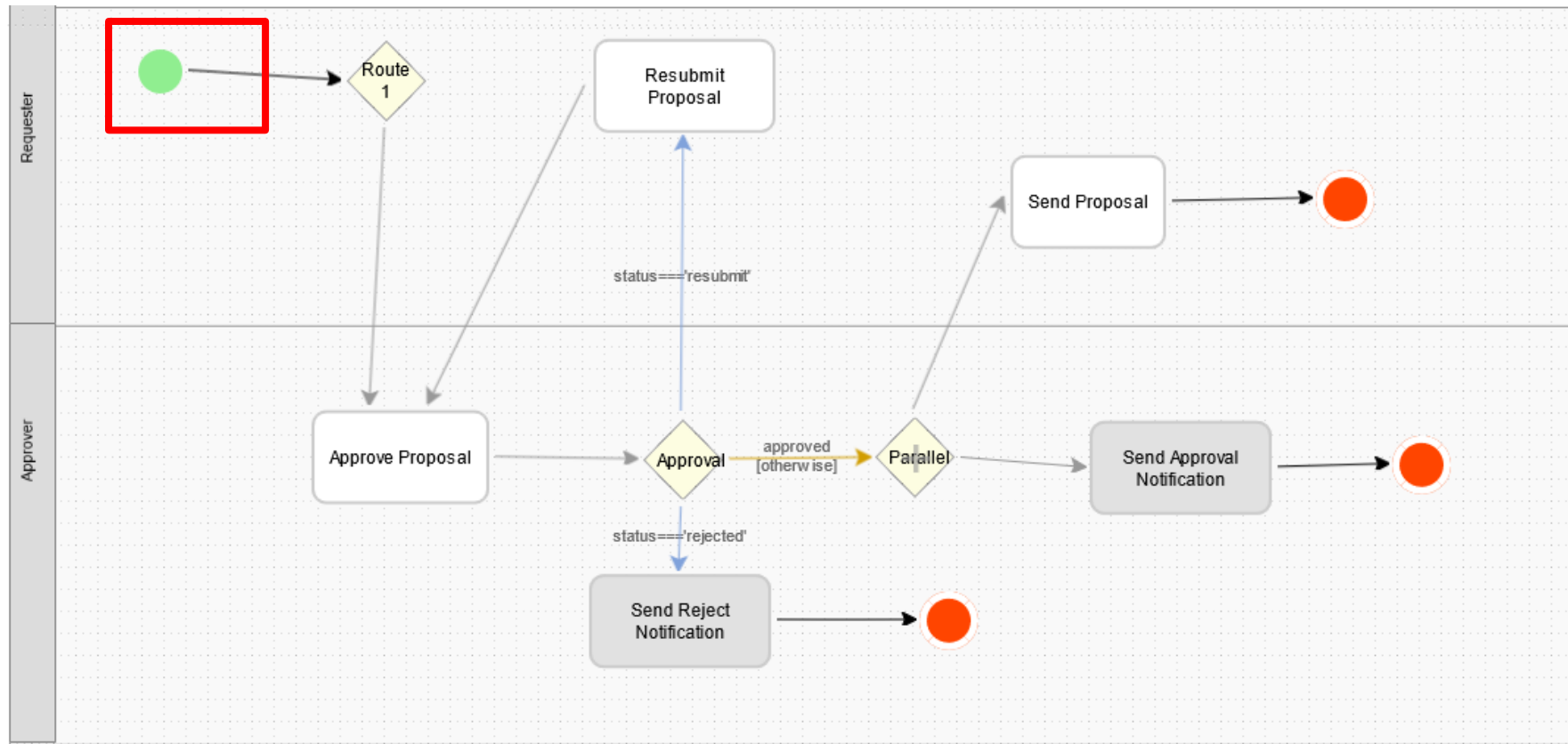
A short horizontal bar with a black left half and an orange right half.

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 RAD Service interface for configuring a process. On the left sidebar, the 'Processes' tab is selected. The main area shows the 'Purchase Request Process' workflow diagram. Below the diagram, the 'Map Activities to Forms' tab is active, displaying a list of activities. The 'Submit Request' activity is highlighted with a red box, and the 'Add/Edit Form' button next to it is also highlighted. The interface includes a top navigation bar with buttons for 'Design Process', 'Configure Mapping', 'Update via Saved XPDL', and 'Run Process'. The 'Process name' field is set to 'Purchase Request Process'. The bottom section shows a list of activities with the 'Submit Request' activity selected. A checkbox labeled 'Show The Next Assignment When Completed' is visible at the bottom.

Design App: purchaseRequestion
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

Submit Request
(0: submitRequest)

Show The Next Assignment When Completed

Add/Edit Form

Add/Edit Form

- Click on “Create New Form”

The screenshot shows the RAD Service interface. On the left is a sidebar with navigation options: Design App: purchaseRequisition (Version 1, Published), Forms & UI, Processes (selected), Properties & Export, Logs, and All Apps. The main area is titled 'MAP ACTIVITIES TO FORMS - SUBMIT REQUEST (SUBMITREQUEST)'. It has two tabs: 'Map to a Form' (selected) and 'External Form'. Below the tabs is a search bar. A table with the following headers is displayed: FORM NAME, TABLE NAME, DATE CREATED, DATE MODIFIED. The table is empty, and a large 'No' message is shown on the right. At the bottom, there is a pagination bar showing '15' items per page, 'Page 1 of 1', and 'No items'. A red box highlights the 'Create New Form' button at the bottom left of the main area.

Design App: purchaseRequisition
Version 1 Published

Forms & UI
Processes
Properties & Export
Logs
All Apps

MAP ACTIVITIES TO FORMS - SUBMIT REQUEST (SUBMITREQUEST)

Map to a Form External Form

Search

| FORM NAME | TABLE NAME | DATE CREATED | DATE MODIFIED |
|-----------|------------|--------------|---------------|
|-----------|------------|--------------|---------------|

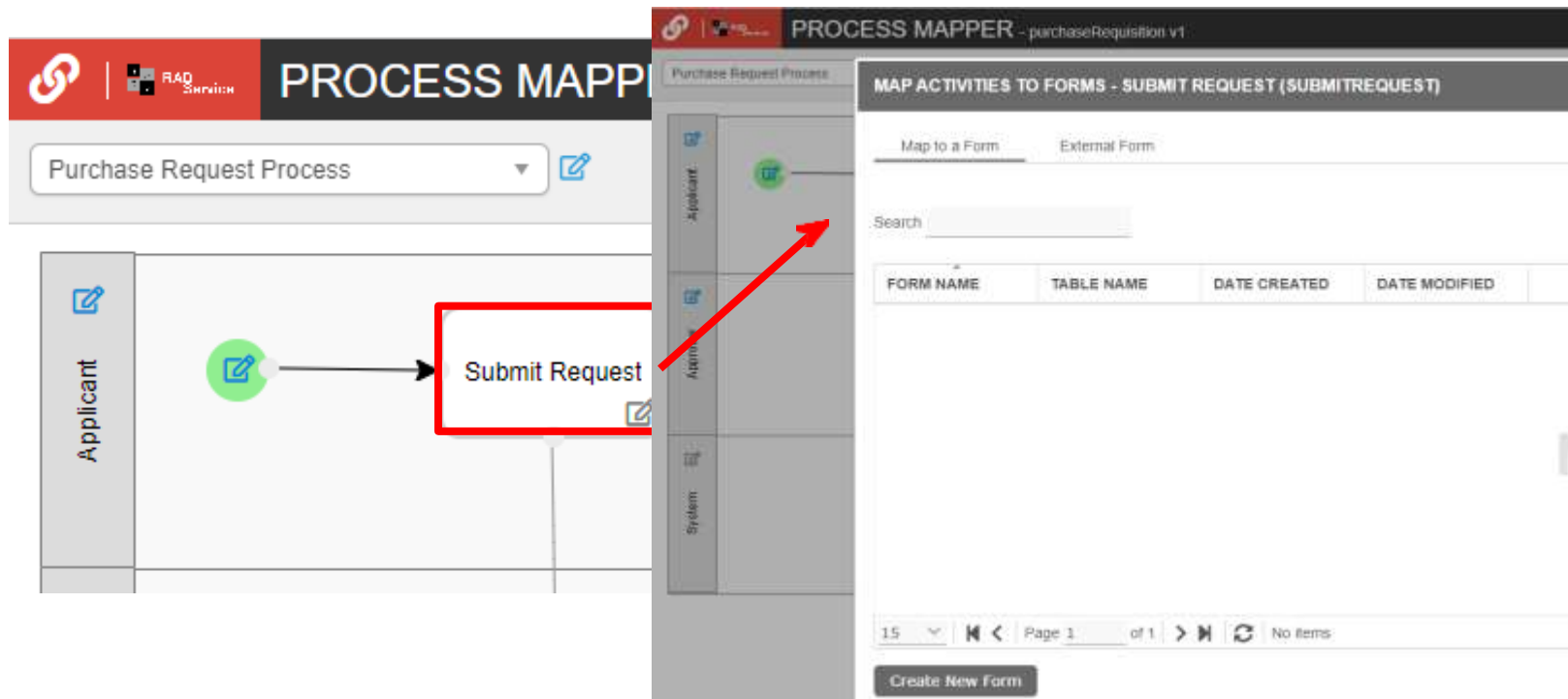
No

15 Page 1 of 1 No items

Create New Form

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.



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

The screenshot shows the 'CREATE NEW FORM' interface. On the left, a sidebar contains 'Design App: purchaseRequisition' with 'Version 1' and 'Published' buttons, and 'All Apps' below. The main form has a title bar 'CREATE NEW FORM' and a 'FORM DETAILS' section. The 'Form ID' field contains 'submitRequest' with a callout box stating 'a-z, A-Z, 0-9, _'. The 'Form Name' field contains '1 - Submit Request'. The 'Table Name' field contains 'app_fd_ purchase_requests' with a callout box stating 'Table name with prefix'. The 'Description' field is empty. At the bottom, there is a 'Show Advanced Options' link and 'Save' and 'Cancel' buttons, with the 'Save' button highlighted by a red rectangle. A callout box on the left states 'Numbered form name according to process flow' pointing to the 'Form Name' field.

Design App: purchaseRequisition
Version 1 Published

Logs
All Apps

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

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

Table name with prefix

Show Advanced Options

Save Cancel

App Prefix for Table Name

- By default, **ALL form data** will be stored into the **same database** that RADS 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 displays the RAD Service Form Builder interface. The top navigation bar includes a 'FORM BUILDER' title and a breadcrumb path: 'purchaseRequisition v1: 1 - Submit Request (Published)'. Below this, there are four tabs: 'DESIGN FORM', 'PROPERTIES', 'PREVIEW', and 'SAVE'. The 'DESIGN FORM' tab is active, showing a 'Basic' section with various form controls like Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, File Upload, and Subform. A 'Section' placeholder is visible in the design area, with a red box highlighting its top-right corner where a context menu is open. The 'Edit Section' dialog is displayed, showing the 'Edit Section > Data Binder > Advanced Options' path. The 'ID' field is set to 'section1', and the 'Label' field is set to 'Section'. A blue callout bubble points to the 'Label' field with the text 'Purchase Request Details'.

FORM BUILDER purchaseRequisition v1: 1 - Submit Request (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

Section

Omni Fields Here

Edit Section ?

Edit Section > Data Binder > Advanced Options

ID * section1

Label Section

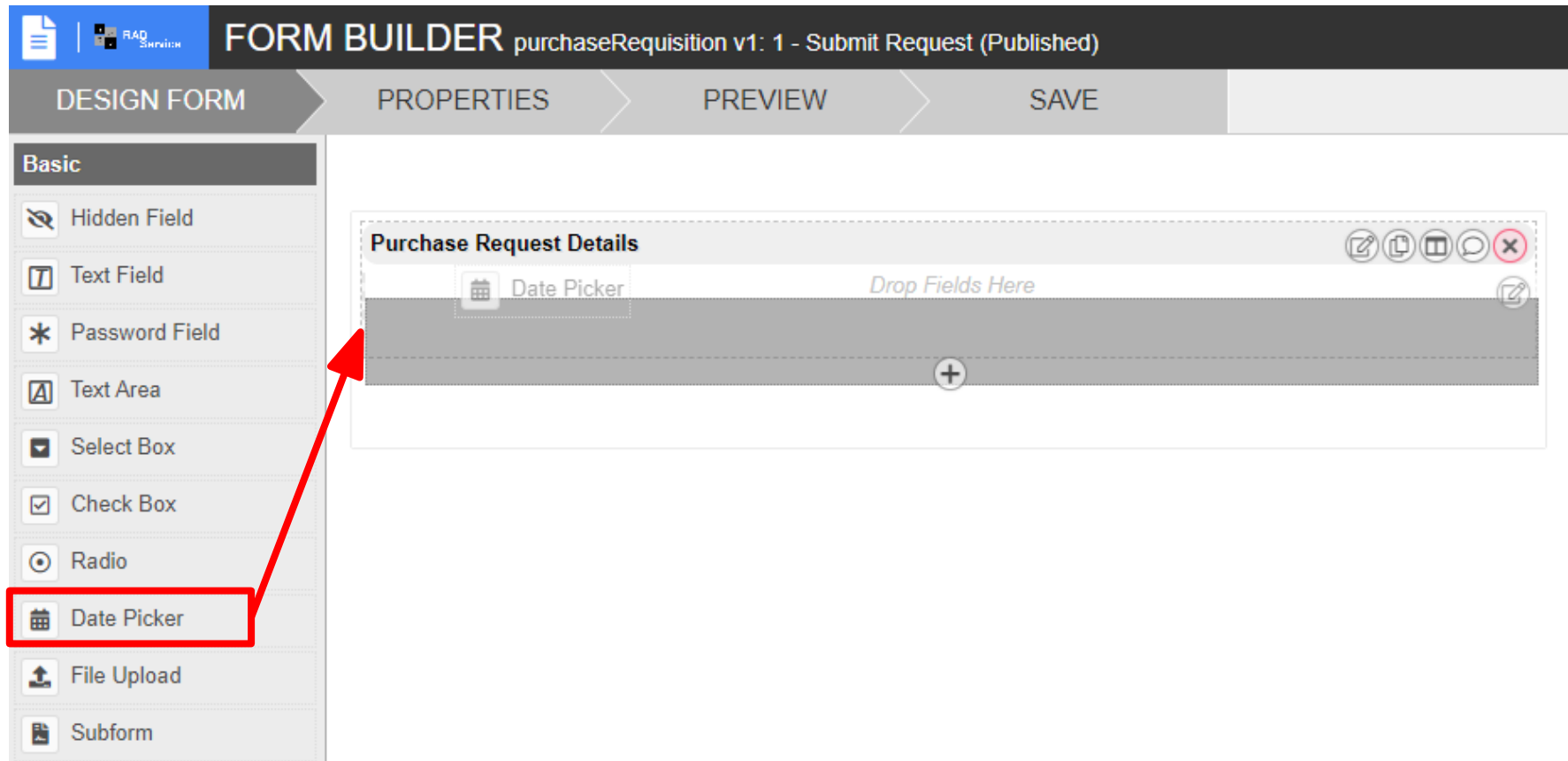
Readonly ☐

Display Field as Label When Readonly?

Purchase Request Details

Insert Date Picker

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



Edit The First Date Picker

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

FORM BUILDER purchaseRequisition v1: 1 - Submit Request (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

Purchase Request Details

Date Pi

Auto save when close? ☐ x

Edit Date Picker ?

Edit Date Picker > Advanced Options

ID * request_date

Label * Request Date

Allow Manual Input ☐

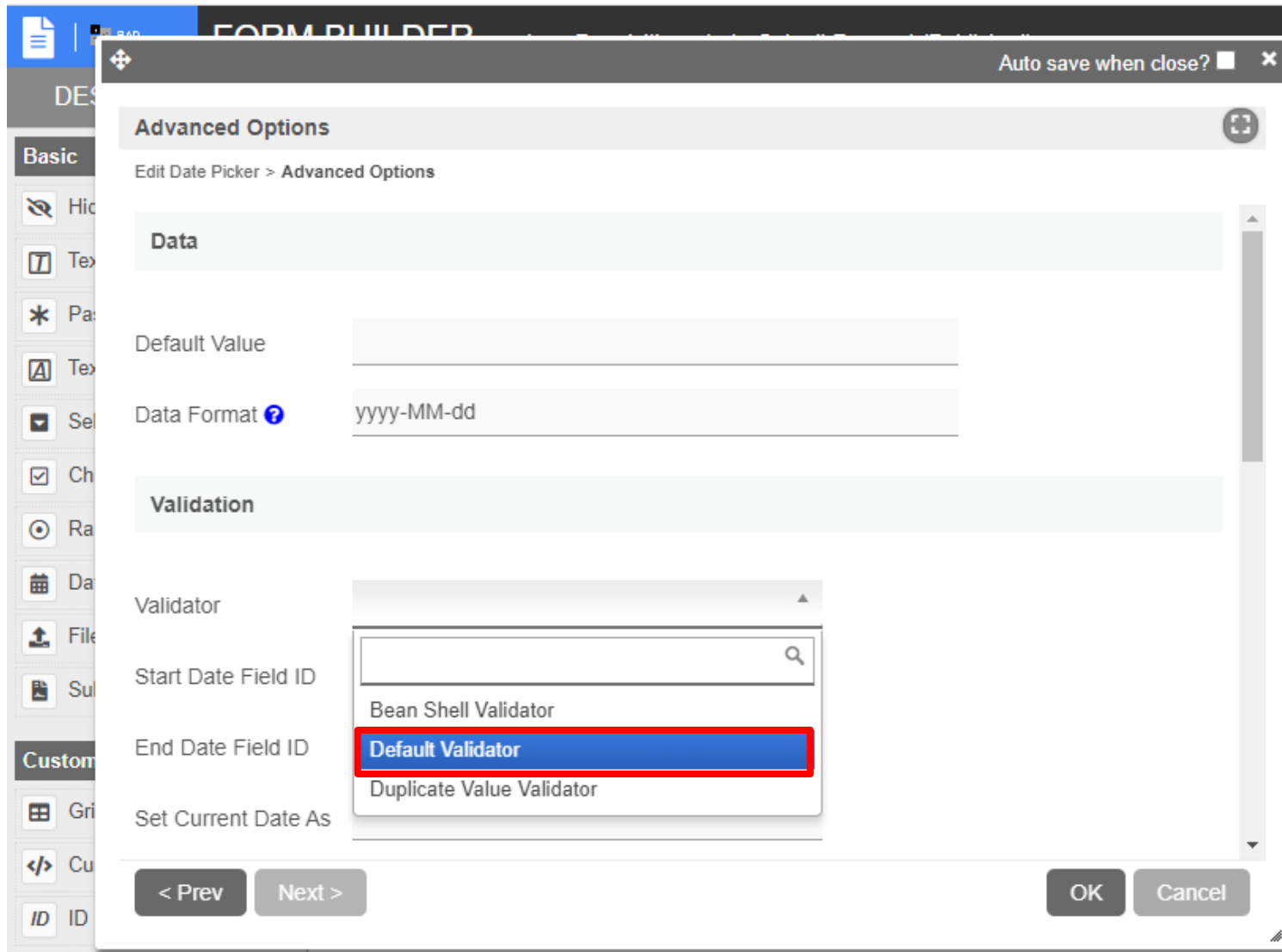
Field Type Date only

< Prev Next > OK Cancel

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

Be cautious about the case used, be consistent!

Set The Date Picker as Mandatory Field



Set The Date Picker as Mandatory Field

The screenshot shows the 'FORM BUILDER' application window for 'purchaseRequisition v1: 1 - Submit Request (Published)'. The 'DESIGN FORM' tab is active. On the left, a 'Basic' sidebar lists various field types: Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, and File Upload. The 'Date Picker' field is 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' text area is empty. The dialog has '< Prev', 'Next >', 'OK', and 'Cancel' buttons at the bottom.

FORM BUILDER purchaseRequisition v1: 1 - Submit Request (Published)

DESIGN FORM | PROPERTIES | PREVIEW | SAVE

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 ☒

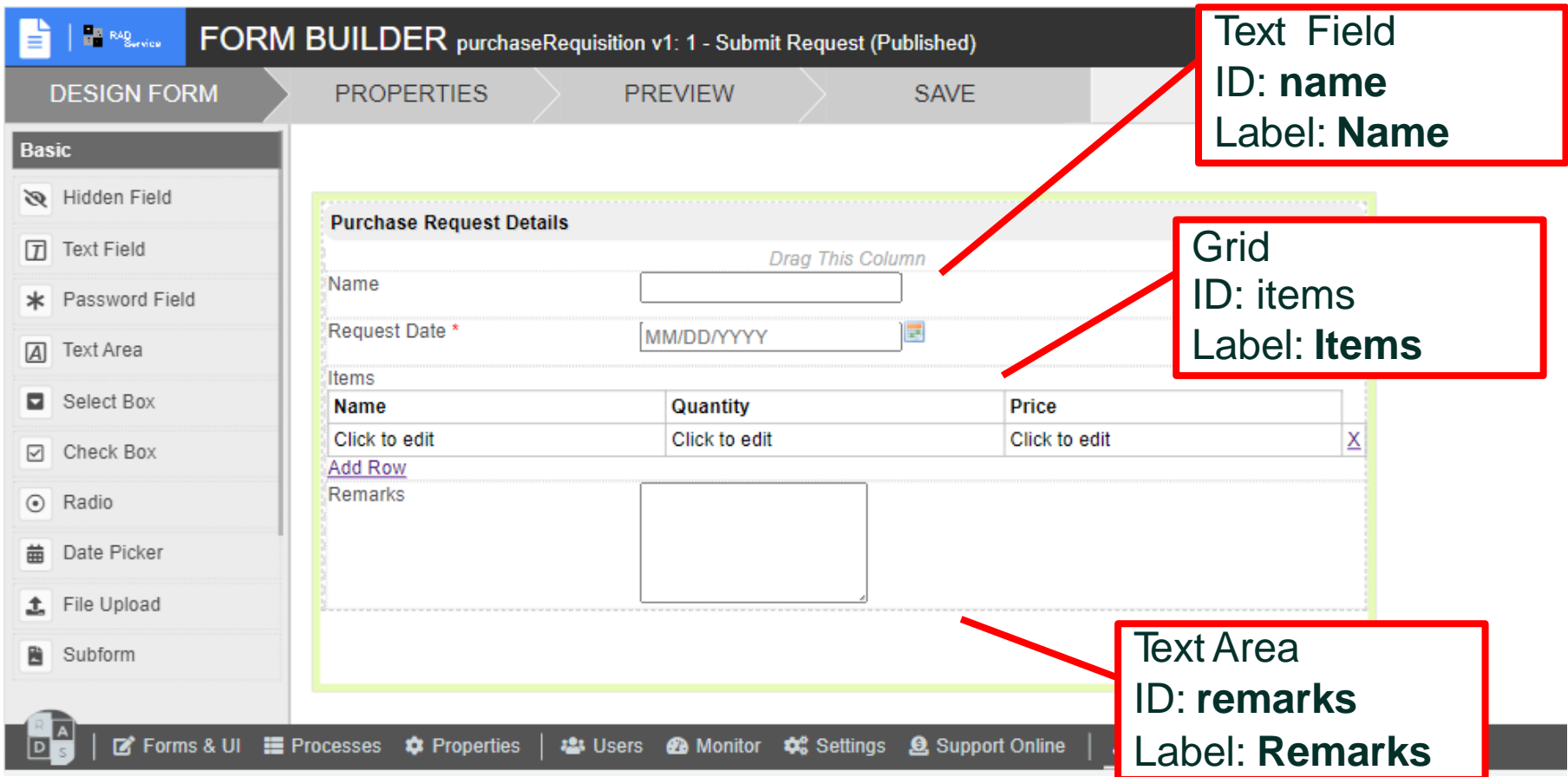
Type

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 purchaseRequisition v1: 1 - Submit Request (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

Purchase Request Details

Name

Request Date *

Items

| 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

Complete the Design of “Submit Request” Form

- Preview and save.

The screenshot displays the RAD Service Form Builder interface. The top navigation bar includes tabs for DESIGN FORM, PROPERTIES, PREVIEW, and SAVE. The PREVIEW tab is currently selected, and a red box with the text "Preview and Save" is overlaid on it. The form being previewed is titled "Purchase Request Details" and contains the following fields:

- Name: A text input field.
- Request Date *: A date picker field showing MM/DD/YYYY.
- Items: A table with three columns: Name, Quantity, and Price. The table has one row with "Click to edit" in each cell and an "Add Row" link below it.
- Remarks: A text area.

The bottom status bar shows various icons and text: Forms & UI, Processes, Properties, Users, Monitor, Settings, Support Online, and Quick Edit : On.

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

The mockup displays the 'PURCHASE REQUEST PROCESS - SUBMIT REQUEST' interface. It features a sidebar with navigation options: Home, Purchase Requisition, Submit New Request, and Inbox. The main content area shows the form title and a 'Request Details' section. This section contains input fields for 'Name' and 'Request Date', and a table for 'Items' with columns for Name, Quantity, and Price. Each table row has a 'Click to edit' link. Below the table is an 'Add Row' link. A 'Remarks' text area is located below the table. At the bottom of the form, there are two 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? ☐ X

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" value=""/> |
| <input type="text" value="medicalEquipment"/> | <input type="text" value="Medical Equipment"/> | <input type="text" value=""/> |
| <input type="text" value="pantrySupplies"/> | <input type="text" value="Pantry Supplies"/> | <input type="text" value=""/> |

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>
```

The screenshot shows a 'Request Details' form. It includes fields for Name, Request Date (with a calendar icon), and Category (a dropdown menu set to 'Stationery'). Below these is a table with columns for Name, Quantity, and Price, each with a 'Click to edit' link. An 'Add Row' link is also present. At the bottom, there is a 'Remarks' text area and a 'Custom HTML' section. The 'Custom HTML' section is highlighted with a red border and contains the text 'CUSTOM HTML'.

| Name | Quantity | Price |
|---------------|---------------|---------------|
| Click to edit | Click to edit | Click to edit |

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.

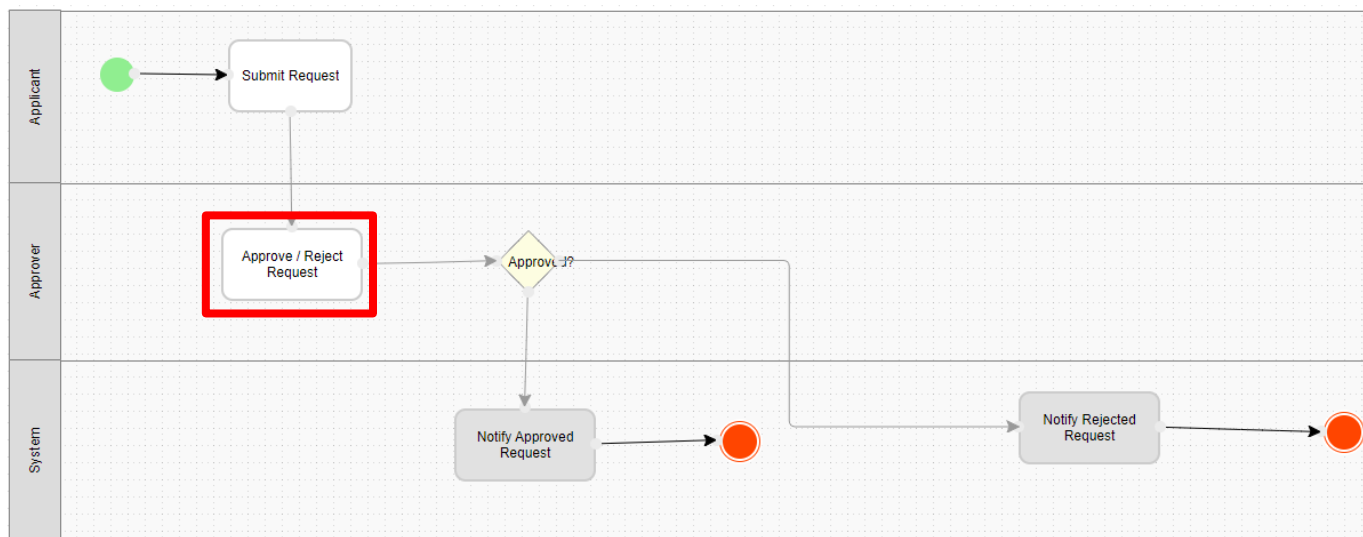


Chapter 3

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: approvalRequest

Form Name: 2 - Approve / Reject Request

Table Name: app_id_ purchase_requests

Description:

[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

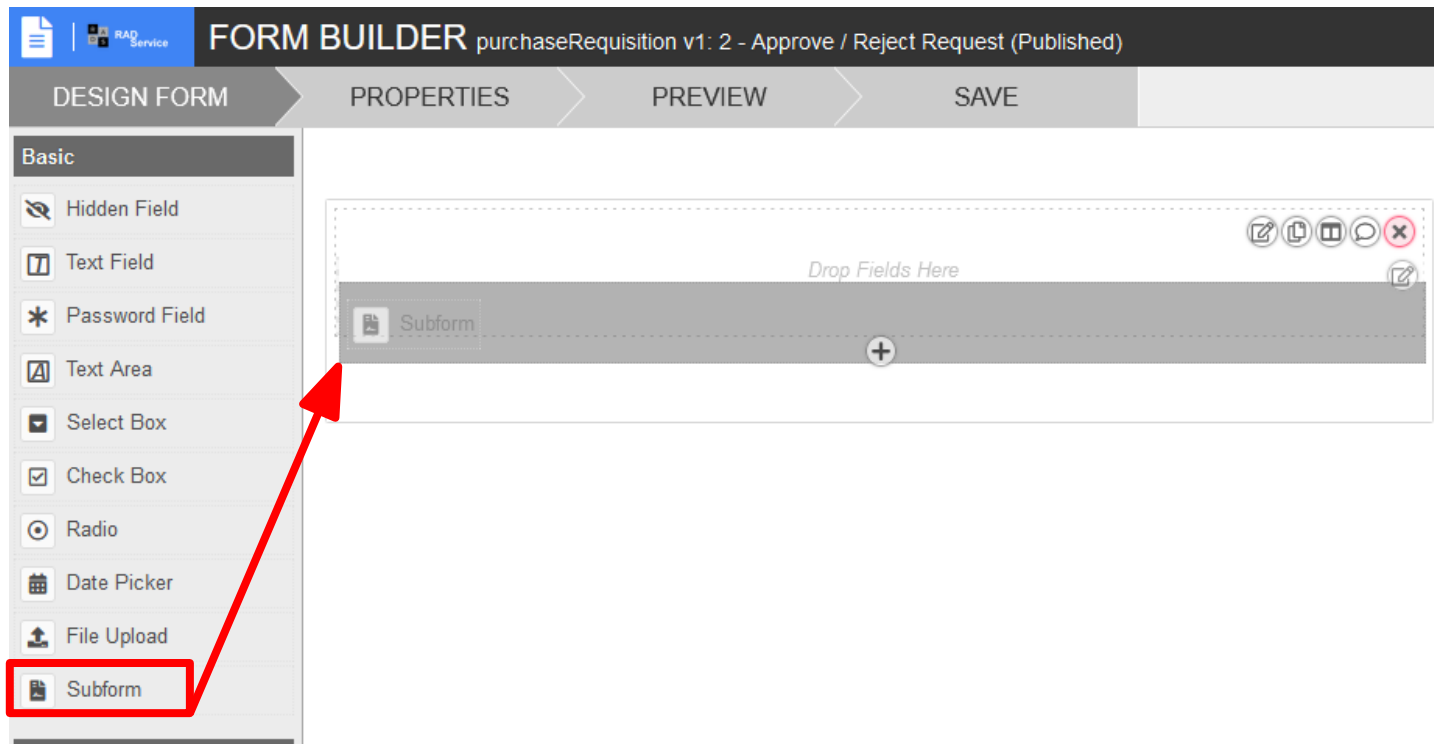
Approve / Reject Request Form

The screenshot displays a form titled "Approve / Reject Request Form" which contains a subform titled "Request Details". The subform includes the following fields:

- Name**: A text input field with a "Drag This Column" hint.
- Request Date ***: A date input field with a calendar icon and the format "MM/DD/YYYY".
- Category**: A dropdown menu currently showing "Stationery".
- Items**: A table with three columns: "Name", "Quantity", and "Price". Each cell contains the text "Click to edit". There is an "Add Row" link and a close button (X) in the top right corner of the table.
- Remarks**: A large text area for additional comments.
- Custom HTML**: A section at the bottom with a "CUSTOM HTML" label.

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

purchaseRequisition v1: 2 - Approve / Reject Request (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

Request Approval

Drag This Column

Purchase Request Details

Name

Request Date *

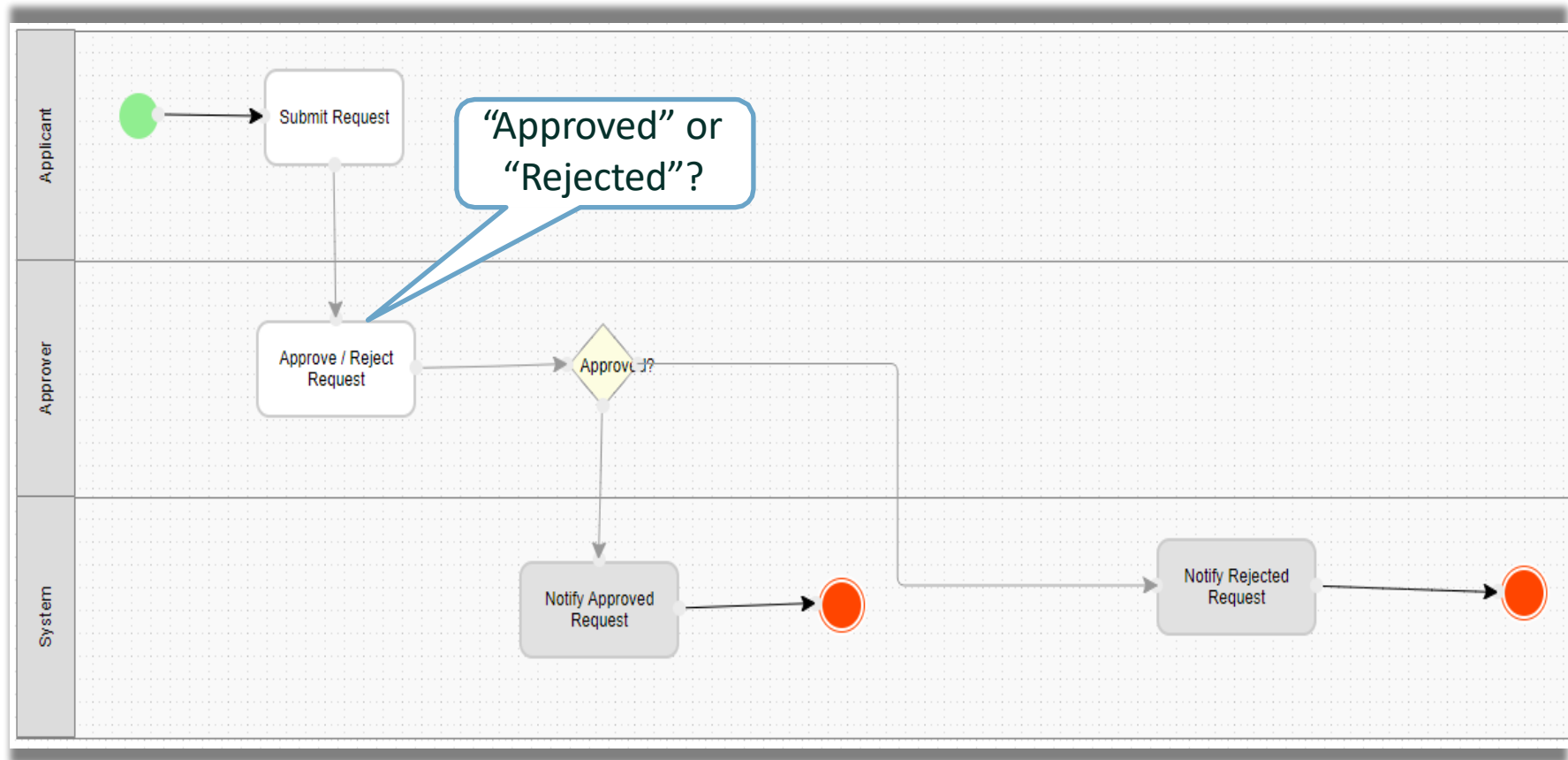
MM/DD/YYYY

Category

Items

| Name | Quantity | Price |
|---------|----------|-------|
| Remarks | | |

What About the Approval Status?



Insert Radio Button

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

Auto save when close? ☐ X

Edit Radio ?

Edit Radio > Advanced Options

ID * approval

Label Approval

Options (Hardcoded)

| VALUE | LABEL | GROUPING |
|----------|----------|----------|
| Approved | Approved | |
| Rejected | Rejected | |

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 web form titled "Advanced Options" with a breadcrumb "Edit Radio > Advanced Options". The form is divided into two sections: "UI" and "Workflow". In the "UI" section, there are two checkboxes: "Readonly" and "Display Field as Label When Readonly?", both of which are currently unchecked. The "Workflow" section contains a single text input field labeled "Workflow Variable" which has the value "status" entered. This input field is highlighted with a red rectangular border. At the bottom of the form, there are navigation buttons "< Prev" and "Next >" on the left, and "OK" and "Cancel" buttons on the right.

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

status Equal To Approved

THEN

Transition transition4 (Notify Approved Request)

Add Rule

ELSE THEN

Transition transition5 (Notify Rejected Request)

Transition

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 *

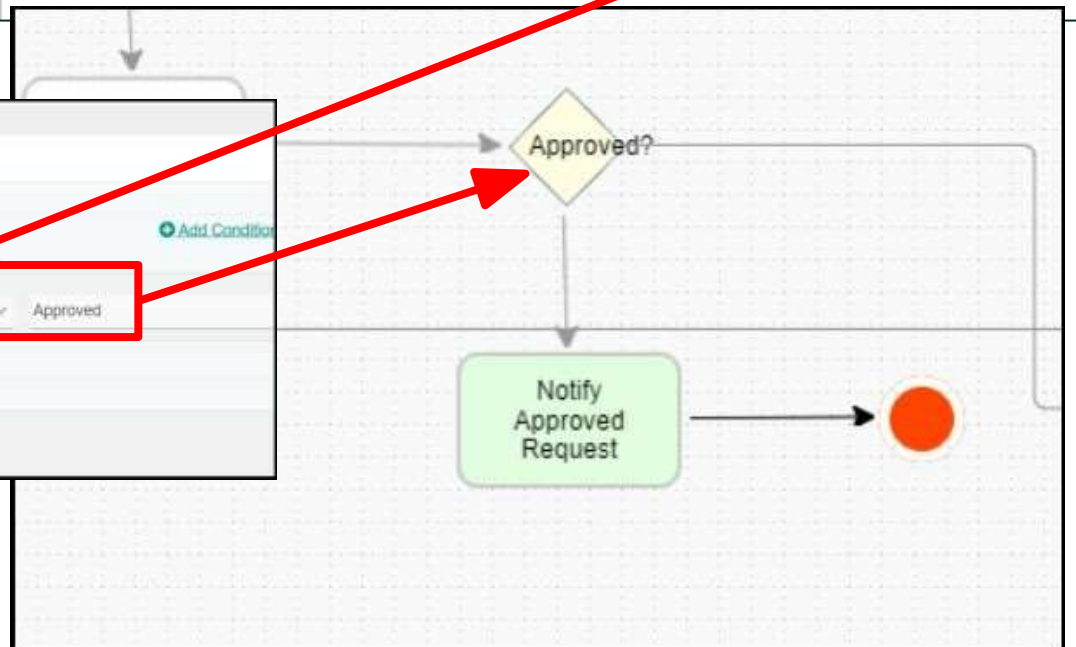
IF

And


Equal To

THEN

Transition



Request Approval Form

 **FORM BUILDER** purchaseRequisition v1: 2 - Approve / Reject Request (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

Request Approval
Drag This Column

Purchase Request Details

Name

Request Date *

Category

Items

| Name | Quantity | Price |
|---------|-------------|-------|
| Remarks | <div></div> | |

Approval ☐ Approved ☐ Rejected

Continue With the Assignment

SELAMAT DATANG TO PURCHASE REQUISITION PORTAL

Admin Admin
admin

Welcome

Purchase Requisition

Submit New Request

Inbox

Home > Purchase Requisition > Inbox

PURCHASE REQUEST PROCESS - APPROVE / REJECT REQUEST

Request Approval

Purchase Request Details

Name: John Smith

Request Date: * 10/24/2022

Category: Stationery

Items

| Name | Quantity | Price |
|--------|----------|-------|
| Pencil | 5 | 10 |

Remarks

Approval ☒ Approved ☐ Rejected

Save As Draft **Complete** Cancel

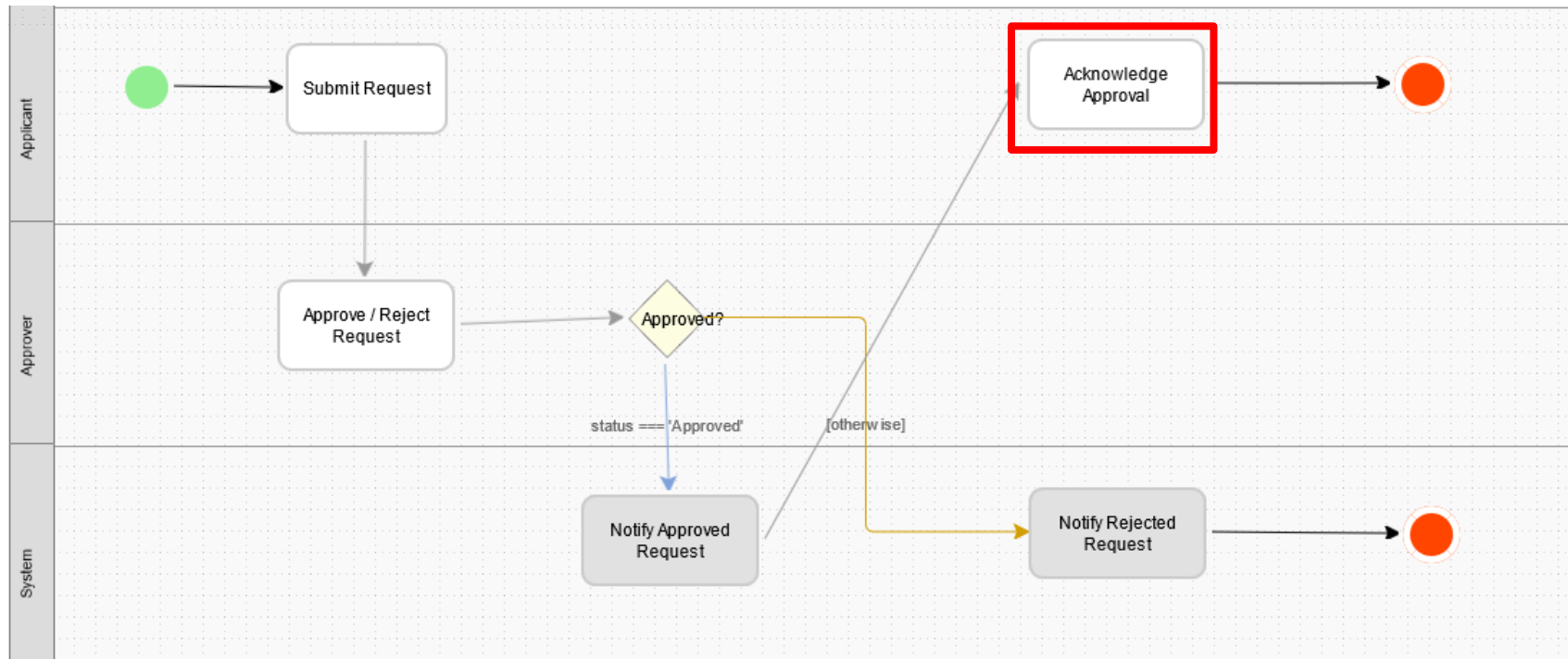
Exercise

- 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

FORM BUILDER purchaseRequisition v1: 3-Acknowledge Approved (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 ID Generator Field

Approval Acknowledgement

Drag This Column

Request Approval

Purchase Request Details

Name

Request Date *

MM/DD/YYYY

Category

Items

| Name | Quantity | Price |
|---------|----------|-------|
| 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@oget.org

Welcome

Purchase Requisition

Submit New Request

Inbox

Home > Purchase Requisition > Inbox

PURCHASE REQUEST PROCESS - ACKNOWLEDGE APPROVAL

Approval Acknowledgement

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

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 RADS

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