

**Practices for Lesson 4:**  
**Configure the Order and Item**  
**Creation and Edit Order**  
**Pages**

## Overview

---

In these practices, you will:

- Use Advanced Create and Edit Page Template to add the Create Order/Item functionality
- Create the Edit Orders page using Simple Create Edit Page Template

**Note:** These practices focus on configuring the Page Templates and do not include implementing the create/edit back-end integration and functionality.

## Practice 4-1: Create the New Order and Line Items Page

### Overview

In this practice, you configure pages to add a new order, add line items, and add products for the line items.

You work with Sections and the Vertical Anchor to navigate around the sections.

You will focus on the layout of the page and working with new components. Some functionality, like inserting rows and saving to the database using the REST service connection and API, are not implemented because these are not relevant for learning the Redwood components and are outside the scope of this course.

**Note:** Inserting a new row in a table is somewhat complex and is performed through coding. You can see how to do this in the Visual Builder Cookbook, an excellent resource for learning how to develop in Visual Builder: <https://vbcookbook.oracle.com/?page=shell&shell=table-add-row>

The screenshot shows the 'New Order' page interface. At the top right are 'Cancel' and 'Submit' buttons. Below the title 'New Order' is a 'Page Subtitle'. A date and time indicator '7/25/2024' and 'Total Amount: \$0.00' are displayed. On the left, a dropdown menu shows 'Select (Single)'. The main area is titled 'Order Details' and contains two columns of input fields. The left column includes 'AttachmentEntityName' (Required), 'BillingAddressCity' (Required), 'BillingAddressLine1\_1' (Required), 'BillingAddressPostalCode' (Required), 'BillToAddressId' (Required), and 'Created By'. The right column includes 'BasketFlag' (Required), 'BillingAddressCountry' (Required), 'BillingAddressLine2\_1' (Required), 'BillingAddressState' (Required), '@context/key' (Required), and 'Created'. To the right of these fields is a vertical sidebar labeled 'Order Details' which lists 'Order Line Items' and 'Products'. At the bottom right is a small icon of a person at a computer.

### Assumptions

You have successfully completed Practice 3-1.

## Tasks

1. Log in to **Visual Builder Studio** and select your project name on the **Projects** tab. Your project name will be **OrderMgt\_< your initials >**, where **< your initials >** corresponds to your initials.

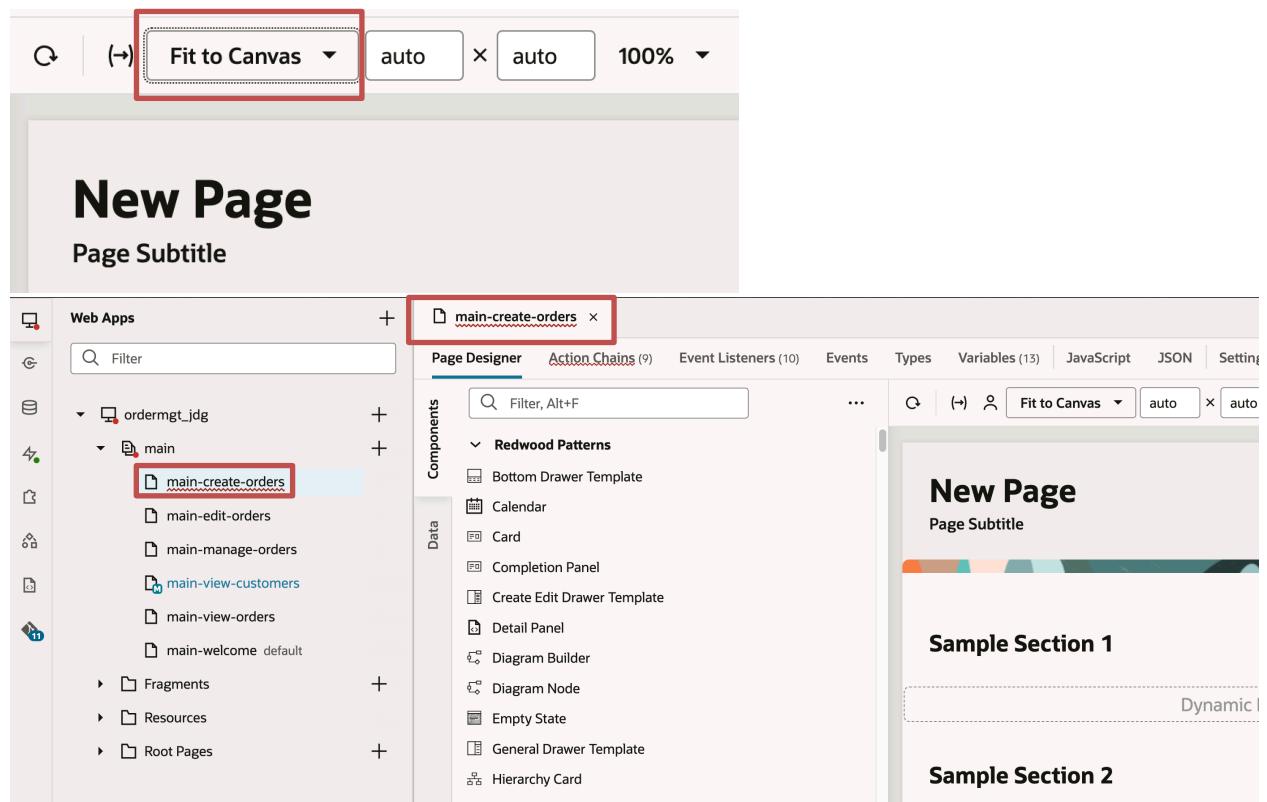
The screenshot shows the Visual Builder Studio interface. On the left, the sidebar has 'Organization' selected, with options like Project Home, Workspaces, Git, Merge Requests, Maven, NPM, and Docker. The main area is titled 'Visual Builder Studio' and 'Organization'. The 'Projects' tab is selected. There is a search bar and filters for Member, Favorites, Owner, Shared, All, and a magnifying glass icon. Below is a table with columns for Name (sorted by ascending Name), Template (sorted by descending Favorite), and Status (sorted by descending Status). A row for 'OrderMgt\_JDG' is highlighted with a red box. The details for 'OrderMgt\_JDG' show it is a 'Current' 'Order Management Application' and is 'Active'.

2. The Project Home page opens. Right-click the **OrderMgt\_<your initials>** workspace and select **Open in New Tab**.

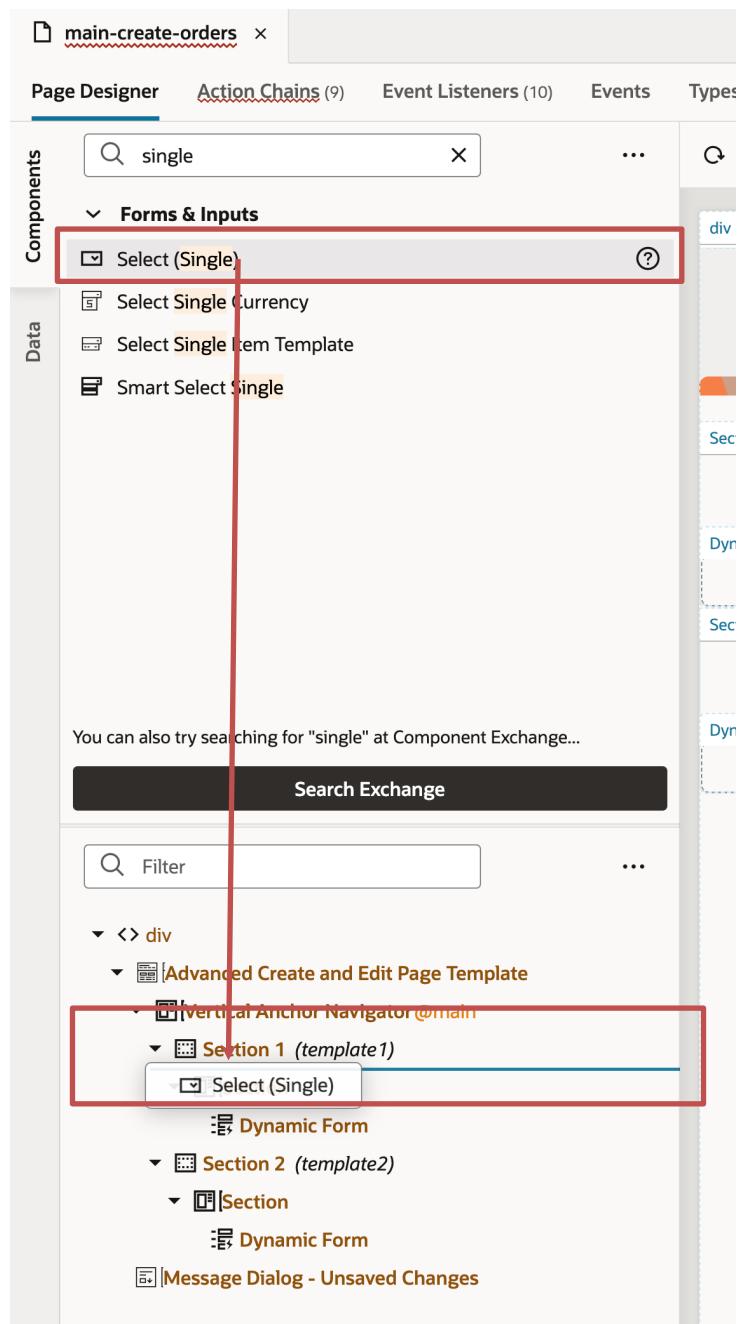
The screenshot shows the Visual Builder Studio interface. On the left, the sidebar has 'Workspaces' selected. The main area is titled 'Visual Builder Studio' and 'OrderMgt\_JDG - Workspaces'. A search bar and buttons for Mine, Others, Workspace, Repository, and Current Branch are shown. The 'OrderMgt\_JDG' workspace is listed with details: 'Visual Application', 'ordermgt\_jdg.git', and 'Order\_Mgt'. A context menu is open over this workspace entry, with the 'Open in New Tab' option highlighted by a red box. Other menu items include Rename, Export Visual Application, Switch Environment, Change Ownership, and Delete.

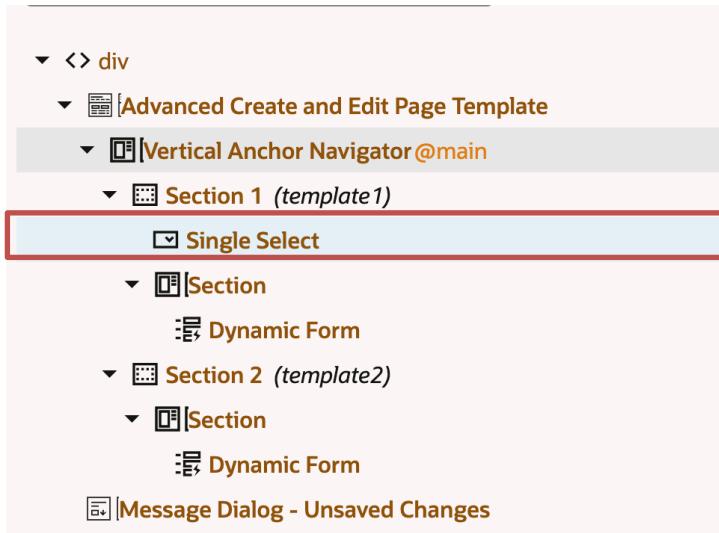
3. Close the **Getting Started** tab if it's open.
4. Select the **Web Apps** tab in the Navigator. Expand **ordermgt\_<your initials>**. Expand **main**. Select the **main-create-orders** page. It opens in the Page Designer.

5. If the screen size is still set to Desktop from the previous practice, change it to **Fit to Canvas**.



6. Add a **Select (Single)** to **Section 1**, above the **Section** component. **Tip:** Note the green horizontal line indicating where it will be inserted.





The screenshot shows the Oracle APEX Page Designer interface. On the left, the 'Components' section under 'Forms & Inputs' is expanded, showing options like 'Select (Single)', 'Select Single Currency', 'Select Single Item Template', and 'Smart Select Single'. A search bar at the top of this panel shows 'single'. In the center, a page titled 'New Page' is displayed with sections for 'Page Subtitle' and 'Sample Section 1'. In the 'Sample Section 1' area, there is a 'Dynamic Form' containing a 'Select (Single)' component. On the right, the 'Vertical Anchor Navigator' is open, showing a tree structure of components. A node labeled 'Single Select' is highlighted with a red box. The full path in the navigator is: Advanced Create and Edit Page Template > Vertical Anchor Navigator @main > Section 1 (template1) > Single Select.

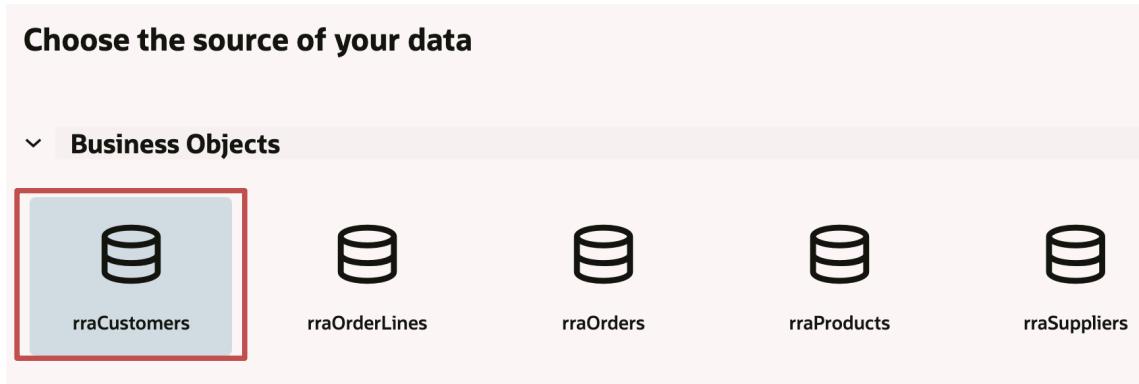
7. With the Single Select selected, click its **Quick Start** tab. Click **Add Options** to define its data.

This screenshot shows the 'Single Select' component's configuration screen. The 'Properties' panel on the right has a tab bar with 'General', 'Data', 'Events', 'All', and 'Quick Start', with 'Quick Start' currently selected. A large callout box highlights the 'Add Options' button, which is described as 'Map the component to a data source, so we can populate it with data.' The 'General' tab is also visible in the background.

8. Select **rraCustomers** as your data source and click **Next**.

**Choose the source of your data**

▼ **Business Objects**



rraCustomers      rraOrderLines      rraOrders      rraProducts      rraSuppliers

9. Select **customerName** for **Label** and **customerId** for **Value**, and click **Next** and then click **Finish**.

**Dropdown options**

**Label**

A customerName X

**Value**

# customerId X

10. Test that it works in **Live** mode. Click the pull-down menu and, after a moment, the list of customers appears. Close the list and return to **Design** mode.

The screenshot shows a "New Page" interface with a "Page Subtitle". At the top right, there are tabs for "Live" (which is highlighted with a red box), "Design", and "Code". Below the tabs, there is a "Cancel" button and a "Submit" button. The main content area has a title "Sample Section 1". On the left, there is a dropdown menu with a placeholder "Select (Single)". The dropdown contains a list of names: Pete Chevis, Joseph Wilke, Azalee Goodwater, Stefany Waninger, Arron Waclawski, Boyd Rearden, Erasmo Riviera, and Ambrose Hiday. To the right of the dropdown, there is a sidebar with a title "Sample Section 1" and a section labeled "Sample Section 2".

11. Notice the single select is wider than it needs to be. Click the Single Select's **General** tab. Select **Small (oj-form-control-max-width-sm)** for **Styling**. Test the single select again. It now adjusts to fit the contents better. Return to Design mode when done.

The screenshot shows the "Styling" tab for a component. The tab has a question mark icon. Below the tab, there are several styling options listed with checkboxes:

- Medium (oj-form-control-max-width-md)
- oj-form-control-text-align-end
- oj-form-control-text-align-start (oj-form-control-text-align-start)
- Small (oj-form-control-width-sm)
- Small (oj-form-control-max-width-sm)
- Medium (oj-form-control-width-md)
- oj-form-control-text-align-start (oj-form-control-text-align-right)

The last checkbox, "Small (oj-form-control-max-width-sm)", is checked and highlighted with a red box.

## New Page

Page Subtitle



### Sample Section 1

Select (Single) ▾

Pete Chevis

Joseph Wilke

Azalee Goodwater

Stefany Waninger

12. Select the **Advanced Create and Edit Page Template**. Set its Page Title to New Order.

The screenshot shows the Oracle ADF 'Advanced Create and Edit Page Template' dialog. At the top, there's a tree view showing the page structure:

- < div
- Advanced Create and Edit Page Template (highlighted with a red box)
- Vertical Anchor Navigator @main
- Section 1 (template1)
  - Single Select

Below the tree, the 'Advanced Create and Edit Page Template' panel has the following fields:

- General** tab selected.
- ID**: oj\_ssce1
- Page Title**: New Order (highlighted with a red box)
- String not externalized for translation.

On the right, a vertical **Properties** panel is visible. At the bottom, a preview window shows the page structure:

Advanced Create and Edit Page Template

New Order

Page Subtitle

Decorative footer bar with colored segments: orange, brown, teal, dark blue.

13. Create variables to display contextual information. Click the **Variables** tab.

The screenshot shows the Oracle ADF Page Designer interface. At the top, there is a header bar with the title "main-create-orders". Below the header, there are several tabs: "Page Designer" (which is underlined, indicating it is the active tab), "Action Chains (9)", "Event Listeners (10)", "Events", "Types (1)", "Variables (14)" (which is highlighted with a red box), and "JavaScript".

14. Create a new variable named `contextualInfo` of type **Array**.

The screenshot shows the "Variable" creation dialog. The "ID" field contains "contextualInfo" (highlighted with a red box). The "Type" field is set to "Array" (highlighted with a blue box). The "Properties" panel on the right is visible.

15. Create two variables of type String to hold the contextual values: `customerName` and `totalAmount`.

The screenshot shows the "Variable" creation dialog for "customerName". The "ID" field contains "customerName". The "Type" field is set to "String". The "Properties" panel on the right is visible.

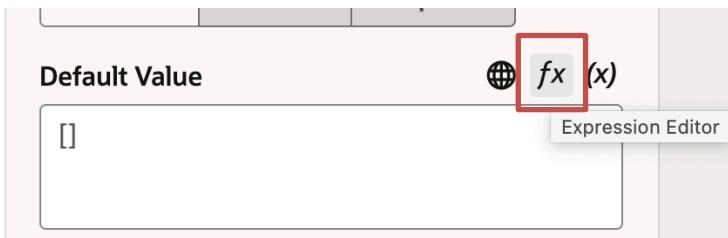
The screenshot shows the "Variable" creation dialog for "totalAmount". The "ID" field contains "totalAmount". The "Type" field is set to "String". The "Properties" panel on the right is visible.

```

▼ [] contextualInfo
  { } item[i]
    □ contextualInfoLabelVisible           SDP rraCustomersListSDP
    □ contextualInfoSticky
    A saveOption
    A customerName
    A totalAmount

```

16. Select **contextualInfo** and open the Expression Editor for its Default Value property by clicking the **fx** button.



17. Paste this code into the editor. Click **Save** when done. The code can be found in your code text file.

```

[
  {
    label: 'customer',
    value: $page.variables.customerName
  },
  {
    label: 'date',
    value: $flow.functions.formatDate( Date.now() )
  },
  {
    label: 'totalAmount',
    value: 'Total Amount: ' +
$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode,
$page.variables.totalAmount)
  }
]

```

**Expression Editor**

```

1  [
2   {
3     label: 'customer',
4     value: $page.variables.customerName
5   },
6   {
7     label: 'date',
8     value: $flow.functions.formatDate( Date.now() )
9   },
10  {
11    label: 'totalAmount',
12    value: 'Total Amount: ' + $flow.functions.formatCurrency($flow.constants.defaultCurrencyCode, $page.variables.totalAmount)
13  }
14 ]
15

```

- Click the **Page Designer** tab. With **Advanced Create and Edit Page Template** selected, click its **All** tab. Choose the **contextualInfo** variable for the **Contextual Info** property.

**<> div**

- Advanced Create and Edit Page Template**
- Vertical Anchor Navigator@main**
- Section 1 (template1)**

**Contextual Info**

**Variables**

Variable	Action
{ } avatar	Edit
[ ] contextualInfo	Edit

- The contextual info displays in the banner. As a new order is built on the page, the **totalAmount** variable would be updated and the banner would reflect the change in value.

Advanced Create and Edit Page Template

## New Order

Page Subtitle

7/25/2024 Total Amount: \$0.00

Select (Single)

### Sample Section 1

Dynamic Form

### Sample Section 2

General Events (4) All

Filter

Class

ID

oj\_ssce1

Advanced Create and Edit Page

Avatar Image

`{{page.variables.avatar}}`

Badge

Contextual Info

`[[ $variables.contextualInfo ]]`

20. Select the **first Dynamic Form** to create the form for the new order. Select its **Quick Start** tab and click **Configure as Create Form**.

Advanced Create and Edit Page Template

Vertical Anchor Navigator @main

Section 1 (template1)

Single Select

Dynamic Form

Dynamic Form

General Data Events (1) All Quick Start

Configure as Create Form  
Configure form component as a create form

Configure as Edit Form  
Configure form component as an edit form

Configure as Detail Form  
Configure form component as a (read-only) detail form

21. Select **rraOrders** for its data source and click **Next**.

**Choose the source of your data**

▼ **Business Objects**

rraCustomers      rraOrderLines      **rraOrders**      rraProducts      rraSuppliers

22. Enter `createOrder` for the **Label**. Click **Select fields to display >**.

Label \*  
createOrder

ID \*  
createOrder

Description

Use Simple Layout

Select fields to display >

23. Click **Finish** to accept all the fields.

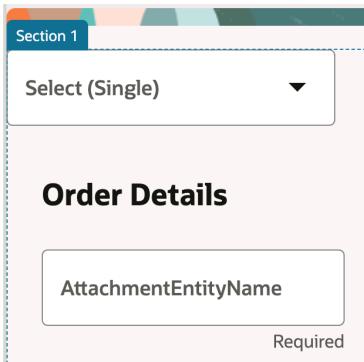
**Note:** In this example, you accept all the fields for simplicity, but in a real application, you would choose what would be entered, and you would want to reference the selected Customer Number and possibly use default values for other fields. You are not creating a working create operation; you are doing this to see the resulting form on the page.

The screenshot shows the Oracle Page Designer interface. At the top, there are two tabs: 'Locate Data' (marked with a green circle '1') and 'Select Rule Set' (marked with a black circle '2'). Below these tabs, there are fields for 'Label \*' (containing 'createOrder') and 'ID \*' (containing 'createOrder'). There is also a 'Description' field with an empty text area. On the left, a sidebar labeled 'Fields' contains a search bar with 'Filter' and a list titled 'Suggested Fields'. This list includes several checked options such as 'attachmentEntityName', 'basketFlag', 'billingAddressCity', etc. To the right of the sidebar, under 'Select fields to display', there is a 'Use template instead' link and a list of available fields with checkboxes, including 'attachmentEntityName', 'basketFlag', 'billingAddressCity', 'billingAddressCountry', 'billingAddressLine11', 'billingAddressLine21', 'billingAddressPostalCode', and 'billingAddressState'.

You are returned to Page Designer. After a moment, the form fields display.

24. Select the **Section 1** element and change its **Label Hint** to Order Details.

The screenshot shows the Oracle Page Designer element tree and properties panel for 'Section 1'. The element tree on the left shows a structure starting from 'div' and branching into 'Advanced Create and Edit Page Template', 'Vertical Anchor Navigator @main', and finally 'Section 1 (template1)'. The 'Section 1 (template1)' item is highlighted with a red border. The properties panel on the right shows the following settings for 'Section 1': 'ID' is set to 'template1', 'Title' is 'Section 1', 'Description' is 'Sample Section 1', and 'Label Hint' is 'Order Details'. The 'Order Details' field is also highlighted with a red border.



25. Select **Section 2** and change its **Label Hint** property to Order Line Items.

Advanced Create and Edit Page Template

Vertical Anchor Navigator @main

Section 1 (template1)

- Single Select

Section

Validation Group

Dynamic Form

Toolbar

Button (Save)

Section 2 (template2)

Section

Dynamic Form

Message Dialog - Unsaved Changes

Section 2

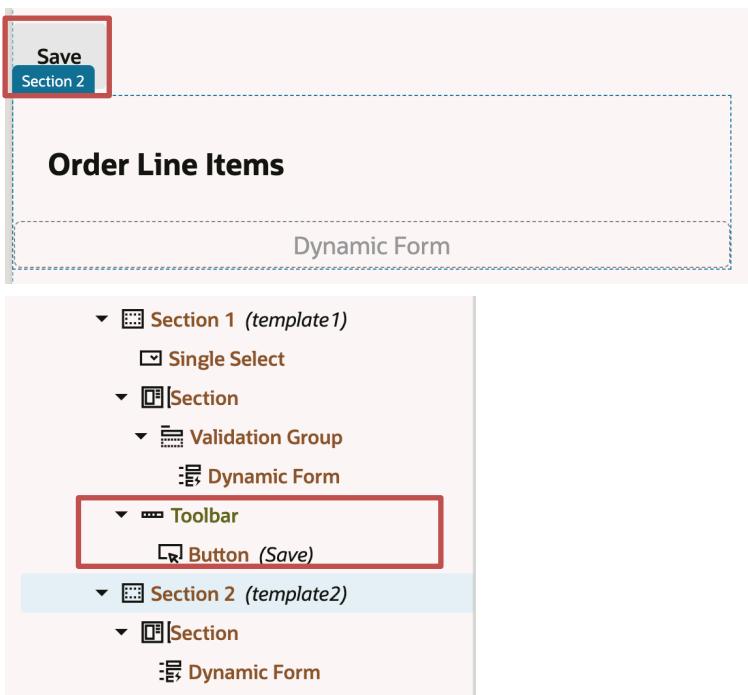
ID: template2

Title: Section 2

Description: Sample Section 2

Label Hint: Order Line Items

Notice a toolbar and **Save** button were added for the **Section 1** form to save the master order data first and then save the detail rows.

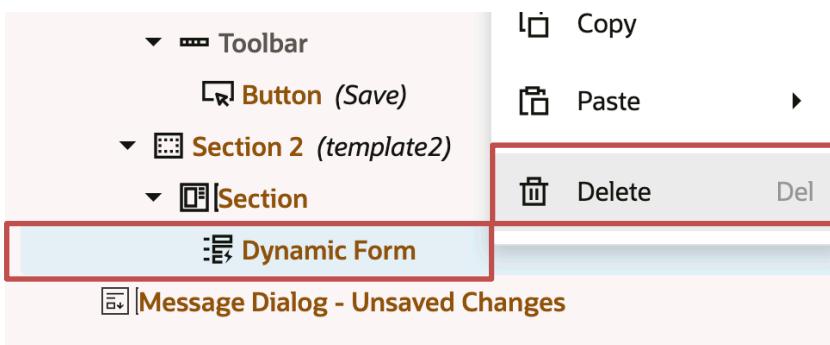


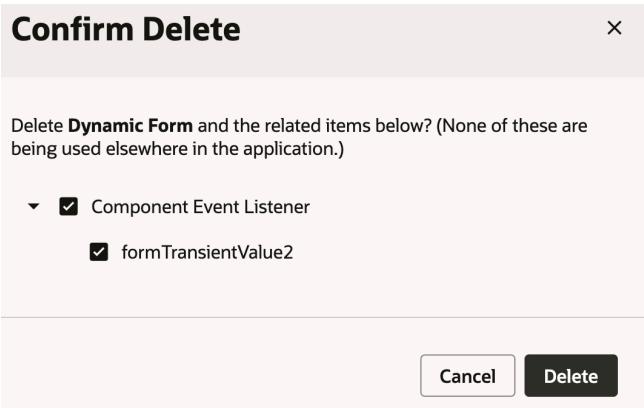
This is the default behavior, but you could combine the save of the order (master record) with its line items (detail records) and use the **Submit** button to save and commit both and then navigate back to the parent page (main-manage-orders), or use **Save** to save the order in case you still wanted to make changes before submitting and committing.

You could then add line items and use the **Submit** button to submit the added line items and order. You could also delete the Save button entirely and let the Submit button save the master record and details records and navigate back to the parent page. The Submit action chain is coded to call the Save action chain and then navigate back to the parent (root) page. This is implemented with action chains and implementing it is beyond the scope of this course. But notice the flexibility built into the page template.

Optional: Feel free to explore the event listeners and action chains to see how this is implemented.

26. Add a table to add line items to the order. Replace the **Dynamic Form** in **Section 2** with a table. Select the **Dynamic Form** in **Section 2** and delete it. Confirm the deletion of the **Component Event Listener** and **formTransientValue2**.





27. Take a moment to study the documentation for the **Advanced Create and Edit Page Template**. Click the **Components** tab, click **Installed** and click **Advanced Create and Edit Page Template** - not the pattern. Notice how objectId is used for the master record and is passed to the detail record. Also, the page can be used to retrieve a parent (master) record and then add children to it and submit the added children. This functionality is built in but is beyond the scope of this course to use.
28. Click the **Data** tab, expand **Business Objects > rraOrders > orderLinesCollection** and drag the **Get Many** to **Section** under **Section 2**. Choose **Table** to Render as. You can use a table to add multiple order lines and then commit them as a batch or use **Create** to add and save one line item at a time. Implementing this functionality is beyond the scope of this course.

Page Designer   Action Chains (10)   Event Listeners (10)   Events

Components   Data

rraOrders

- Get Many
- Create
- Get One
- Update
- customerIdObject
- orderLinesCollection

**Get Many**

**GET** /rraOrders/{rraOrders\_id}/child/orderLinesCollection

Business Object  
getall\_rraOrders-orderLinesCollection getMany  
Get all

rraSuppliers

Filter

<> div

Advanced Create and Edit Page Template

Vertical Anchor Navigator @main

Section 1 (template1)

Single Select

Section

Validation Group

Dynamic Form

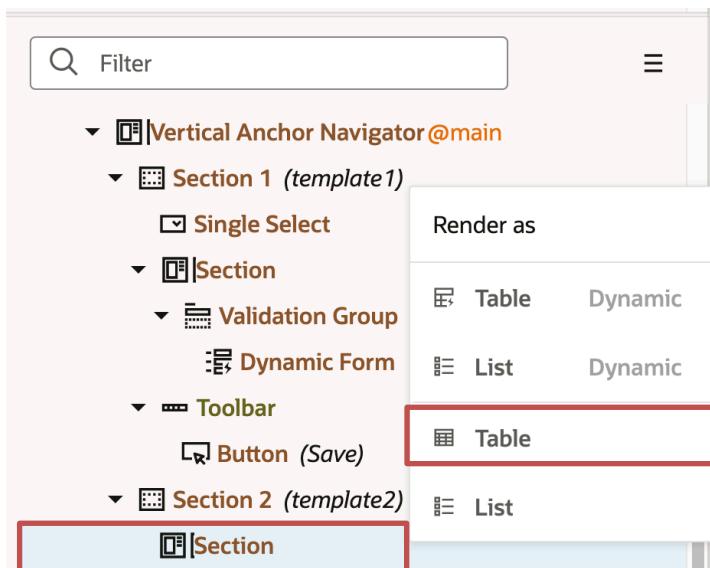
Toolbar

Button (Save)

Section 2 (template2)

Section

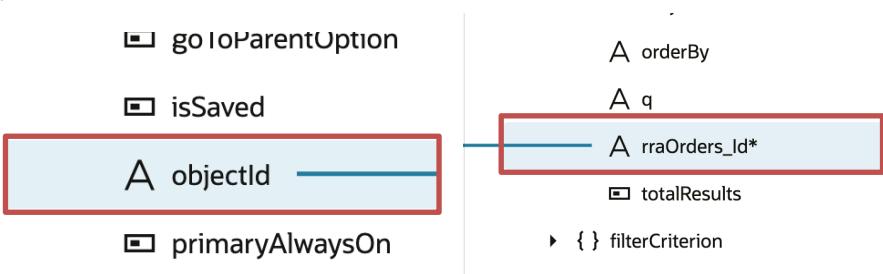
Message Dialog - Unsaved Changes



29. From the Endpoint Structure, select the following fields in order: **productNumber**, **productName**, **listPrice**, **quantity**, and **amount**. Change the component type for **quantity** to **Input Number**. Click **Next**.

Column	Type
A productNumber X	Text ▾
A productName X	Text ▾
# listPrice X	Text ▾
# quantity X	Input Number ▾
# amount X	Text ▾

30. Drag **objectId** to **Target > getall\_rra-rraOrders-orderLinesCollection > rraOrders\_Id**. ObjectId is the value from the parent record that is needed for the detail record as a "foreign key." Click **Finish**.



You see the Table display. You don't see any rows, as no order lines have been created yet. Normally, you would code this to allow you to use buffer variables to store the values for the new row and then insert it into the table. The technique to do this is

presented in the Visual Builder Cookbook that was mentioned at the beginning of this practice. You will not be implementing that functionality here because it is not relevant to Redwood.

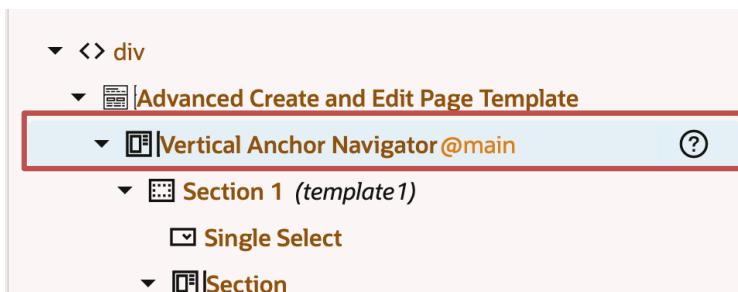
## Order Line Items

ProductNumber	ProductName	ListPrice	Quantity	Amount
---------------	-------------	-----------	----------	--------

No data to display.

To add a line item, you need to select the desired product, have its Product Number and List Price added to the new row, and then enter the quantity. The Amount would be calculated in a JavaScript or API function based on the quantity and list price. While you won't be doing all that in this practice, you will add a Table to display the Products. You first add a new Section to hold the table of products. You want the new section to be accessible via the right-hand side Vertical Anchor Navigator. You will add a new section using this component.

31. Select the **Vertical Anchor Navigator**.



32. In its **Property** pane, click the **+** to add a new **Section**. Name it ProductSelector and click **OK**.

The screenshot shows the Oracle ADF Vertical Anchor Navigator Properties pane. At the top, there is a navigation bar with tabs: General (selected), Sections (2), Events, and All. To the right of the navigation bar, a red box highlights the 'Properties' tab. Below the navigation bar, there are sections for 'ID' (with a text input field) and 'Section Fragment' (with a dropdown menu). The 'Display Logic' section contains a tree structure under a 'Case' node. The 'Case 1' node has a condition 'Always Show' and two sections: 'Section 1' and 'Section 2'. To the right of the 'Case 1' node is a red box highlighting a plus sign (+) button. A modal dialog box is open over the main pane, specifically over the 'Case 1' node. This dialog box has a title 'Case 1' and contains a 'Condition' field with a question mark icon, a 'fx (x)' button, and a 'New Section' button highlighted with a red box. Below this, there are sections for 'Selected Section' (containing 'page-header' and 'shippingaddress') and other sections like 'Section 1' and 'Section 2'.

The screenshot shows two overlapping windows. The top window is titled "Create Section" and contains fields for "Title" (ProductSelector), "ID" (productSelector), and "Description". The "Title" field is highlighted with a red border. The bottom window is titled "Case" and shows a hierarchical structure under "Case 1". It includes sections for "Condition" (Always Show), "Sections" (Section 1, Section 2, ProductSelector), and a "+" button for adding more sections.

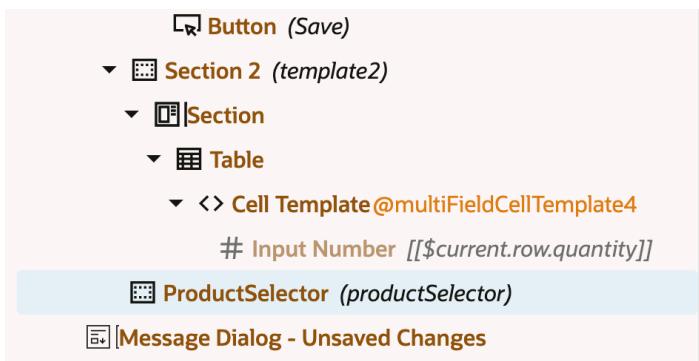
**Tip:** Did you notice the fragment you created earlier: **shippingaddress** was available to add as a section? This is another good use of fragments for leveraging repeatable code.

33. Select the new **ProductSelector** section and set its **Label Hint** to **Products**.

The screenshot shows the "Properties" panel for the "ProductSelector" section. It displays the following properties:

- ID: productSelector
- Title: ProductSelector
- Description: (empty)
- Label Hint: Products

Notice you don't see the new section because it is too low on the page and is hidden. When you test the page, you will see how it is displayed.



34. Click the **Data** tab and expand **Business Objects > rraProducts** and drag **Get Many** into the new **ProductSelector** Section. Render as a Table when prompted.

main-create-orders x

Page Designer Action Chains (10) Event Listeners (10) Events

Components Filter

Business Objects

- rraCustomers
- rraOrderLines
- rraOrders
- rraProducts

rraProducts

Get Many

GET /rraProducts

Service Business Object  
ID, Action Hint getall\_rraProducts getMany  
Description Get all

Vertical Anchor Navigator @main

Section 1 (template1)

- Single Select
- Section

  - Validation Group
  - Dynamic Form
  - Toolbar
  - Button (Save)

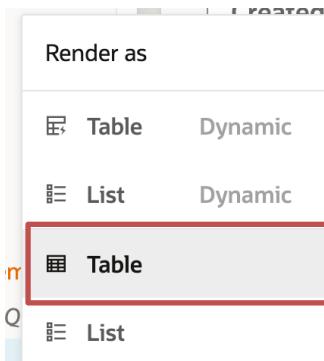
Section 2 (template2)

- Section
- Table
- Cell Template @multiFieldCellTemplate4

# Input Number [[\${current.row.quantity}]]

ProductSelector (productSelector)

Message Dialog - Unsaved Changes



35. Select the following fields in order: **productNumber**, **productName**, **unitPrice**, **imageUrl**. Set the component type for **ImageUrl** to **Image**. Click **Next** and **Finish**.

36. Select the **Image** component, click its **Data** tab, and change the **Source URL** to:
- ```
[[ $application.path + 'resources/' +$current.row.imageUrl ]]
```

37. Test this in **Live** Mode. You should be able to click the **Products** link on the right and the page will scroll to the list of products. You can also use the navigator to jump to the **Order Details** or **Order Line Items** sections.

**Hints:**

- If the page does not scroll to the different sections, click the **Reload Design Preview** icon.
- If you don't see **Jump to**, click the Actions icon or make the page wider.

The screenshot shows two views of the Oracle APEX application. The top view is the 'Design' mode, where a 'Jump to' menu is open, showing options like 'Order Details', 'Order Line Items', and 'Products'. The bottom view is the 'Live' mode, showing the 'New Order' page. In the 'Live' mode view, the 'Order Details' section is highlighted with a red box, and a similar 'Jump to' menu is visible on the right side of the page, mirroring the one in the Design mode.

**Note:** Notice how the page is responsive and displays the Anchor component or an action button as needed.

In a real app, you would add a row selection to the Products table to select a row in the table and use its values to populate the corresponding fields in a new row in the Order Lines table.

38. Return to Design mode.  
39. Close all tabs.

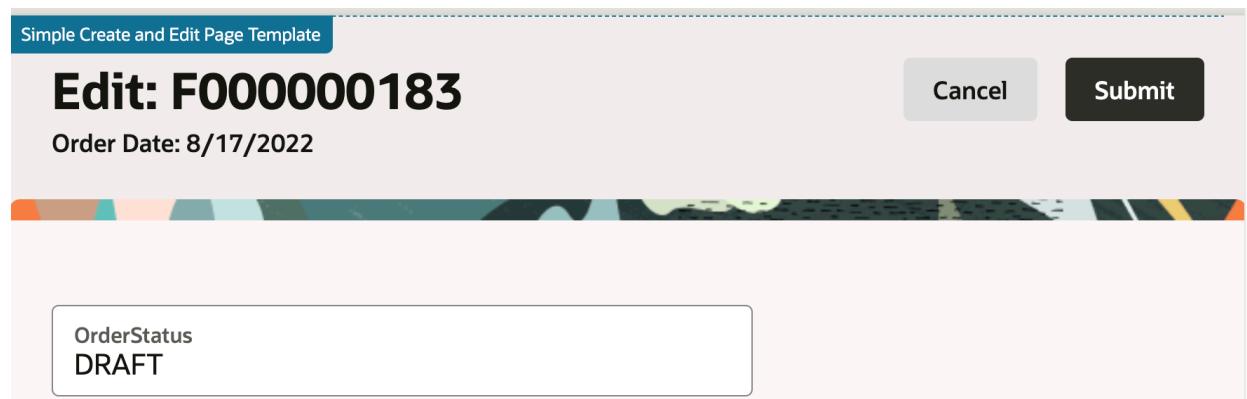
This completes this practice.

## Practice 4-2: Create the Edit Orders Page

### Overview

In this practice, you create the Edit Orders page, which accepts a passed order number, retrieves the order, and presents fields for edit and save. You will use the Simple Create and Edit Page Template as a basis for the Edit page.

Note the edit functionality backend is not implemented, so you're working with the frontend design only.



### Assumptions

You have successfully completed practice 3–1.

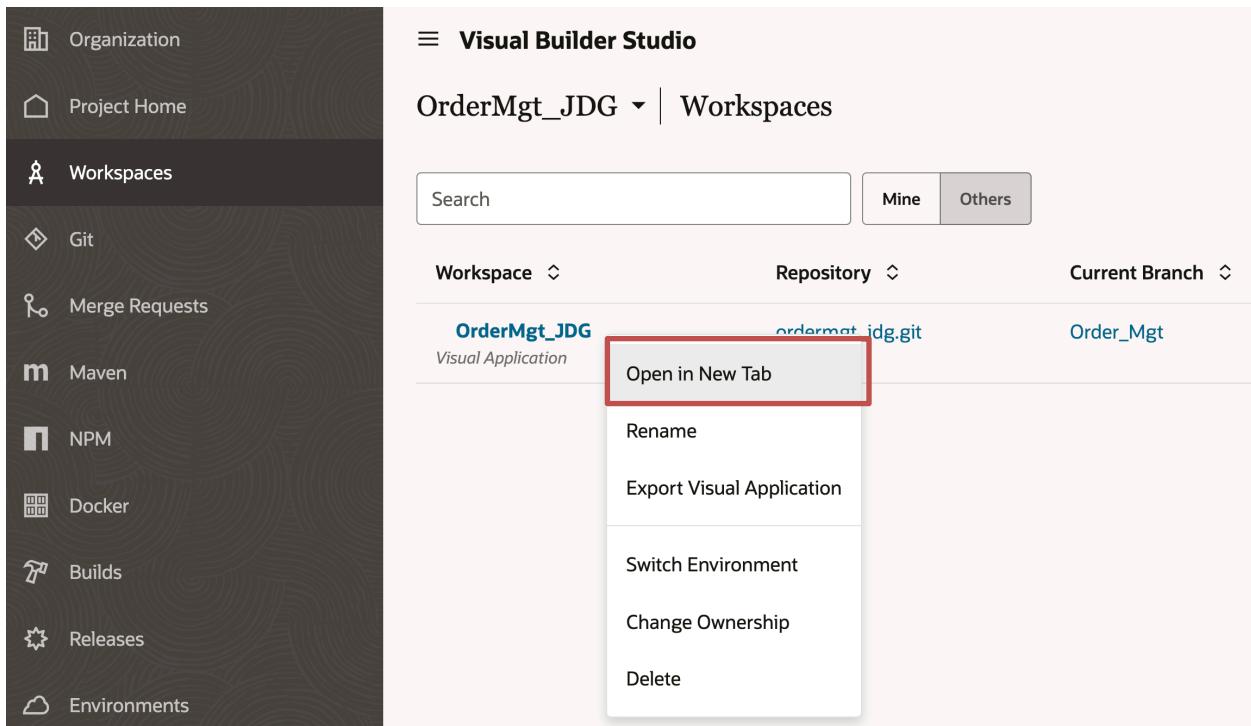
### Tasks

If Visual Builder Studio Designer is still open from the last lesson, skip to step 4.

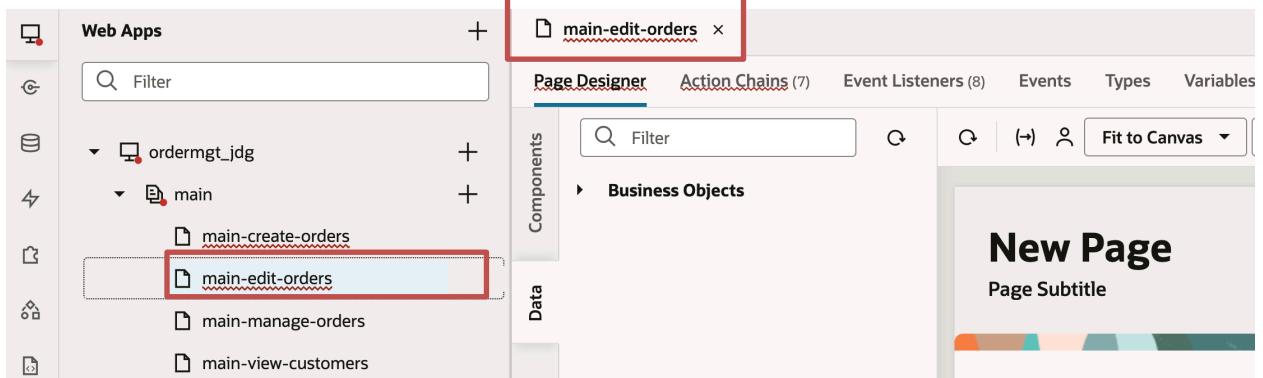
1. Log in to **Visual Builder Studio** and select your project name on the **Projects** tab. Your project name will be **OrderMgt\_< your initials >**, where **< your initials >** corresponds to your initials.

The screenshot shows the Visual Builder Studio interface. On the left is a sidebar with icons for Organization, Project Home, Workspaces, Git, Merge Requests, Maven, NPM, and Docker. The main area is titled "Visual Builder Studio" and "Organization". It has tabs for "Projects", "OCI Account", "Build Executor Templates", "Build Executors", "Component Exchange", and "Grid". Below the tabs is a search bar and filters for "Member", "Favorites", "Owner", "Shared", and "All". A "Projects: 3" section lists three projects: "OrderMgt\_JDG" (highlighted with a red box), "Order Management Applic...", and another partially visible project. The "OrderMgt\_JDG" row includes columns for "Name" (with a dropdown arrow), "Template" (with a dropdown arrow), "Favorite" (with a dropdown arrow), and "Status" (with a dropdown arrow). The "OrderMgt\_JDG" project is marked as "Current" and "Active".

2. The Project Home page opens. Right-click the **OrderMgt\_<your initials>** workspace and select **Open in New Tab**.



3. Close the **Getting Started** tab if it's open.  
 4. Select the **Web App** tab in the Navigator. Expand **ordermgt\_<your initials>**. Expand **main**. Select the **main-edit-orders** page. It opens in the Page Designer.



Similar to the Advanced Create and Edit Page Template, this has a dynamic form for its main section. This is where you choose the fields that will be included for edit. Notice the page comes with Cancel and Submit buttons and will have a Save button as well. The intent is that the Save button would allow the user to make changes to the editable fields and save without navigating off the page back to the parent or root page. The Submit button calls the Save action chain and then navigates back to the parent or root page. If you desire, you can have Submit be the only button and perform save and navigation at once and not have a separate Save button.

For example, in this application, the business rule is that orders cannot be edited once they're created. New orders are set to DRAFT order status, and so appear on the welcome page, and the edit will simply change the DRAFT order status to PROCESSING.

So, in this case, you would remove the Save button, but you won't do so in order to see a typical edit page.

Notice this page template comes with predefined actions, event listeners, and variables, and they will all be used to support the edit or create operations, depending on what is selected.

The screenshot shows the Oracle ADF Page Designer interface. At the top, there's a header with a file icon, the page name 'main-edit-orders', and a close button. Below the header, there are several tabs: 'Page Designer' (which is underlined in blue), 'Action Chains (7)', 'Event Listeners (8)', 'Events', 'Types', and 'Variables (11)'. The 'Variables' tab is highlighted with a red box.

5. Select the **Dynamic Form** in the **Structure** pane, click its **Quick Start** tab, and select **Configure as Edit Form**.

The screenshot shows the Oracle ADF Dynamic Form configuration dialog. At the top left, there's a back arrow and the text 'Dynamic Form'. Below that, there are tabs: 'General', 'Data', 'Events (1)', 'All', and 'Quick Start' (which is underlined in black).  
Under the 'Quick Start' tab, there are three options:

- Configure as Create Form**: Configure form component as a create form
- Configure as Edit Form**: Configure form component as an edit form (this option is highlighted with a red box)
- Configure as Detail Form**: Configure form component as a (read-only) detail form

6. Select **rraOrders** as the source of your data and click **Next**. On the **Select Save Data** step, click **Next** again. Notice you can choose a different endpoint to save the data to. In this case, you're using the same data source for both locating the data and saving it.

## Choose the source of your data

### Business Objects



rraCustomers



rraOrderLines



rraOrders



rraProducts



rraSuppliers

7. Enter `editOrder` for the **Label** and then click **Select fields to display >**.

Label \*

 ID \*

Description

Use Simple Layout

Select fields to display >

8. Remove all fields except **orderStatus**. Click the **X** at the end of the field to remove that field. Then, from the list of fields to the left, select **orderNumber** and **orderDate**. Click **Next**.

# objectVersionNumber

Field

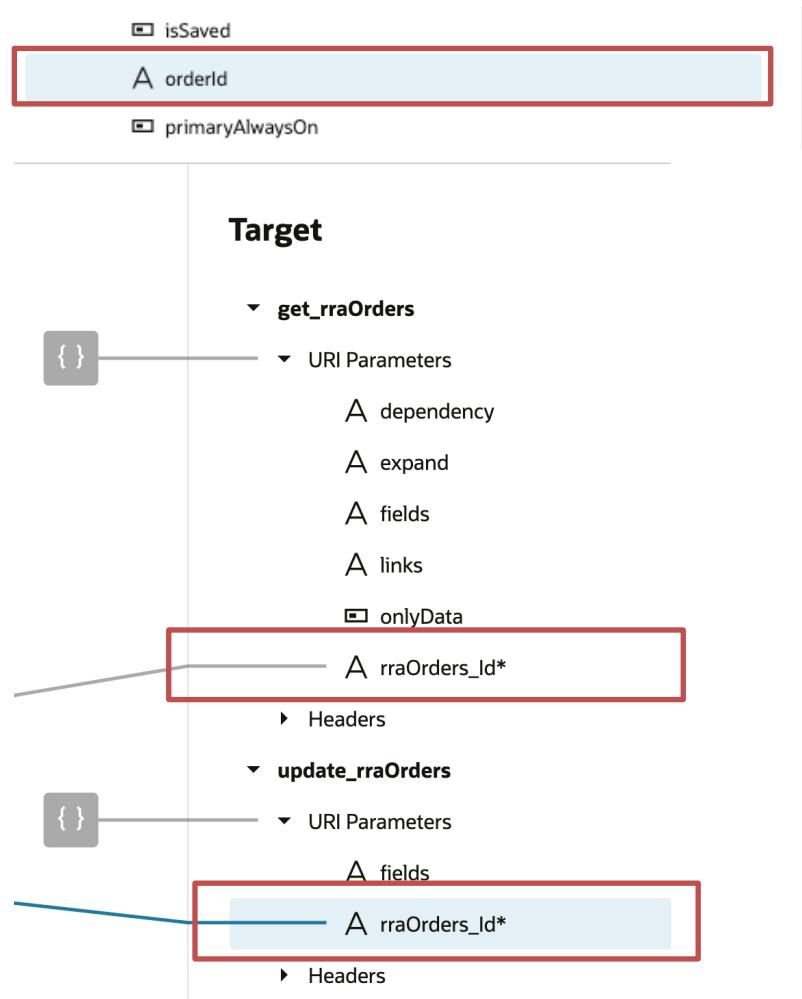
# objectVersionNumber1

Select fields to display [Use template instead](#) ↶

- ... A orderStatus
- ... A orderNumber
- ... A orderDate

9. Map **orderId** toTarget > **get\_rraOrders** > **rraOrders\_Id** (used to retrieve the record for editing) and **Target** > **update\_rraOrders** > **rraOrders\_Id** (used to perform the update) fields. Click **Finish**.

**Hint:** If you have difficulty dragging the **orderId** to the **rraOrders\_Id** fields, try making the page narrower.



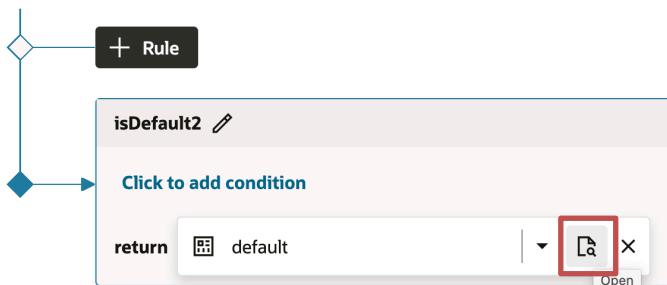
- The three fields you selected appear on the form and you see an error. The 500 means it cannot load the data because it does not have a valid orderId. Enter **Live** mode and click the **X** to dismiss the message and return to Design mode. Click the **General** tab for the dynamic form to make some changes to the form fields. Click the **Go to Rule Set** link to enter the **Rule Set** and layout.

The screenshot shows the Oracle Forms interface. At the top, there are tabs for 'Live', 'Design', and 'Code'. The 'Live' tab is active. Below the tabs, a message box displays an error: 'Could not load data: status 500'. In the center, a window titled 'New Page' is shown with a 'Page Subtitle'. On the right, there are 'Cancel' and 'Submit' buttons. A red box highlights the close button ('X') in the top right corner of the message box. Below the window, the 'Dynamic Form' properties panel is visible. It has tabs for 'General', 'Data', 'Events (1)', 'All', and 'Quick Start', with 'General' selected. The 'General' tab includes fields for 'ID' (containing 'oj-dynamic-form--346859652-1') and 'RuleSet' (containing 'editOrder'). To the right of the 'RuleSet' field is a 'Rule Sets' dropdown. A red box highlights the 'Go to Rule Set' link under the 'editOrder' section. A vertical 'Properties' sidebar is on the right.

**Note:** Your ID will be different.

- Click the **Open** icon for the layout.

#### Display Logic ?



12. You included the **orderNumber** and **orderDate** fields to use them to display in the form header but you don't want them to display as part of the form itself. Select each one, one at a time, and set the rule for when the field will appear. Click **orderNumber** first. Click the **Always** link under the **Show Field** property.

The screenshot shows the Oracle Forms builder interface. On the left, the 'Select fields to display' panel lists three fields: 'orderStatus', 'Field' (selected), and 'orderDate'. On the right, the 'Properties' panel shows the 'Show Field' property set to 'Always'. A red box highlights the 'orderNumber' field in the list.

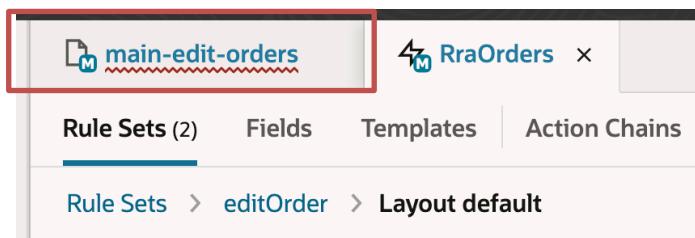
13. Change the operator to **==** and the attribute to **false** and click **Save**. Because true is not equal to false, the field will not appear on the form, but it is still available to reference in another part of the form.

The screenshot shows the 'Edit Show Field Condition' dialog box. It contains a single condition: 'if true == false'. The 'true' and 'false' fields are highlighted with a red box.

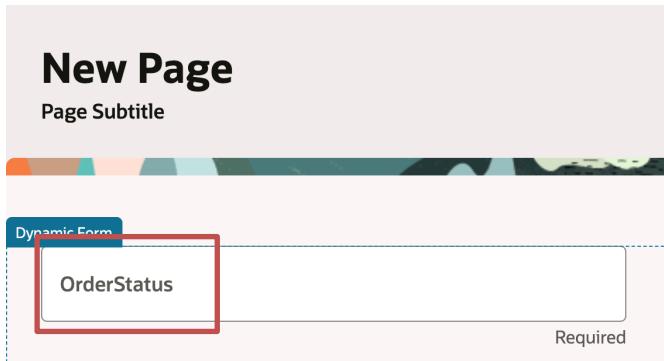
14. Repeat these steps with **orderDate**.

The screenshots show the Oracle Forms builder interface. The top screenshot shows the 'orderDate' field selected with its 'Show Field' property set to 'Always'. The bottom screenshot shows the 'orderDate' field selected with its 'Show Field' property set to a condition: 'if true strictly equals false'.

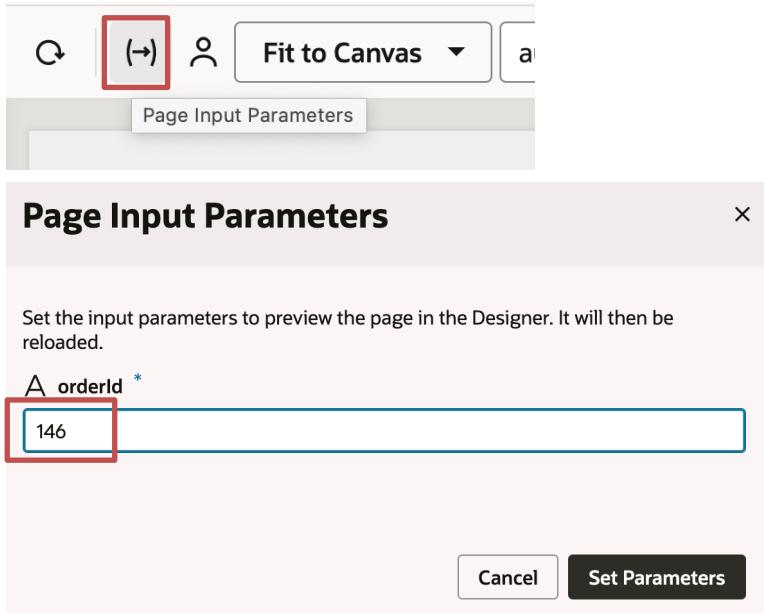
15. Click the **main-edit-orders** tab to return to the page. Dismiss the 500 error as you did before.



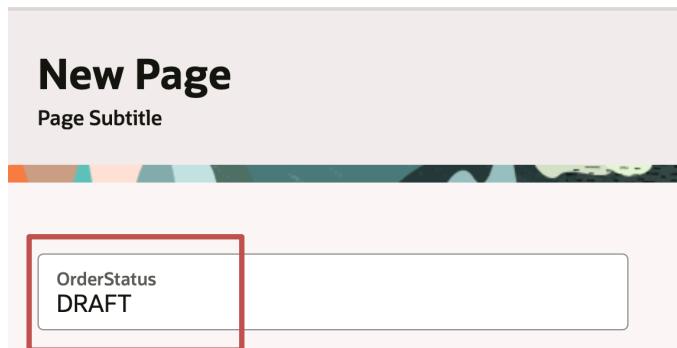
Notice only the OrderStatus field now appears.



16. Click the icon to enter Page Parameters. Enter a value of 146 for **orderId**.



17. The input parameter causes the orderId to refresh the page and retrieve the record and you'll see the value **DRAFT** appear in the **Order Status**.



The important functionality of doing the update for an edit is performed in action chains that are pre-created. How those work is beyond the scope of his course. You'll finish by configuring the page title and subtitle to display the order number and order date.

18. Select the **Simple Create and Edit Page Template**, and then enter the following code for the **Page Title**.

```
[[ 'Edit: ' + $variables.rraOrders.orderNumber ]]
```

|              |                                                                                                                                                                  |     |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| General      | Events (4)                                                                                                                                                       | All |
| ID           | oj_ssce1                                                                                                                                                         |     |
| Page Title * | <input style="border: 2px solid red; width: 100%;" type="text" value="[[ 'Edit: ' + \$variables.rraOrders.orderNumber ]]"/> <span>(?)</span> <span>fx (x)</span> |     |

19. Enter the following code for **Subtitle**:

```
[ [ 'Order Date: ' +
$flow.functions.formatDate($variables.rraOrders.orderDate) ]]
```

Page Title \*

[[ 'Edit: ' + \$variables.rraOrders.orderNumber ]]

Page Subtitle

[[ 'Order Date: ' + \$flow.functions.formatDate(\$variables.rraOrders.orderDate) ]]

The Page Title and Subtitle change to reflect the data from the retrieved record.

# Edit: F000000146

Order Date: 8/13/2022

OrderStatus  
DRAFT

20. Lastly, open the **main-manage-orders** page, select the List View and click the link for the **ListViewFirstSelectedItemChangeChain** event under the **Events** tab. Select the **toMainEditOrders** action and map the **orderId** parameter to the **\*rowKey**.

main-edit-orders RraOrders main-manage-orders

Page Designer Action Chains (7) Event Listeners (5) Events Types (4)

- Smart Search Page Template
  - Input Text @search {{\$variables.orderSearchString}}
- Collection Container
  - List View
    - Row Template @itemTemplate (itemTemplate)
      - List Item Template

**List View**

General Data **Events (1)** All Quick Start

first-selected-item → **ListViewFirstSelectedItemChangeChain**

**Event** Select

first-selected-item

**Action Chain** Select

**ListViewFirstSelectedItemChangeChain**

**Input Parameters** +

\* rowData \*  
{{ \$event.detail.value.data }}

\* rowKey \*  
{{ \$event.detail.value.key }}

**toMainEditOrders = Navigate To Page main-edit-orders**

Parameters

A orderId: ''

**Page \*** Create

main-edit-orders

Go to Page

Parameters

A orderId \*  
[[ rowKey ]]

**toMainEditOrders = Navigate To Page main-edit-orders**

Parameters

A orderId: rowKey

21. Return to the **Page Designer** to test that this works in **Live** mode. Select a record from the list of orders and the app will navigate to the **main-edit-orders** page and display the correct information.

**Note:** In the Redwood reference application, the Advanced Create and Edit Page Template was hard-coded to support both edits and creates on the same page, depending on whether an orderId value was passed into the page or the page was entered without an orderId. If an orderId is passed in, then it is assumed to be an edit operation and the edit sections of the page are displayed. If no orderId is passed to the page, it is assumed to be a create operation and the edit sections are hidden and create sections are displayed.

This is accomplished using many Bind If controls and it makes the page more complicated. By default, the Advanced Create and Edit Page Template and the Simple Create and Edit Page Template do not support using the same page for both create and edit operations. It is recommended to use a separate page for each operation as you've done here.

**Optional:** If you're curious, you may wish to examine the action chains and events that implement the Create and Edit page functionality. Examine the events and action chains for the Save button and Primary Action – the Submit button. Select the Events tab for the Simple Create and Edit Page Template and view the events and action chains that were created automatically as part of the page template, like the spCancelChain, spSaveChain, goToParent, and spSubmit chains. One of the benefits of using a page template is it comes preconfigured with events, buttons, and other features.

22. Close all open tabs.

This completes this practice.

