RAD
Service

RADS

Improving Your Form Design & Presentation

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

Last Revised on Oct 2021

Prerequisites

1. Good understanding on the basic functionality of the Form Builder.

Content

Chapter 1

Introduction

Introduction

- Learning about more RADS plugins/elements to improve your form design and presentation.
- In this module, we will be covering the following elements.

1. Grid (Form Element)
2. Form Grid (Form Element)
3. Multirow Form Binder (Form Binder)
4. List Grid (Form Element)
5. CRUD (Userview Menu)
6. Custom HTML (Form Element)

RAD Service

List of Available Elements

- Check out https://docs.rads.purwana.net/Form+Builder for the list of **Form** related elements (Form Element, Form Validator, Form Binder, Form Options Binder).
- Check out https://docs.rads.purwana.net/Userview+Builder for the list of **Userview** related elements.

RAD Service

Chapter 2

Grid

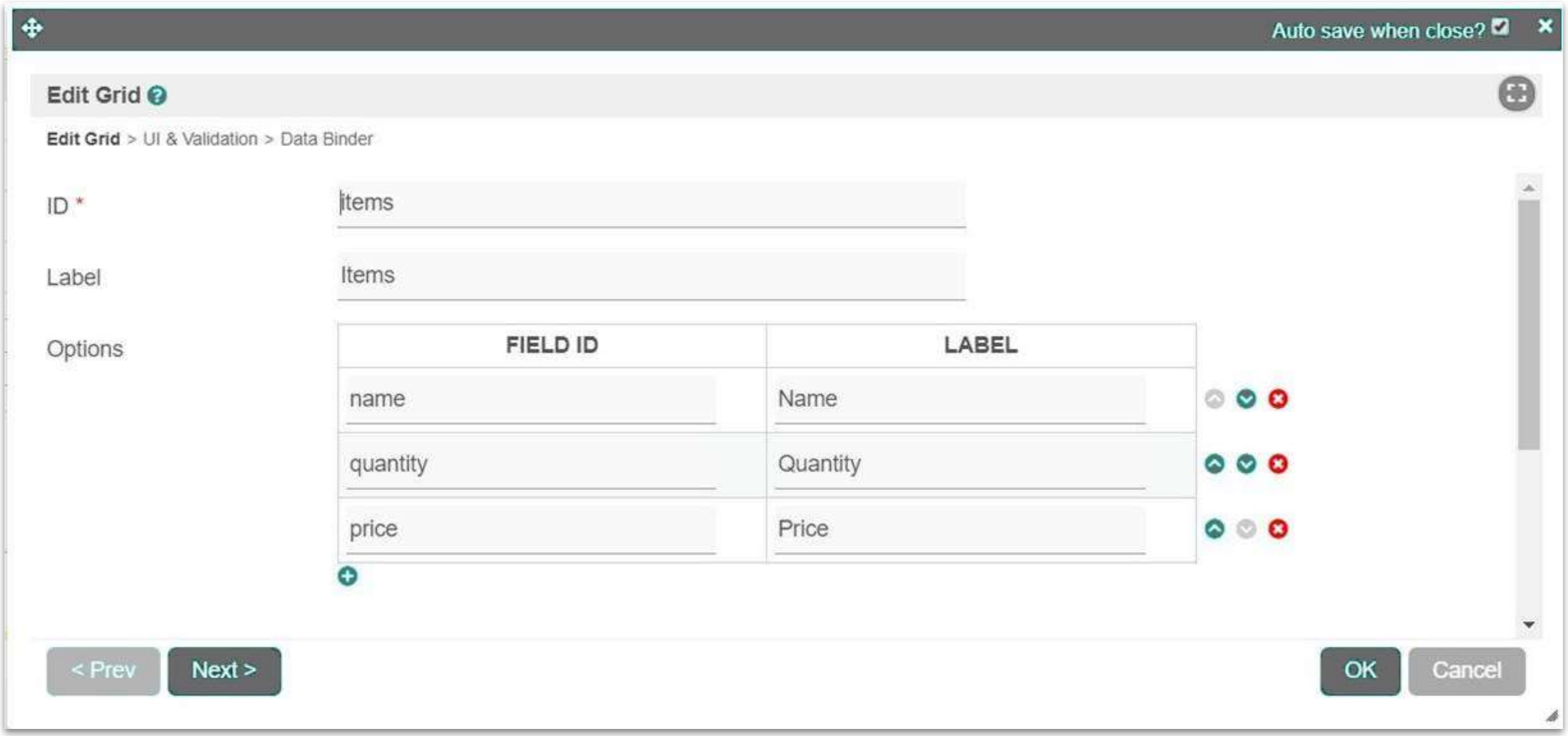
Grid

- Grid is the most basic element available in the Form Builder in capturing multi-row data.

- Reference: https://docs.rads.purwana.net/Grid
- For your information: The other grid-like element available in Form Builder is Form Grid.

Refresh

- Refresh your memory on what you did back in module 5 - Designing your first Form
- Take a look at your “items” field element



ID * items

Label Items

Options

FIELD ID	LABEL
name	Name
quantity	Quantity
price	Price

Just In Case...

- You can download the base Purchase Requisition app - **13.jwa** to start ‘playing around’ with it.
- Completed Form Definition for “1-Submit Request“ can be obtained from **13.2.1.txt**.

Chapter Review

- Able to use the Grid element.

RAD Service

Chapter 3

Form Grid

Form Grid

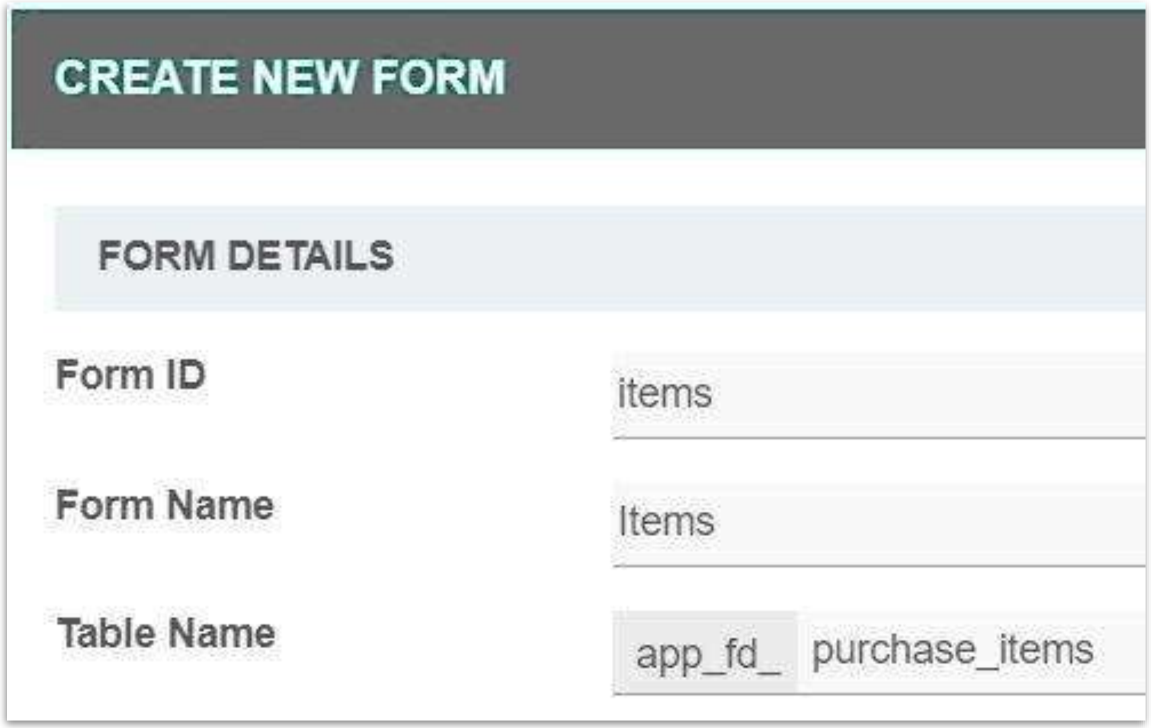
- Form Grid works similarly like the basic Grid.
- Instead of editing data row inline, editing is done on a full fledged Form that opens up in a dialog.
- Able to reuse validation and formatting from the selected Form.

- Reference: https://docs.rads.purwana.net/Form+Grid

RAD Service

Exercise

- Re-import the base app “**13.jwa**” into your copy of RADS, OR delete the “items” Grid element.
- Create a new Form with the following details.

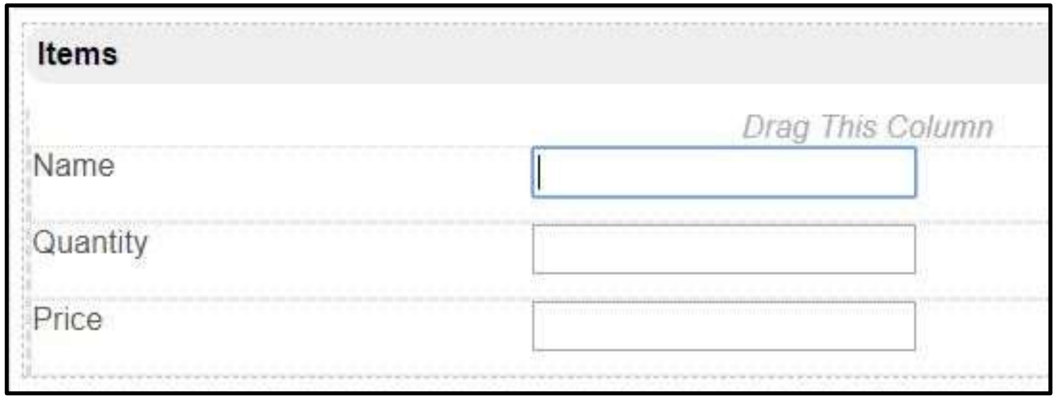

CREATE NEW FORM

FORM DETAILS

Form ID	items
Form Name	Items
Table Name	app_fd_ purchase_items

Exercise

- Add 3 text fields with the following details:-
  - ID: name, Label: Name
  - ID: quantity, Label: Quantity
  - ID: price, Label: Price


Items

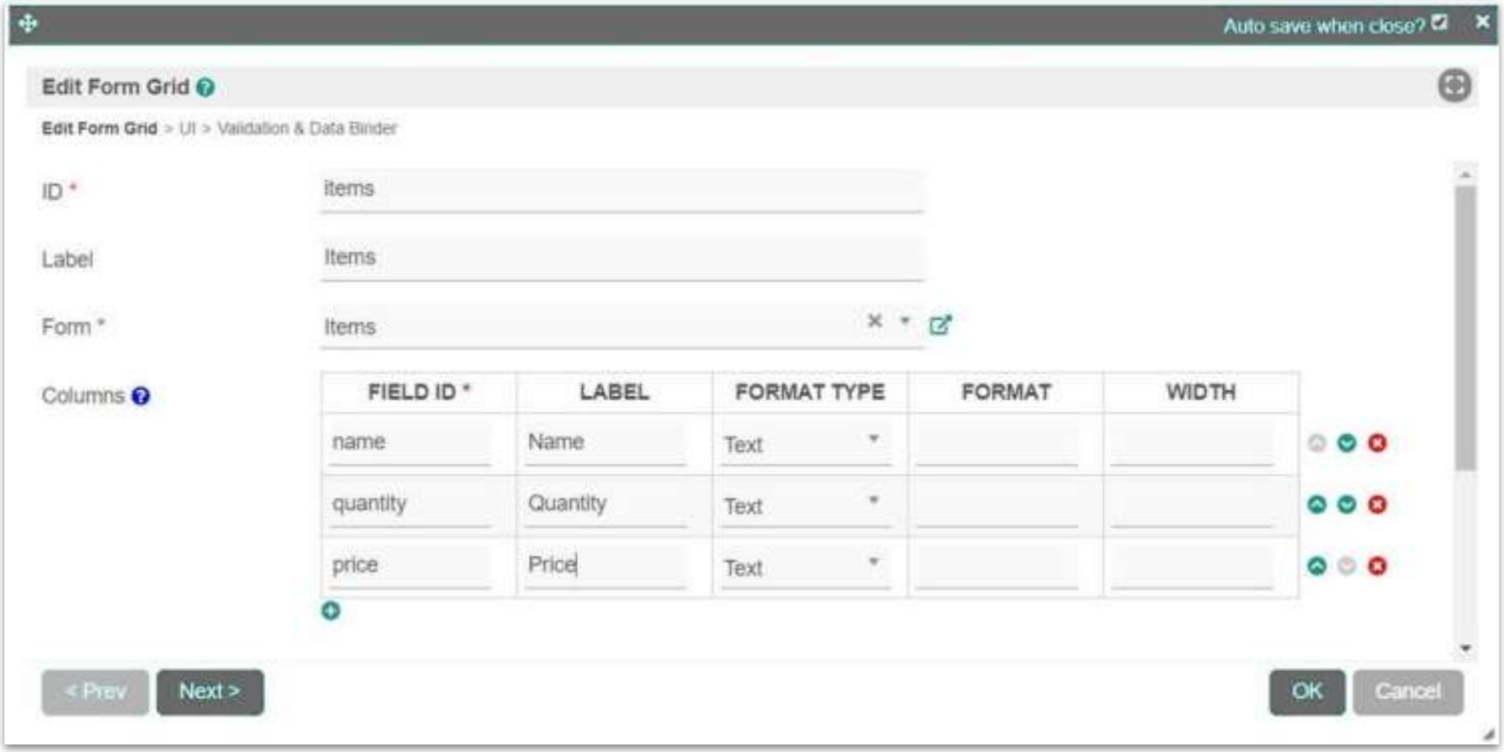
Drag This Column

Name	
Quantity	
Price	

- Save the form

Exercise

- Edit the “1-Submit Request” form.
- Add a “Form Grid” wherever relevant.
- Configure accordingly.



ID *	items
Label	Items
Form *	Items

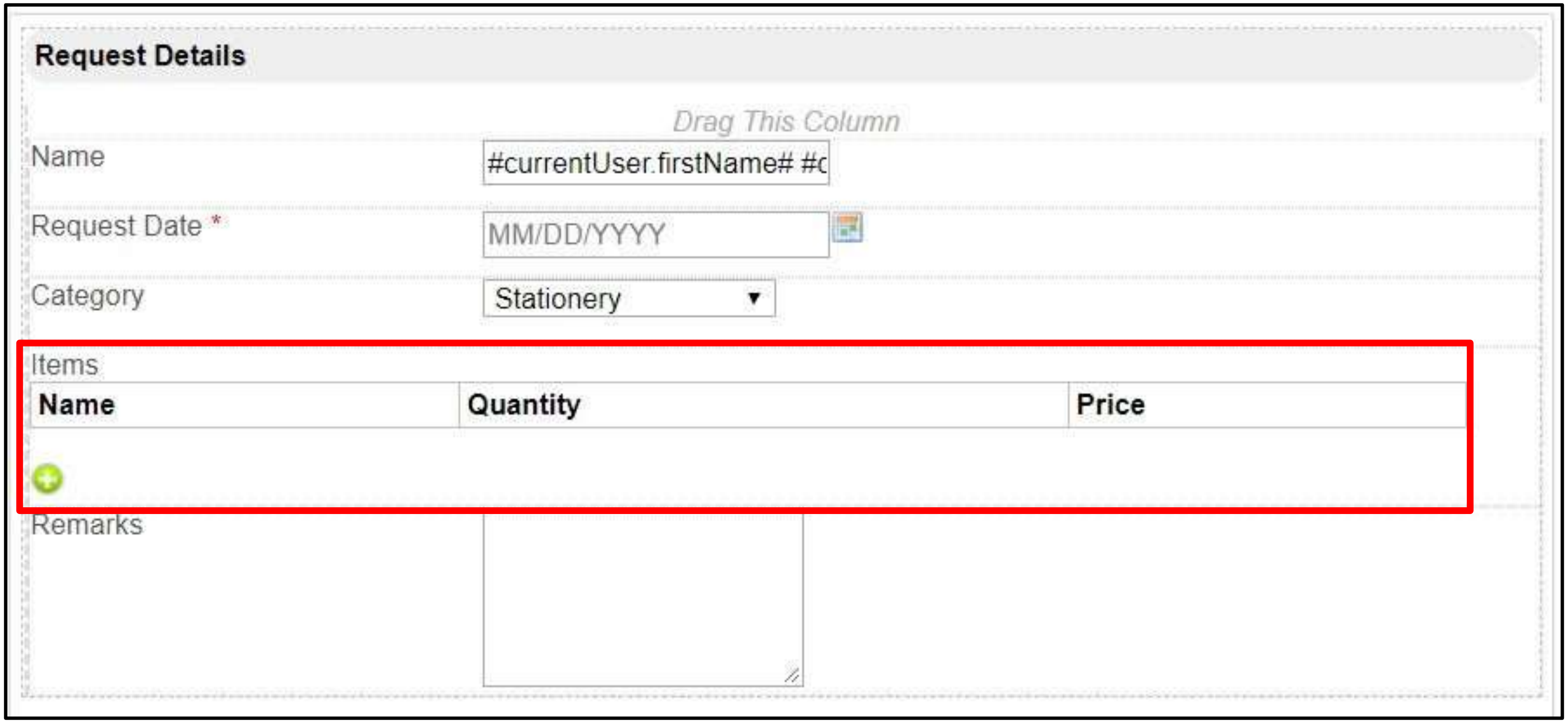
Columns

FIELD ID *	LABEL	FORMAT TYPE	FORMAT	WIDTH
name	Name	Text		
quantity	Quantity	Text		
price	Price	Text		

RAD Service

Exercise

- This is how your form design should look like.



Request Details

Drag This Column

Name	#currentUser.firstName# #c
Request Date *	MM/DD/YYYY
Category	Stationery ▼

Items

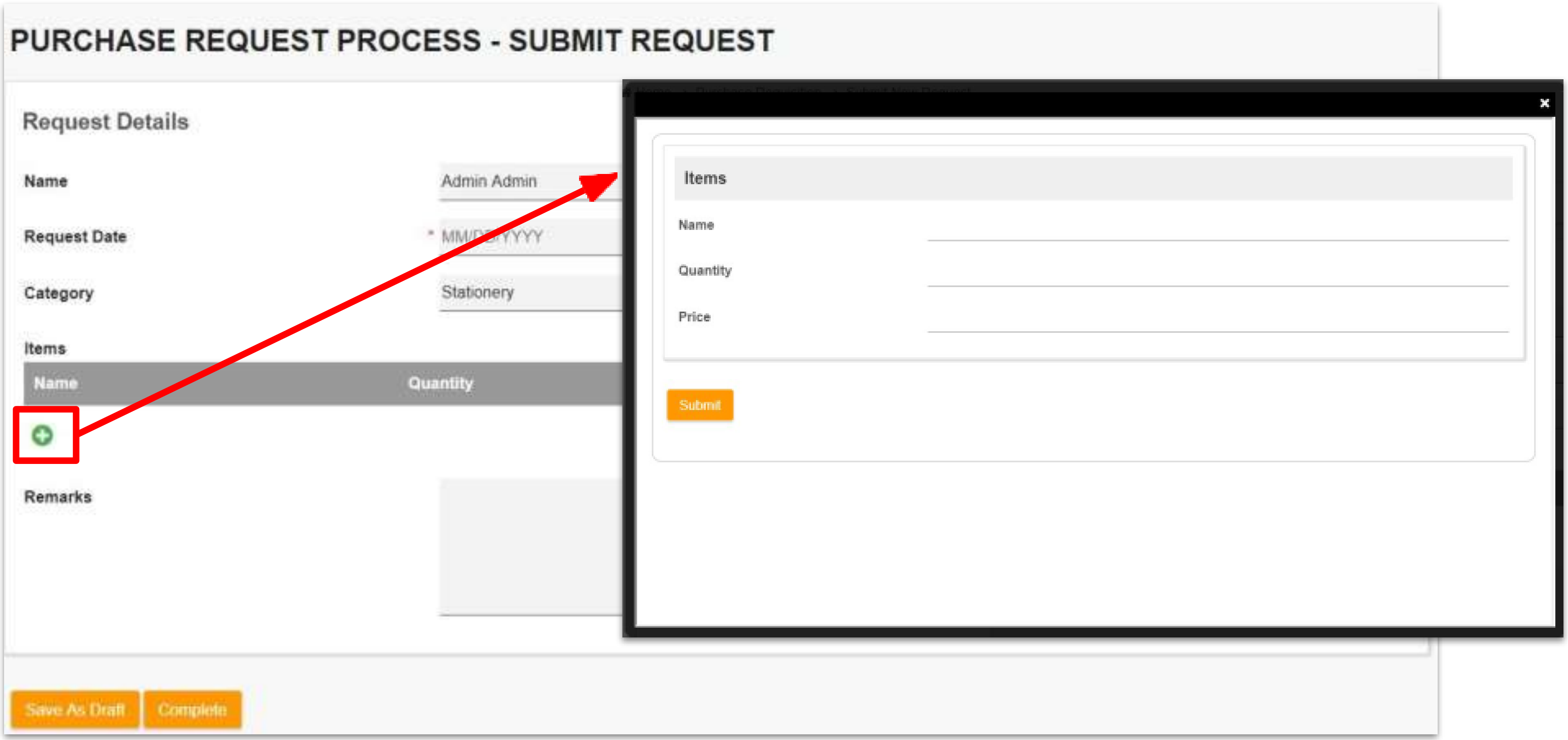
Name	Quantity	Price

Remarks

RAD Service

Exercise

- This is how the form should look like in runtime.



PURCHASE REQUEST PROCESS - SUBMIT REQUEST

Request Details

Name	Admin Admin
Request Date	* MM/DD/YYYY
Category	Stationery
Items	
Remarks	

Name	Quantity

Items

Name	
Quantity	
Price	

Exercise

- Run a new “Submit New Request” process, and submit the form to observe.

Materials

- "1-Submit Request" form definition can be obtained from **13.3.1**.txt
- “Items” form definition can be obtained from **13.3.2**.txt

Chapter Review

- Being able to use the Form Grid element.
- PS: Check out Advanced Grid (https://docs.rads.purwana.net/Advanced+Grid) Form Element that performs similarly as Form Grid.

RAD Service

Chapter 4

Multirow Form Binder

RAD Service

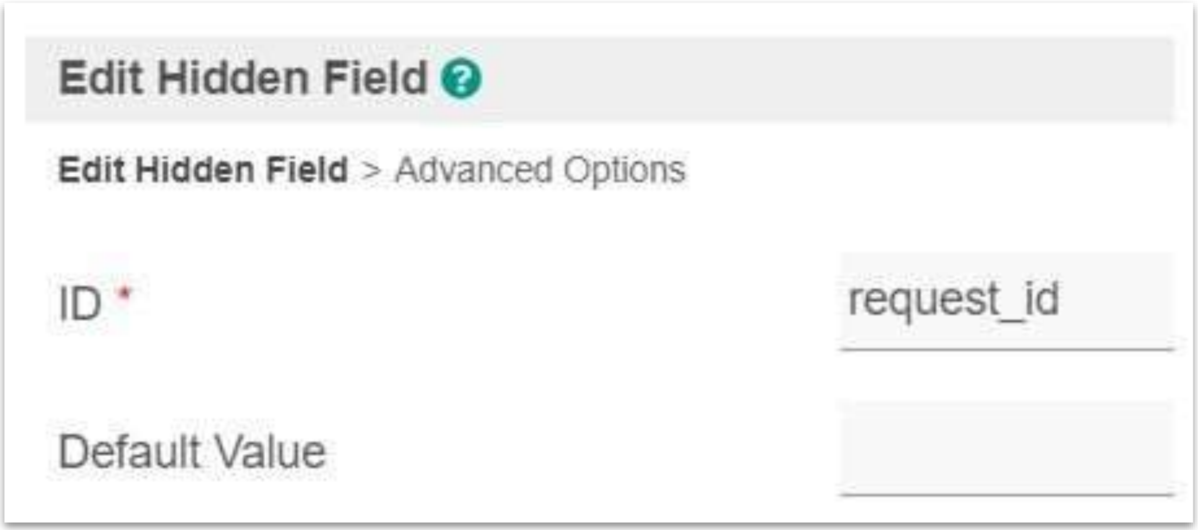
Multirow Form Binder

- Multirow Form Binder is a Store/Load Form Binder that is designed to treat multi-row data for grid form element.
- Rather than storing as traditional JSON data format in a single column cell, the Multirow Form Binder saves the data into its respective tables.
- This would make data retrieval easier for sorting, statistics, and indexing/performance purpose.

- Reference: https://docs.rads.purwana.net/Multirow+Form+Binder

Exercise

- Continue to use the application from the previous chapter OR import app from **13.4.1**.jwa.
- Edit the “Items” form.
- Add a Hidden Field to the form.
- Configure accordingly.



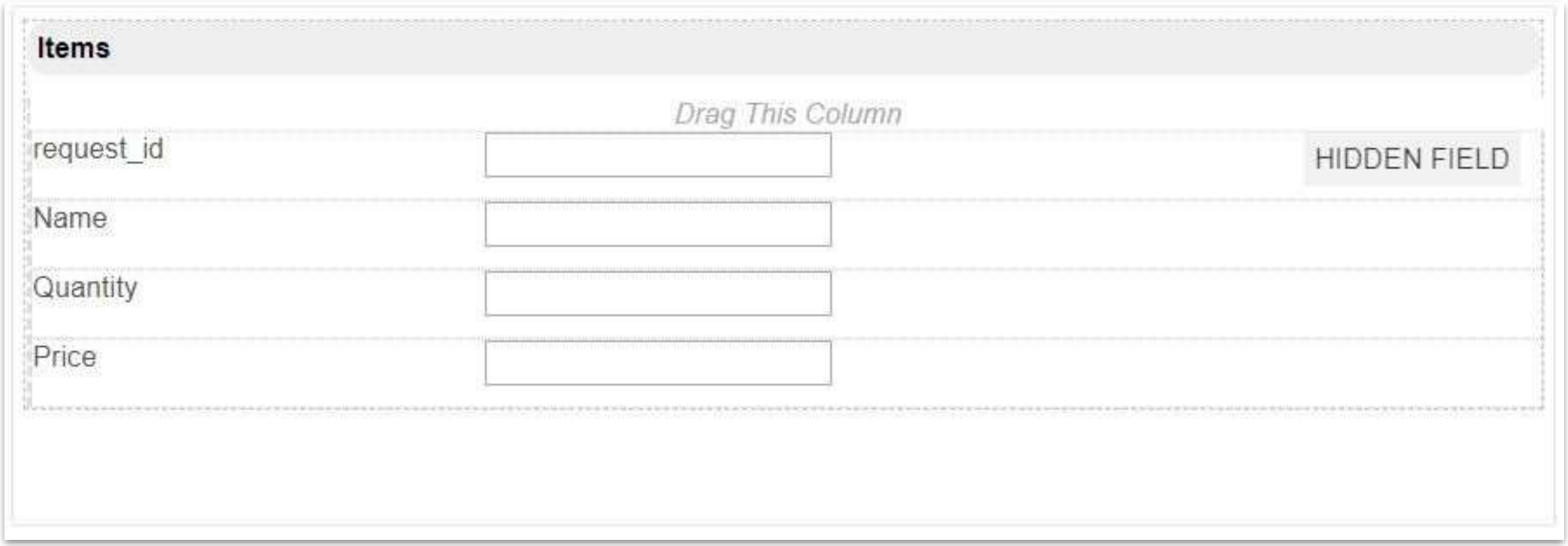
Edit Hidden Field

Edit Hidden Field > Advanced Options

ID *	request_id
Default Value	

Exercise

- This is how your “Items” form should look like.



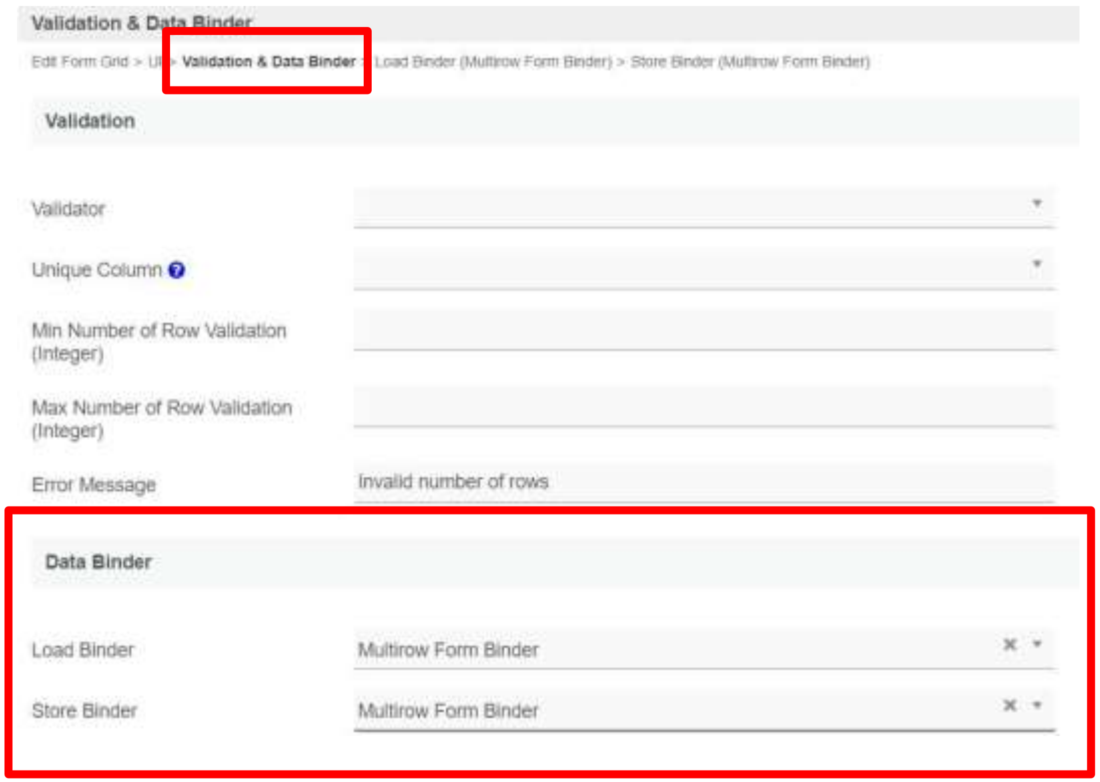
Items

Drag This Column

request_id		HIDDEN FIELD
Name		
Quantity		
Price		

Exercise

- Edit the "1-Submit Request" form.
- Configure the Form Grid element to utilize the Multirow Form Binder in Data Binder.



Validation & Data Binder

Edit Form Grid > UI > **Validation & Data Binder** > Load Binder (Multirow Form Binder) > Store Binder (Multirow Form Binder)

Validation

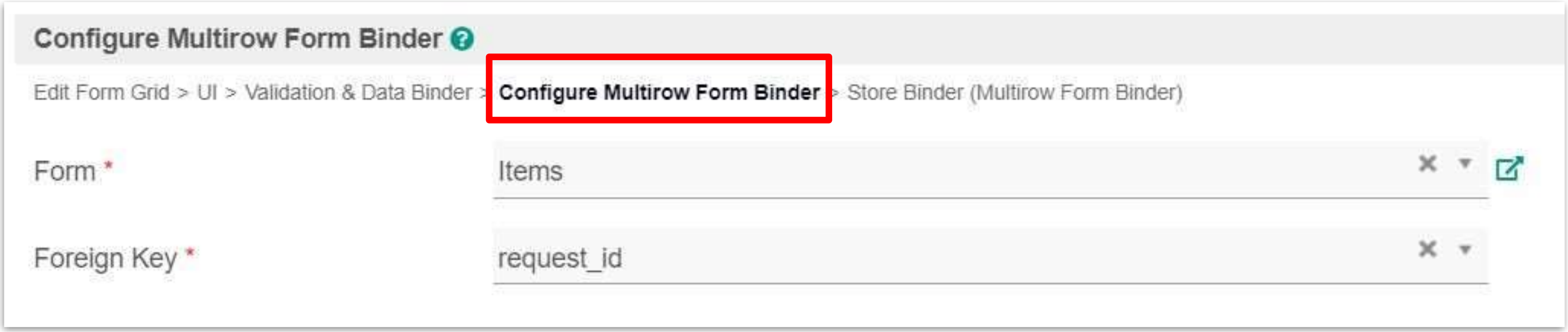
Validator	
Unique Column	
Min Number of Row Validation (Integer)	
Max Number of Row Validation (Integer)	
Error Message	Invalid number of rows

Data Binder

Load Binder	Multirow Form Binder
Store Binder	Multirow Form Binder

Exercise

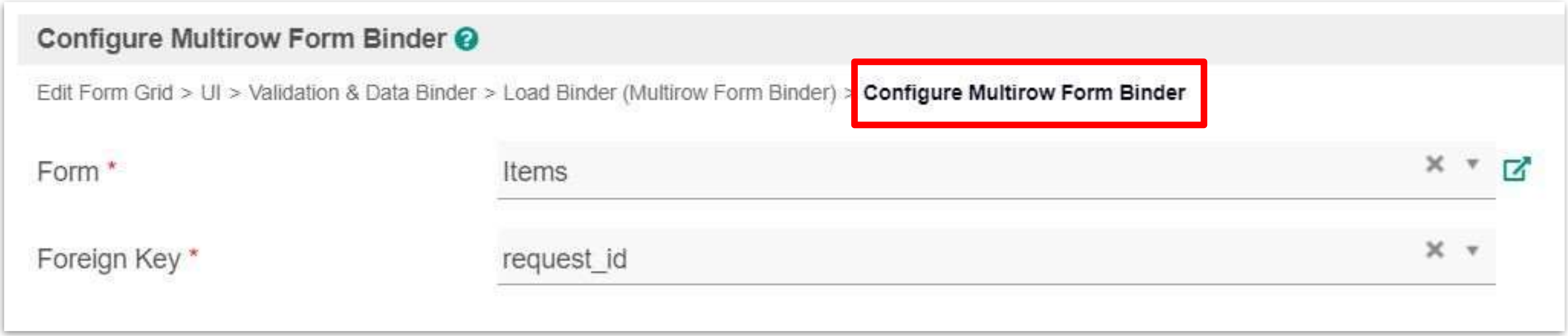
- Click next to configure the Binder.
- Configure accordingly.



Configure Multirow Form Binder

Edit Form Grid > UI > Validation & Data Binder > **Configure Multirow Form Binder** > Store Binder (Multirow Form Binder)

Form *	Items
Foreign Key *	request_id



Configure Multirow Form Binder

Edit Form Grid > UI > Validation & Data Binder > Load Binder (Multirow Form Binder) > **Configure Multirow Form Binder**

Form *	Items
Foreign Key *	request_id

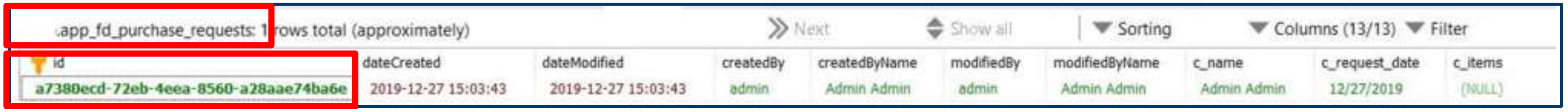
Exercise

- Run a new “Submit New Request” process, and submit the form to observe.

RAD Service

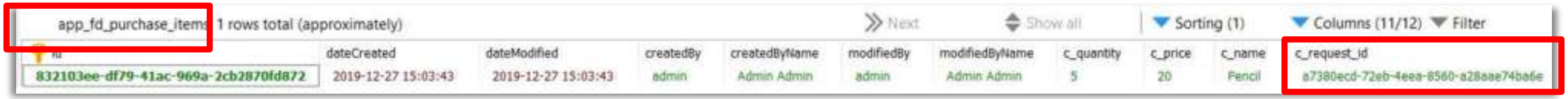
Exercise - Optional

- Inspect the database table of “Items”, you will notice that rows of data is now being saved into this table rather than the parent table as JSON.



.app_fd_purchase_requests: 1 rows total (approximately) Next Show all Sorting Columns (13/13) Filter

id	dateCreated	dateModified	createdBy	createdByName	modifiedBy	modifiedByName	c_name	c_request_date	c_items
a7380ecd-72eb-4eea-8560-a28aae74ba6e	2019-12-27 15:03:43	2019-12-27 15:03:43	admin	Admin Admin	admin	Admin Admin	Admin Admin	12/27/2019	(NULL)



app_fd_purchase_items 1 rows total (approximately) Next Show all Sorting (1) Columns (11/12) Filter

id	dateCreated	dateModified	createdBy	createdByName	modifiedBy	modifiedByName	c_quantity	c_price	c_name	c_request_id
832103ee-df79-41ac-969a-2cb2870fd872	2019-12-27 15:03:43	2019-12-27 15:03:43	admin	Admin Admin	admin	Admin Admin	5	20	Pencil	a7380ecd-72eb-4eea-8560-a28aae74ba6e

Materials

- "1-Submit Request" form definition is available at **13.4.2**.txt
- “Items” form definition is available at **13.4.3**.txt
- Complete app is available at **13.4.4**.jwa

Chapter Review

- Understand the use case of the Multirow Form Binder and its benefits.

RAD Service

Chapter 5

List Grid

RAD Service

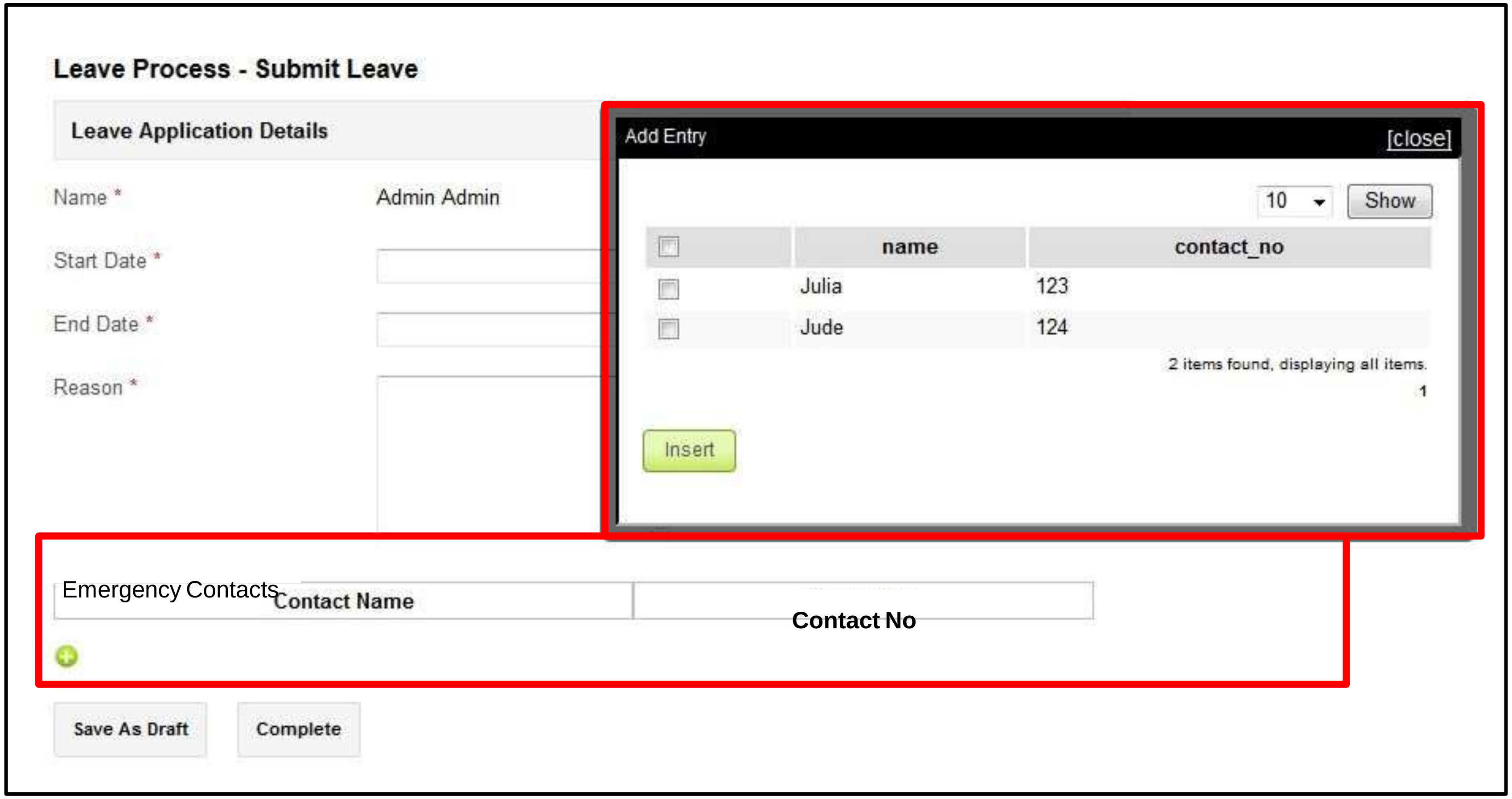
List Grid

- **List Grid** is a grid table that populates its data from a Datalist.
- It behaves similarly like a Grid (Chapter 2) but new rows are added from a specified Datalist instead.
- It also behaves similarly like a Form Grid that allows one to open up a Form for editing.

- Reference: https://docs.rads.purwana.net/List+Grid

RAD Service

Sample Use Case



Leave Process - Submit Leave

Leave Application Details

Name *	Admin Admin
Start Date *	
End Date *	
Reason *	

Add Entry [close]

10 Show

☐	name	contact_no
☐	Julia	123
☐	Jude	124

2 items found, displaying all items.

1

Insert

Emergency Contacts

Contact Name	Contact No

Save As Draft

Complete

Chapter Review

- Understand the List Grid element and be able to think of use cases of it.
- Able to differentiate Grid, Form Grid, and List Grid.

Chapter 6

CRUD

RAD Service

CRUD

- CRUD is a Userview Menu allows one to easily achieve the functionality of **C**reate, **R**etrieve, **U**pdate, and **D**elete on a data entity.
- In short, manipulate records on a specified table.

- Reference: https://docs.rads.purwana.net/CRUD

What Is Needed For CRUD To run?

- A Form entity
- A List of the same data entity as the form
- A Userview

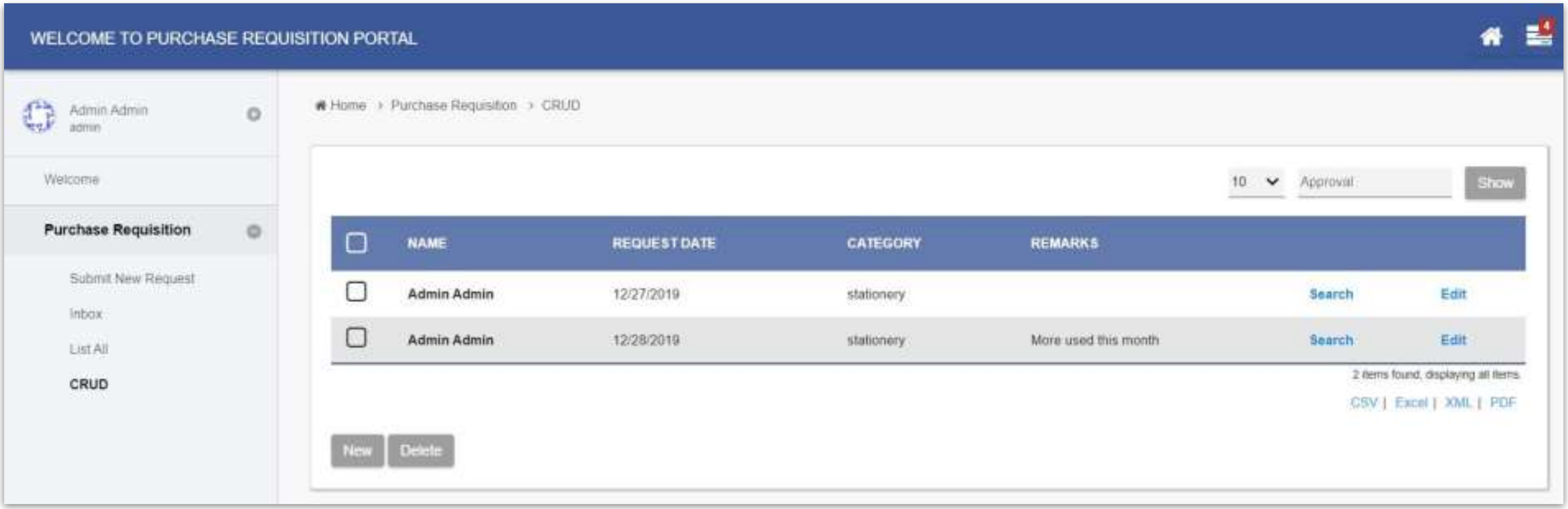
Refresh

- You've already done it! Refresh what you did back in Module 8 - Designing your first Userview
- If you do not have the CRUD for "Request List", ask your colleague how to, or raise hand...

RAD Service

How CRUD Looks Like...

- This is how the CRUD element would look like in runtime.



	NAME	REQUEST DATE	CATEGORY	REMARKS		
☐	Admin Admin	12/27/2019	stationery		Search	Edit
☐	Admin Admin	12/28/2019	stationery	More used this month	Search	Edit

2 items found, displaying all items.

CSV | Excel | XML | PDF

Chapter Review

- Able to use CRUD and understand the linkages.

Chapter 7

Custom HTML

Custom HTML

Custom HTML in Form Builder can be used to achieve advanced form design by putting in any valid -

- **HTML**

  E.g: `<b>this text is in bold</b>`

- **JavaScript** (jQuery is supported)

  Remember to put in `<script type="text/javascript"></script>` block

- **CSS**

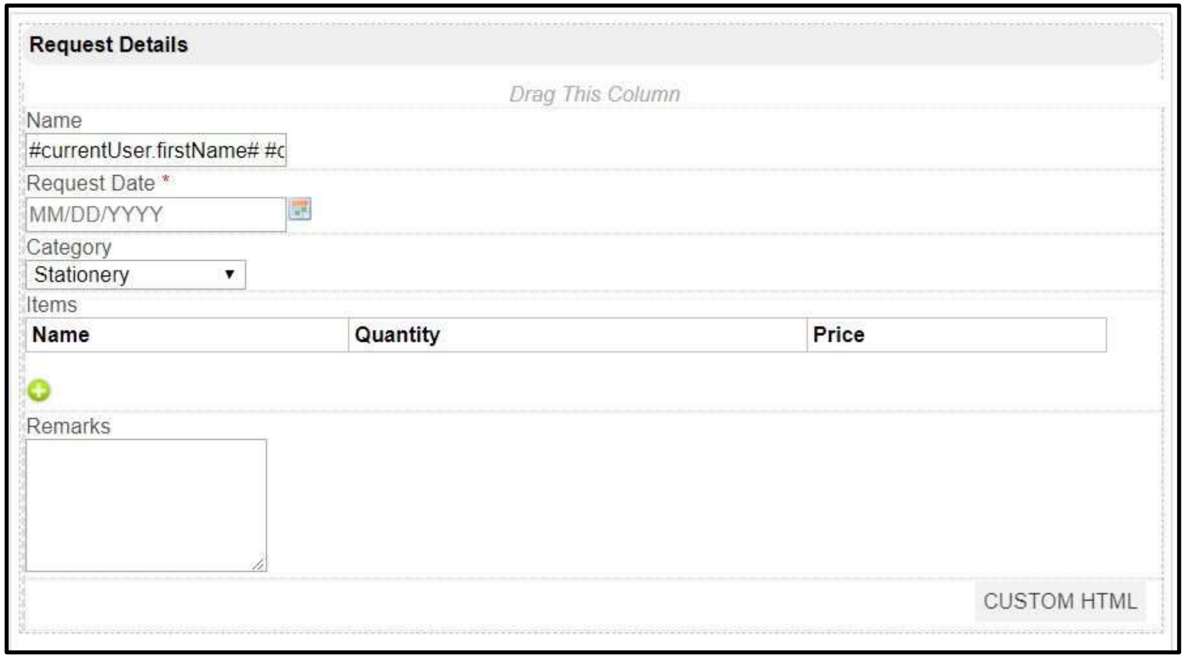
  Don't forget to put in <style type="text/css"></style> block

Reference: https://docs.rads.purwana.net/Custom+HTML

Exercise - CSS (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
#currentUser.firstName# #c

Request Date *
MM/DD/YYYY

Category
Stationery

Items

Name	**Quantity**	**Price**

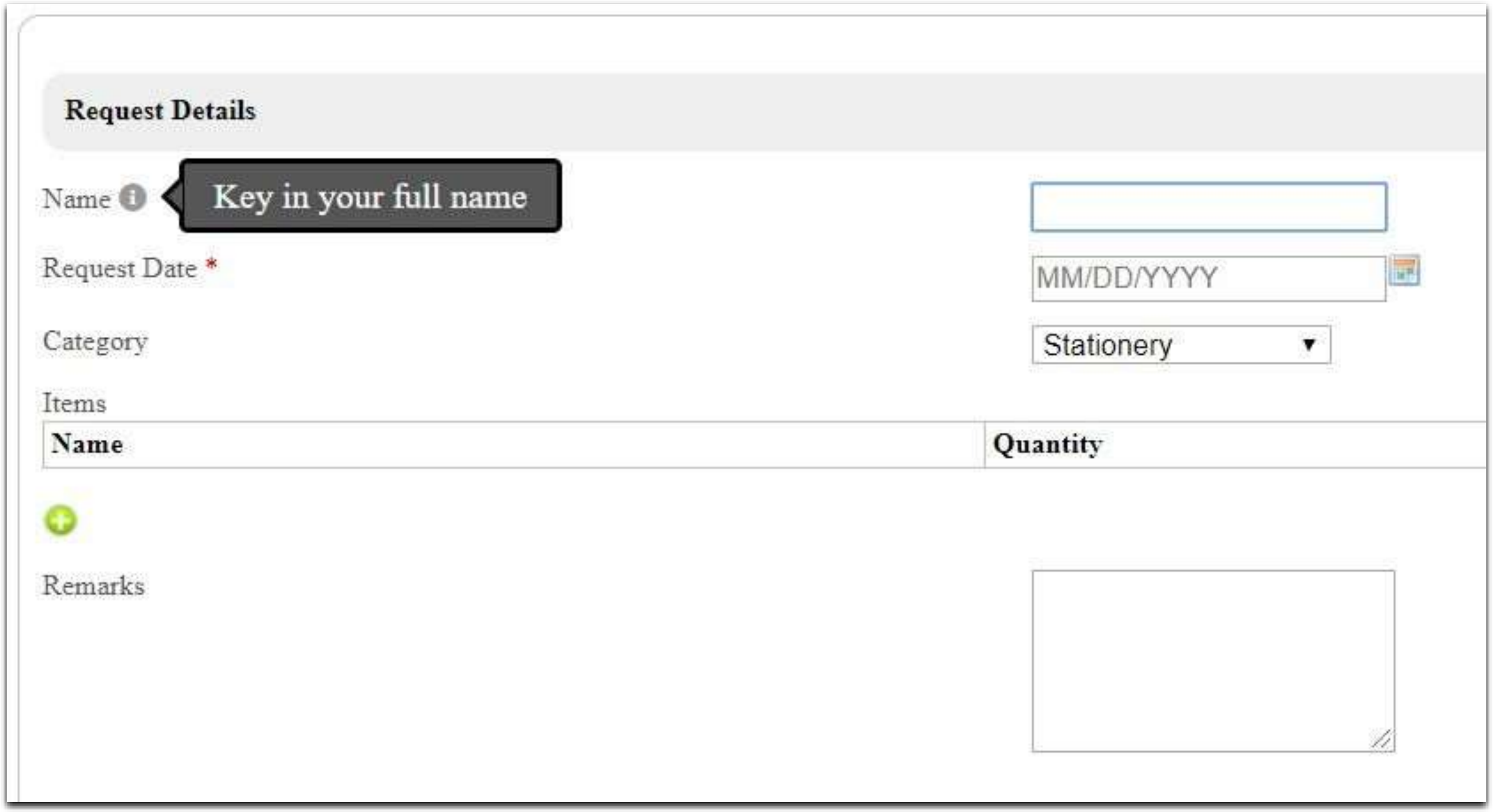
Remarks

CUSTOM HTML

RAD Service

Exercise - Tooltip

- Make use of Advanced Tool's Tooltip to add hints into existing form "Submit Request" using Custom HTML also.



Request Details

Name
Key in your full name

Request Date *
MM/DD/YYYY

Category
Stationery

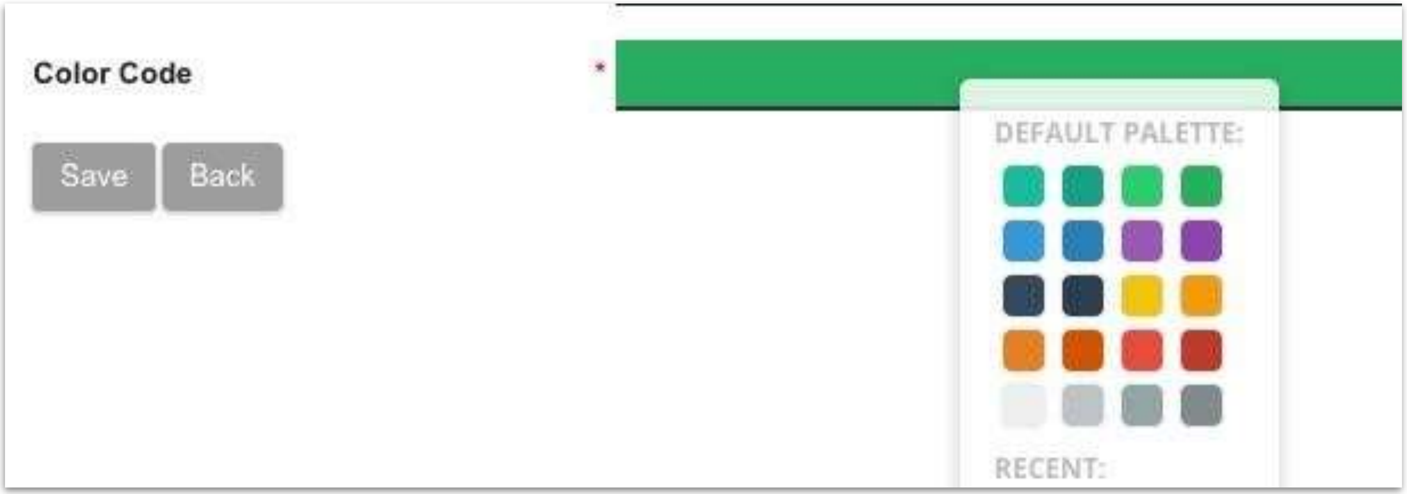
Items

Name	Quantity

Remarks

Exercise - Color Picker (Optional)

- Create a text field “Color Code” with ID “color_code”
- Create a Custom HTML. Make use of color picker library to turn text field into a color picker.



Exercise - Color Picker - Materials

Do import these JS and CSS libraries into the app's resource from the materials folder:

- colorPick.min.css
- colorPick.min.js

Exercise - Color Picker

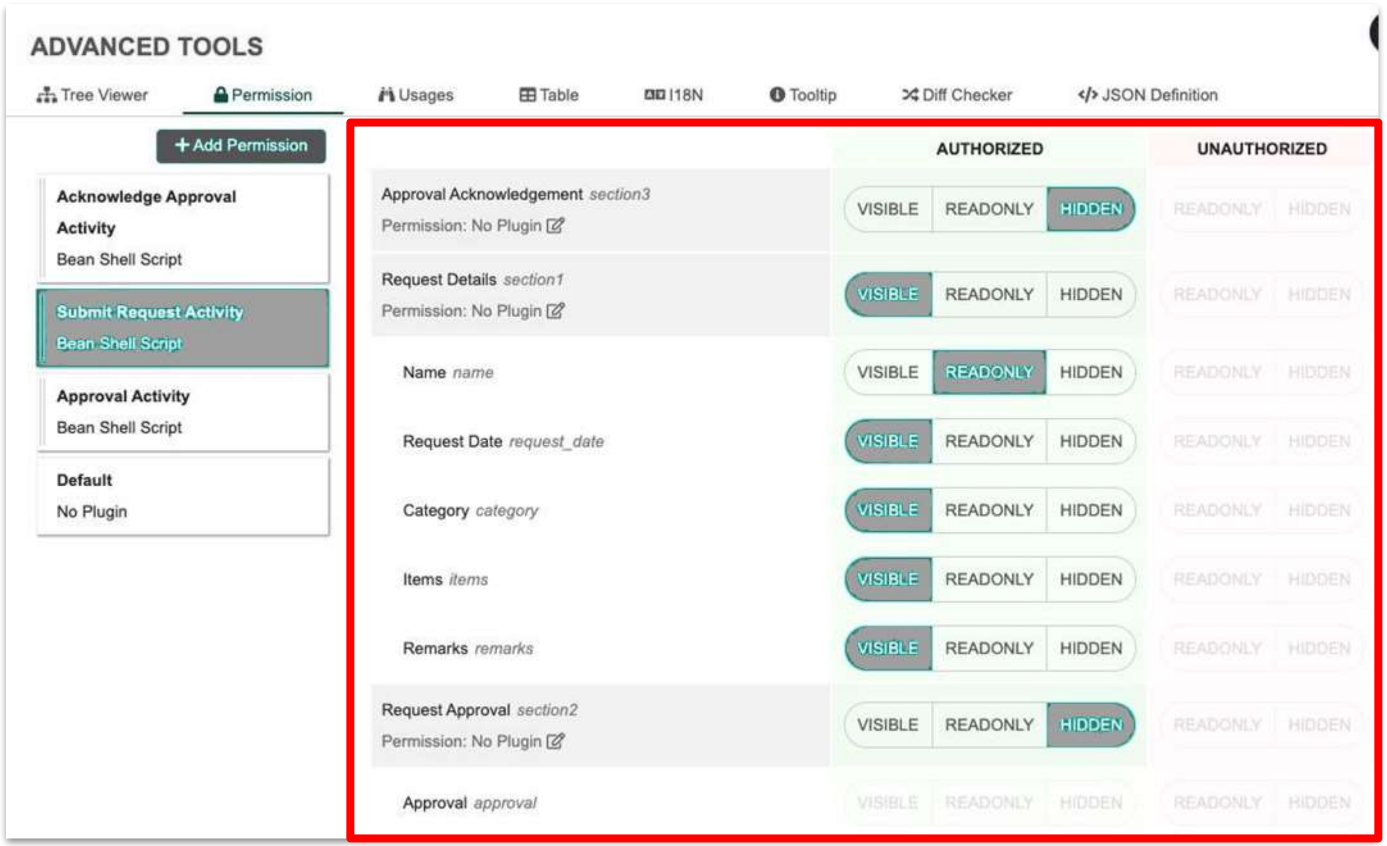
```
<script src="#appResource.colorPick.min.js#"></script>
<link rel="stylesheet" href="#appResource.colorPick.min.css#">

<script type="text/javascript">
  $(function(){
     initialColor = FormUtil.getField("color_code").val();
     //console.log("initial " + initialColor);
     FormUtil.getField("color_code").colorPick({
       'initialColor' : initialColor,
       'onColorSelected': function() {
          //console.log("The user has selected the color: " + this.color);
          FormUtil.getField("color_code").val(this.color);
          this.element.css({'backgroundColor': this.color, 'color': this.color});
       }
     });
  });
</script>
```

Chapter 8

Using Advanced Tool's Permission

Permission Control

- The Permission in Advanced Tools allows fine grain control up to field level.



ADVANCED TOOLS

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

+ Add Permission

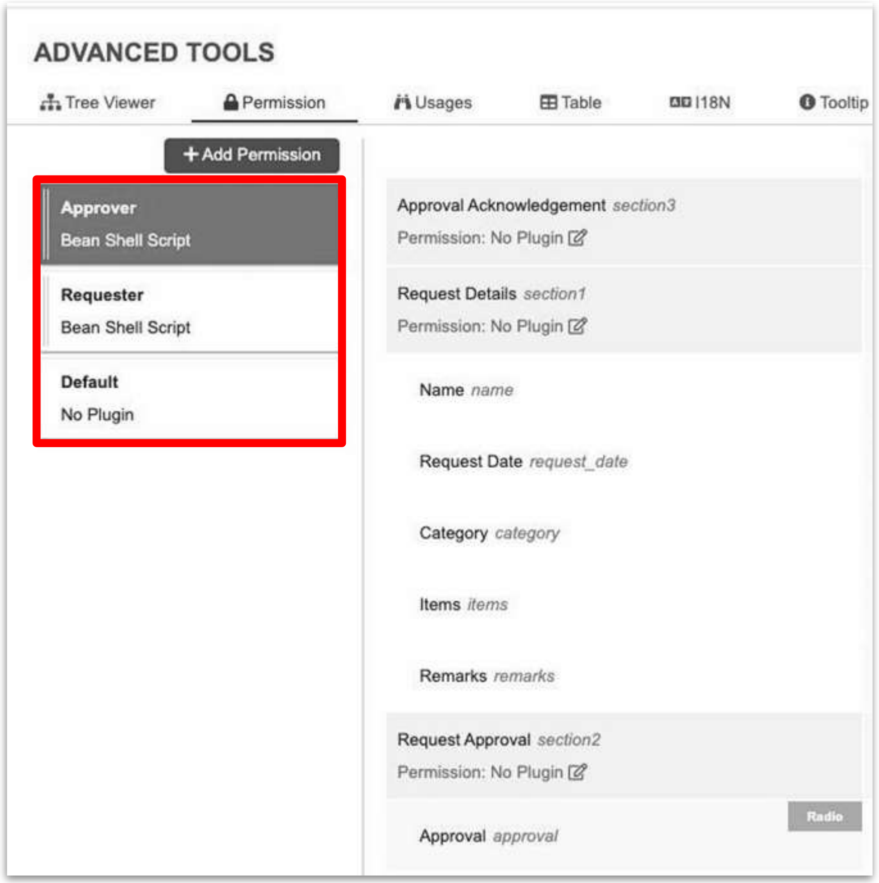
- **Acknowledge Approval Activity** Bean Shell Script
- **Submit Request Activity** Bean Shell Script
- **Approval Activity** Bean Shell Script
- **Default** No Plugin

	AUTHORIZED	UNAUTHORIZED
Approval Acknowledgement *section3* Permission: No Plugin	VISIBLE READONLY [✓ HIDDEN]	READONLY HIDDEN
Request Details *section1* Permission: No Plugin	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Name *name*	VISIBLE [✓ READONLY] HIDDEN	READONLY HIDDEN
Request Date *request_date*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Category *category*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Items *items*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Remarks *remarks*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Request Approval *section2* Permission: No Plugin	VISIBLE READONLY [✓ HIDDEN]	READONLY HIDDEN
Approval *approval*	VISIBLE READONLY HIDDEN	READONLY HIDDEN

RAD Service

Permission Control

- Typically, one would control **based on User Role** most of the time to show/hide fields.



ADVANCED TOOLS

Tree Viewer | Permission | Usages | Table | I18N | Tooltip

+ Add Permission

Approver
Bean Shell Script

Requester
Bean Shell Script

Default
No Plugin

Approval Acknowledgement *section3*
Permission: No Plugin

Request Details *section1*
Permission: No Plugin

Name *name*

Request Date *request_date*

Category *category*

Items *items*

Remarks *remarks*

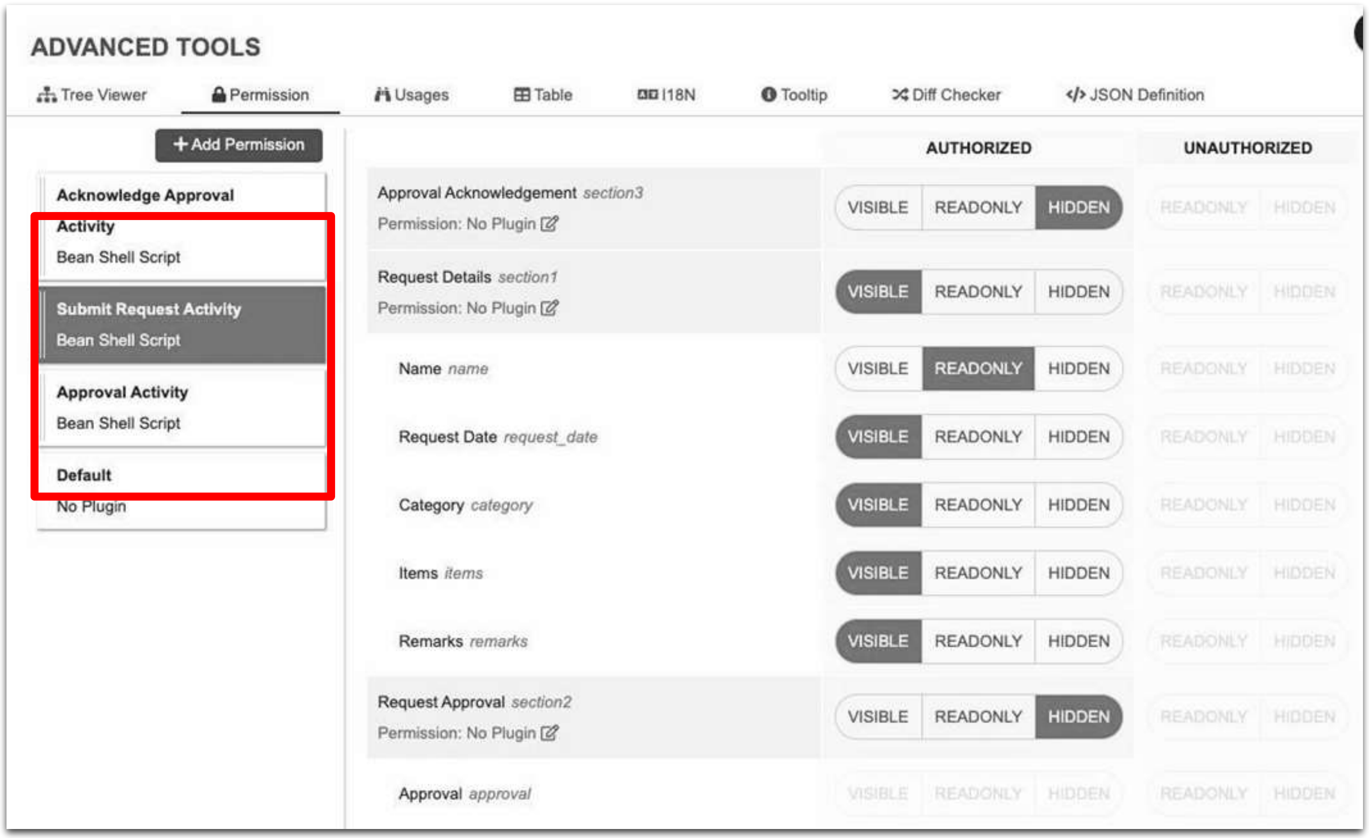
Request Approval *section2*
Permission: No Plugin

Approval *approval*

Radio

Permission Control

- If the form is used part of a process flow, it is also possible to exert permission **based on activity**, rather than user role.



Acknowledge Approval Activity
Bean Shell Script

Submit Request Activity
Bean Shell Script

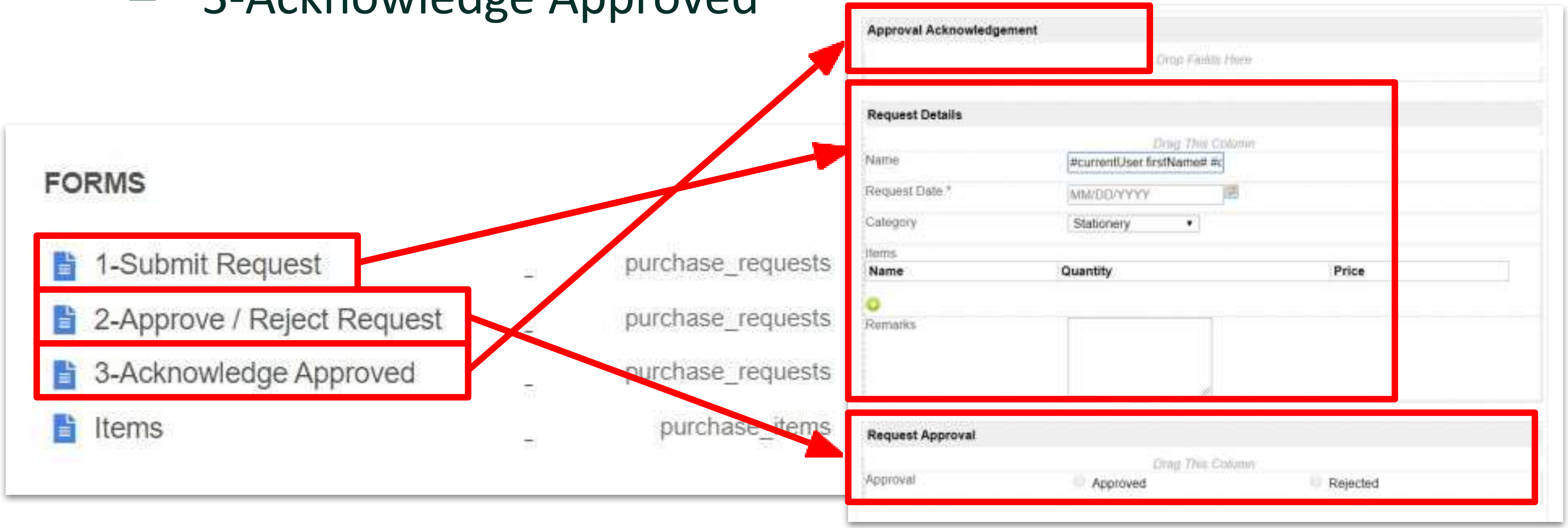
Approval Activity
Bean Shell Script

Default
No Plugin

	AUTHORIZED	UNAUTHORIZED
Approval Acknowledgement *section3* Permission: No Plugin	VISIBLE READONLY [✓ HIDDEN]	READONLY HIDDEN
Request Details *section1* Permission: No Plugin	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Name *name*	VISIBLE [✓ READONLY] HIDDEN	READONLY HIDDEN
Request Date *request_date*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Category *category*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Items *items*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Remarks *remarks*	[✓ VISIBLE] READONLY HIDDEN	READONLY HIDDEN
Request Approval *section2* Permission: No Plugin	VISIBLE READONLY [✓ HIDDEN]	READONLY HIDDEN
Approval *approval*	VISIBLE READONLY HIDDEN	READONLY HIDDEN

Exercise - Advanced Tool's Permission

- *One Master Form Concept* - Try combining the 3 forms in Purchase Requisition into a single form by using the Form Builder's Advance Tool Permission Tab:
  - 1-Submit Request
  - 2-Approve / Reject Request
  - 3-Acknowledge Approved



FORMS

1-Submit Request	-	purchase_requests
2-Approve / Reject Request	-	purchase_requests
3-Acknowledge Approved	-	purchase_requests
Items	-	purchase_items

Approval Acknowledgement

Drop Fields Here

Request Details

Drag This Column

Name: #currentUser.firstName# #c

Request Date *: MM/DD/YYYY

Category: Stationery

Items

Name	Quantity	Price

Remarks

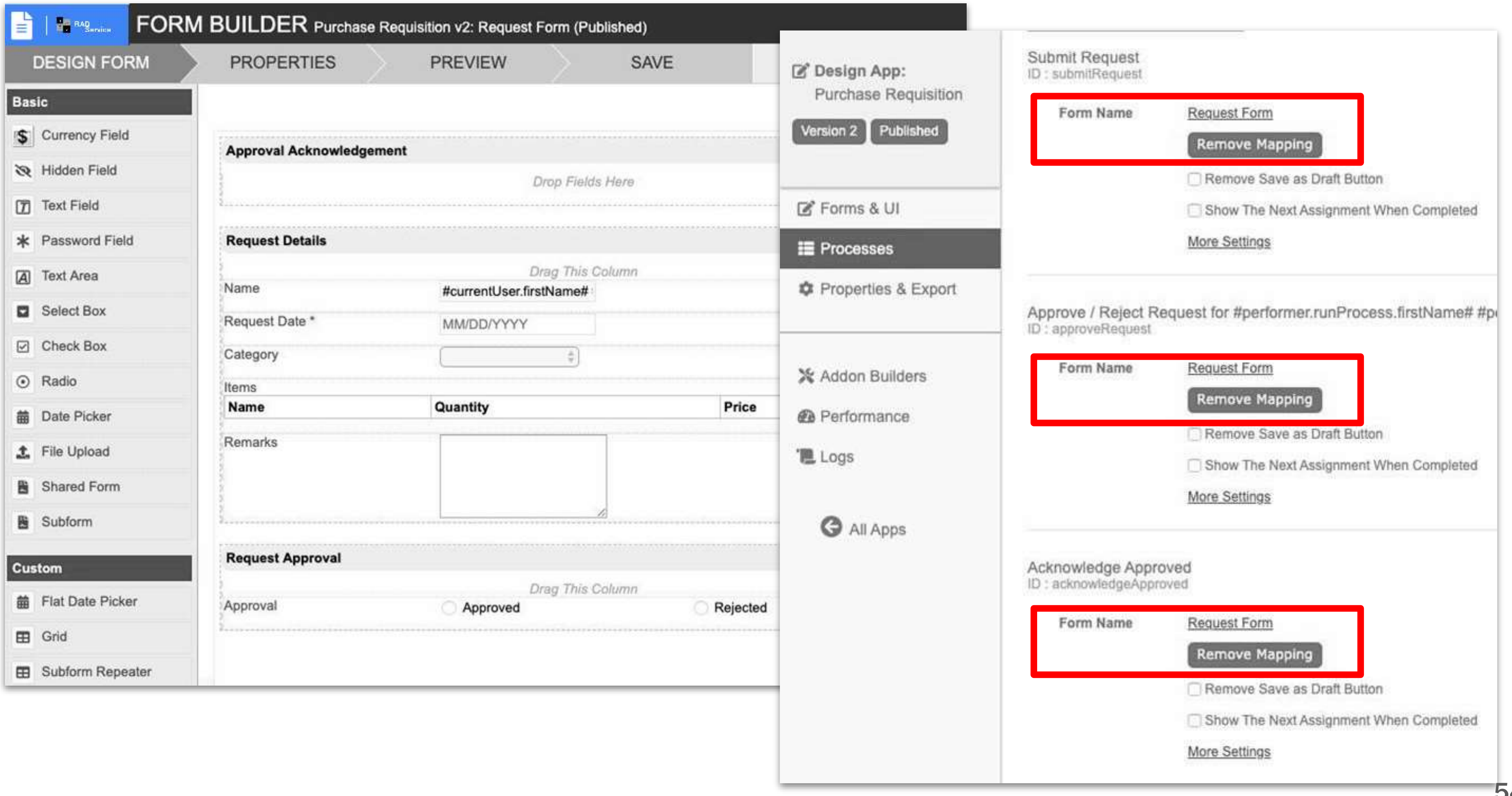
Request Approval

Drag This Column

Approval: Approved Rejected

Exercise - Advanced Tool's Permission

- The same form will be used in all activities.



FORM BUILDER Purchase Requisition v2: Request Form (Published)

DESIGN FORM | PROPERTIES | PREVIEW | SAVE

Basic: Currency Field, Hidden Field, Text Field, Password Field, Text Area, Select Box, Check Box, Radio, Date Picker, File Upload, Shared Form, Subform

Custom: Flat Date Picker, Grid, Subform Repeater

Approval Acknowledgement

Drop Fields Here

Request Details

Drag This Column

Name: #currentUser.firstName#

Request Date *: MM/DD/YYYY

Category

Items

Name	Quantity	Price

Remarks

Request Approval

Drag This Column

Approval: Approved / Rejected

Design App: Purchase Requisition — Version 2, Published

Forms & UI · Processes · Properties & Export · Addon Builders · Performance · Logs · All Apps

Submit Request
ID : submitRequest

Form Name: Request Form — Remove Mapping

- [ ] Remove Save as Draft Button
- [ ] Show The Next Assignment When Completed

More Settings

Approve / Reject Request for #performer.runProcess.firstName# #p
ID : approveRequest

Form Name: Request Form — Remove Mapping

- [ ] Remove Save as Draft Button
- [ ] Show The Next Assignment When Completed

More Settings

Acknowledge Approved
ID : acknowledgeApproved

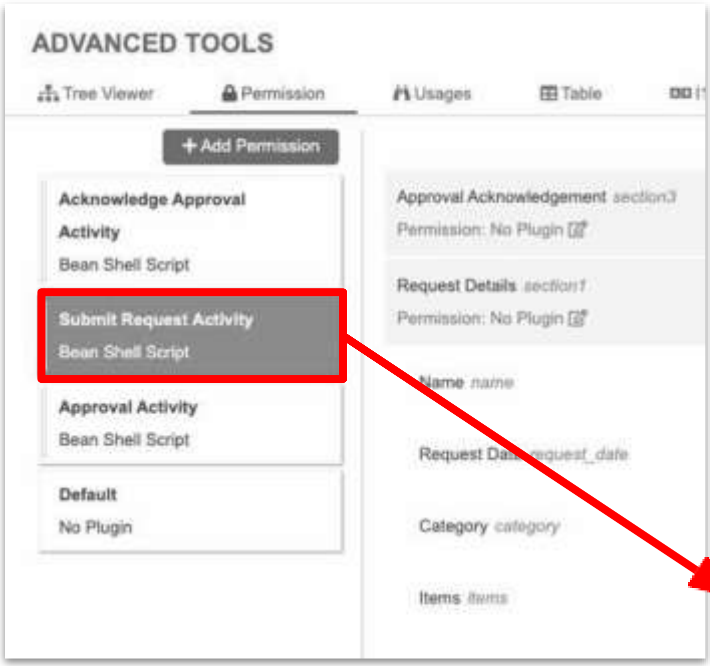
Form Name: Request Form — Remove Mapping

- [ ] Remove Save as Draft Button
- [ ] Show The Next Assignment When Completed

More Settings

Exercise - Advanced Tool's Permission

- In the Permission tab, we can control on what to show based on current activity.



Acknowledge Approval Activity
Bean Shell Script

Submit Request Activity
Bean Shell Script

Approval Activity
Bean Shell Script

Default
No Plugin

Approval Acknowledgement section3
Permission: No Plugin

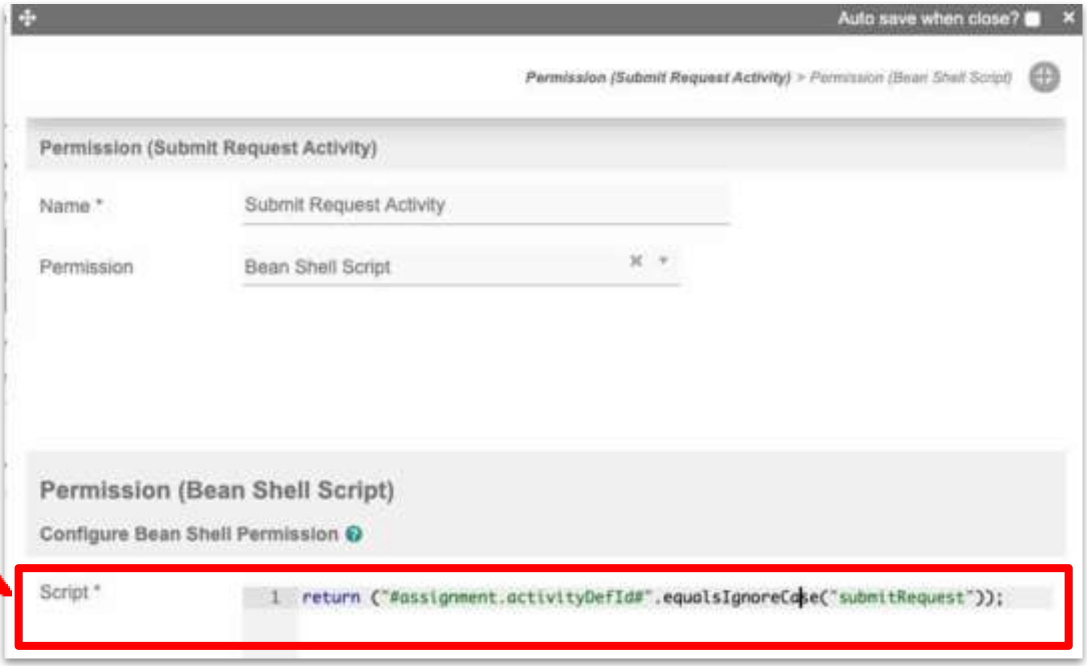
Request Details section1
Permission: No Plugin

Name name

Request Date request_date

Category category

Items items


Auto save when close?
Permission (Submit Request Activity) > Permission (Bean Shell Script)

Permission (Submit Request Activity)

Name *	Submit Request Activity
Permission	Bean Shell Script

Permission (Bean Shell Script)

Configure Bean Shell Permission

Script *

```
return ("#assignment.activityDefId#".equalsIgnoreCase("submitRequest"));
```

```
return ("#assignment.activityDefId#".equalsIgnoreCase("submitRequest"));
```

Exercise - Advanced Tool's Permission

- Once you are done setting up, try to run through the whole flow and check if the form is showing up correctly as per the activity.

Materials

- Complete app is available at **13.8.1**.jwa

RAD Service

Module Review

1. Introduction
2. Grid
3. Form Grid
4. Multirow Form Binder
5. List Grid
6. CRUD
7. Custom HTML
8. Using Advanced Tool’s Permission

Learn More...

- Check out https://docs.rads.purwana.net/Form+Builder for the list of **Form** related elements (Form Element, Form Validator, Form Binder, Form Options Binder).
- Check out https://docs.rads.purwana.net/Userview+Builder for the list of **Userview** related elements.

Stay Connected With RADS

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