

Practice 3: Create Manage Orders Pages

Overview

In these practices, you will:

- Create the Manage Orders page to list and search existing orders
- Add list capability for actions and multiple columns
- Add navigation for View Orders and View Customers
- Add the View Orders page
- Add the View Customers page

Practice 3-1: Create the Manage Orders Page

Overview

In this practice, you create the manage-orders page functionality to manage the create, edit, and view detail functionality for orders. You configure the List Item Template to show multiple columns and actions.

The screenshot shows a page template for managing orders. At the top left is a "Smart Search Page Template" button. On the right is a "Create Order" button. The main title is "Manage Orders" with a dropdown arrow. Below it is a "Page Subtitle". A search bar contains the placeholder "Try a customer last name or order number". The page features a decorative banner at the bottom. The main content area displays three order items, each with a customer name, order ID, creation date, total amount, and a "DRAFT" status indicator. Each item also has a three-dot menu icon on the right.

Customer Name	Order ID	Creation Date	Total Amount	Status	Action
Kathryn Rogers	F000000313	8/7/2022	\$5,151.00	DRAFT	...
Kathryn Rogers	F000000314	8/11/2022	\$1,664.00	DRAFT	...
Kathryn Rogers	F000000315	8/21/2022	\$4,481.00	DRAFT	...

Assumptions

You have successfully completed Practice 2-1.

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 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". Below it is a navigation bar with tabs: Projects (which is selected), OCI Account, Build Executor Templates, Build Executors, Component Exchange, and Global. A search bar and filter buttons for Member, Favorites, Owner, Shared, All, and a magnifying glass icon are also present. The main content area shows "Projects: 3" and a table with columns for Name, Template, Favorite, and Status. The first row in the table is highlighted with a red box and shows "OrderMgt_JDG Current" under the Name column, "Order Management Applic..." under the Template column, a star icon under Favorite, and a green dot under Status.

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 with the "Workspaces" tab selected. The sidebar on the left is identical to the previous screenshot. The main area shows "OrderMgt_JDG ▾ | Workspaces". A search bar and filter buttons for Mine and Others are at the top. Below is a table with columns for Workspace, Repository, and Current Branch. The first row shows "OrderMgt_JDG" under Workspace, "ordermgt_jdg.git" under Repository, and "Order_Mgt" under Current Branch. A context menu is open over the "OrderMgt_JDG" workspace entry, with "Open in New Tab" highlighted by a red box. Other options in the menu include Rename, Export Visual Application, Switch Environment, Change Ownership, and Delete.

3. Close the **Getting Started** tab if it's open.

- Select the **Web Apps** tab in the Navigator. Expand **ordermgt_<your initials>**. Expand **main**. Select the **main-manage-orders** page. It opens in Page Designer.

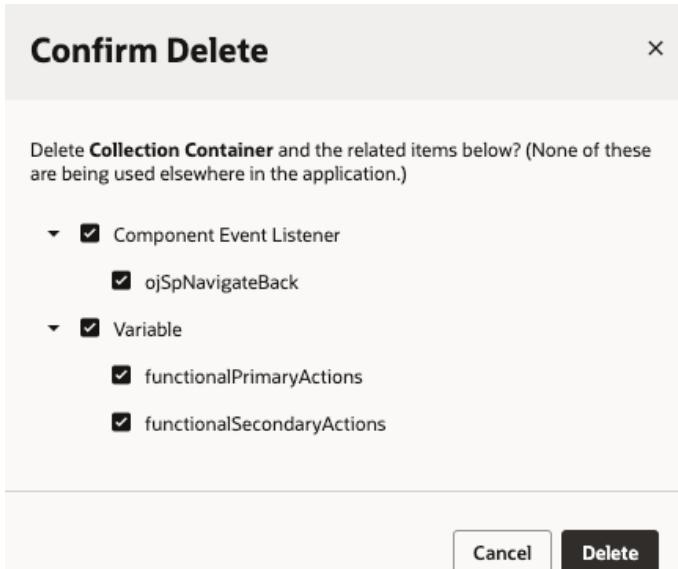
The screenshot shows the Oracle ADF Page Designer interface. On the left, the Navigator pane displays the page structure: 'Web Apps' is selected, followed by 'ordermgt_jdg', 'main', and finally 'main-manage-orders'. This last item is highlighted with a red box. The main workspace on the right is titled 'main-manage-orders' and shows the 'Page Designer' tab is active. The page content includes a title 'Manage Orders' and a 'Page Subtitle' section with a 'Search Slot' placeholder.

Note: The Smart Search Page Template is evolving and provides new structures and functionality when you create a page using this template. To better understand these new features, you will re-create these features manually. This will give you a better understanding of why the page template is structured this way and how to work with it.

When a page is created using this page template, it provides a complete page structure with a Collection Container to hold a collection of records retrieved from the datasource. These are presented in a list view. The row template and list item template are created for you, and you need to define the columns and actions. For this practice, you will delete these pre-created structures and re-create them manually to better understand how the page operates. Then, using the page in the future, you can leverage the pre-existing structures more easily.

- Delete the Collection Container and all its children from the Structure pane. Right-click **Collection Container** and select **Delete**. Click the **Delete** button to confirm the delete.

The screenshot shows the Oracle ADF Structure pane. A context menu is open over a 'Collection Container' node, which is highlighted with a red box. The menu options include 'Select Parent', 'Surround', 'API Doc', 'Insert Component', 'Cut', 'Copy', 'Paste', and 'Delete'. The 'Delete' option is also highlighted with a red box at the bottom of the menu.



- Add an **Input Text** component to the Smart Search Page Template's **Search Slot**. Select it and change its **Label Hint** property to Try a customer last name or order number.

The screenshot shows the Oracle ADF Page Designer interface. The page title is 'main-manage-orders'. The 'Page Designer' tab is selected. In the 'Components' panel, under 'Forms & Inputs', the 'Input Text' component is selected and highlighted with a red box. A vertical red arrow points from this component down to the 'Smart Search Page Template' in the 'Search Exchange' panel below. The 'Search Exchange' panel also has a red box around it. The 'Smart Search Page Template' panel contains an 'Input Text' component, which is also highlighted with a red box. The 'Data' panel on the left is empty. A note at the bottom says 'You can also try searching for "input text" at Component Exchange...' and a 'Search Exchange' button is present.

Search Exchange

Smart Search Page Template | Slots

Select a Slot:

- Default
- Search Slot**

Input Text

General Data Events All

ID

Label Hint (?) fx (x)

Try a customer last name or order number

(x) String not externalized for translation.

- Click the Input Text component's **Data** tab and click the select variable icon (x) and then click the **Create** link and bind a new Page variable: `orderSearchString` of type String.

Input Text

General **Data** Events All

Value (?) fx (x)

Variables

Filter

Page Create

Variables

allFilters Edit

< Input Text

General Data Events All

Value

Create Variable

Variable Constant

ID *

Type * ▾

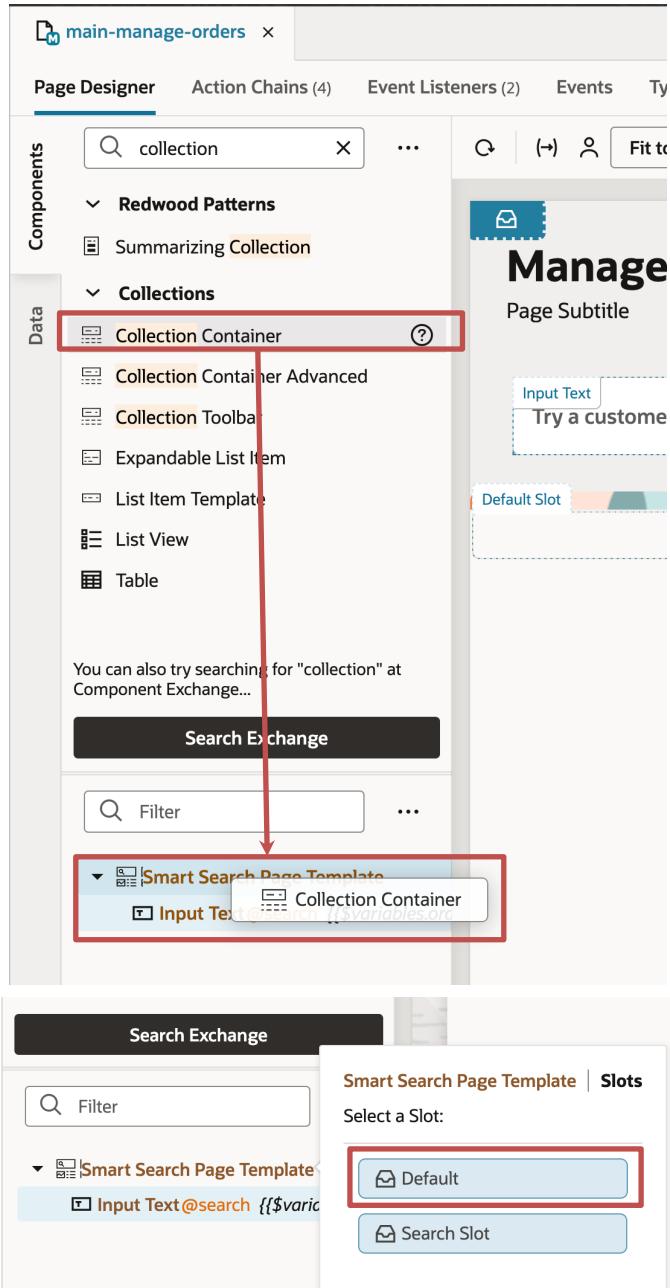
Input Parameter

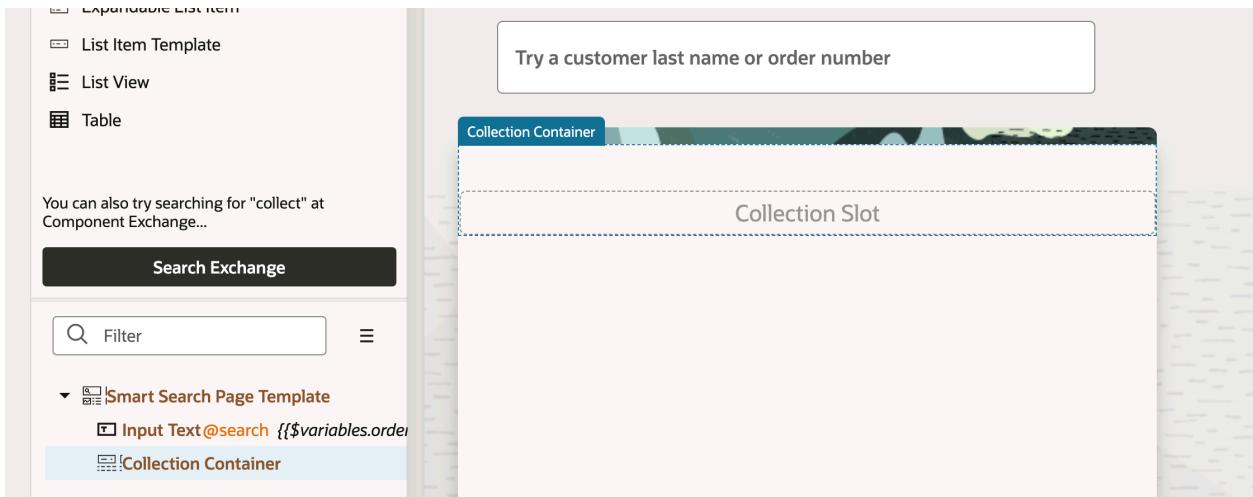
Cancel **Create**

Value

The screenshot shows the Oracle ADF Faces InputText component configuration dialog. The 'Data' tab is selected. A 'Create Variable' modal is open, showing 'orderSearchString' as the ID and 'String' as the type. The 'Value' field in the main dialog contains the expression {{ \$variables.orderSearchString }}.

8. Drag a Collection Container component onto the Smart Search Page Template Default slot.





9. Click the **Data** tab, expand **Business Objects > rraOrders**, and drag **Get Many** to the **Collection Slot** on the canvas. Select **Render as List** when prompted. The Add Data dialog opens.

The screenshot shows the Oracle ADF Page Designer interface. The top navigation bar includes 'main-manage-orders', 'Page Designer', 'Action Chains (4)', 'Event Listeners (2)', 'Events', 'Types (3)', 'Variables (20)', 'JavaScript', 'JSON', and 'Settings'. The left sidebar has 'Components' and 'Data' tabs, with 'Data' selected and highlighted with a red box. Under 'Business Objects', 'rraOrders' is selected and highlighted with a red box. A 'Get Many' operation is listed under rraOrders, with a red arrow pointing from the 'List' option in the 'Render as' dropdown to the 'Collection Slot' on the canvas. The 'Collection Slot' is a dashed rectangular area on the canvas where the 'Get Many' operation is being dropped. The 'Render as' dropdown menu shows 'Table' (Dynamic), 'List' (Dynamic), 'Table', and 'List', with 'List' highlighted with a red box.

10. Click **Next** to select fields for the list. Select **customerFirstName**, **customerLastName**, **orderDate**, **orderNumber**, **orderStatus**, and **totalAmount** and click **Next**.

Item Template Fields

You can put multiple fields into a single spot. You can leave any of the spots empty - the template will adapt.

Slot	Value	Type
Leading Slot	A customerFirstName X	Text
Overline slot	A customerLastName X	Text
Default slot	A orderDate X	Text
Secondary slot	A orderNumber X	Text
Tertiary slot	A orderStatus X	Text
Metadata slot	# totalAmount X	Text

11. Under **Target**, click **filterCriterion** to filter for the search string. Click the **Click to add condition** link.

Target

- ▼ getall_rraOrders
 - ▶ { } uriParameters
 - ▶ { } filterCriterion

{ } filterCriterion [Builder](#) [Code](#)

Include items Match All Match Any

[Click to add condition](#)

12. Filter for **orderNumber** or **customerLastname** to match the **orderSearchString**.

- Select **orderNumber** for the first attribute.

The screenshot shows the 'filterCriterion' builder interface. At the top, there are tabs for '{ } filterCriterion', 'Builder' (which is selected), and 'Code'. Below the tabs, there is an 'Include items' section with two radio buttons: 'Match All' (selected) and 'Match Any'. Underneath this, the word 'IF' is followed by a text input field containing 'orderNumber'.

- Leave the operator set to contains (\$co).

Note: You can select other operators like **starts with**, **equals**, **ends with**, **greater than**, or **less than** depending on what you're comparing.

The screenshot shows the 'filterCriterion' builder interface. A dropdown menu is open, showing the option 'contains (\$co)'.

- Set the second attribute to **\$variables.orderSearchString**.

The screenshot shows the 'filterCriterion' builder interface. The 'contains (\$co)' dropdown has been replaced by a text input field containing '\$variables.orderSearchString', which is highlighted with a red border.

The screenshot shows the 'filterCriterion' builder interface. Both the 'orderNumber' condition and the '\$variables.orderSearchString' condition are highlighted with red boxes.

13. Click **Match Any** (OR condition) and click the **Add Condition** link.

The screenshot shows the 'filterCriterion' builder interface. The 'Match Any' radio button is selected and highlighted with a red box. Below it, there is a 'IF' section with 'orderNumber' and a 'Condition' section with 'Add Condition' and 'Add Group' buttons, where 'Add Condition' is also highlighted with a red box.

14. Repeat the sequence: choose **customerLastName** for the first attribute, **contains (\$co)** for the operator and **\$variables.orderSearchString** for the second attribute. Click **Done** and **Finish**.

The screenshot shows the Oracle ADF Filter Criterion Builder interface. It displays two separate filter criteria, each consisting of an IF condition followed by an OR clause.

- Filter Criterion 1:**
 - IF** `orderNumber contains $variables.orderSearchString`
 - OR** `customerLastName contains ($co) $variables.orderSearchString`
- Filter Criterion 2:**
 - IF** `orderNumber contains $variables.orderSearchString`
 - OR** `customerLastName contains $variables.orderSearchString`

15. Test that it works. Click the **Live** button and enter Robinson in the search field and press **Enter**.

The screenshot shows the "Manage Orders" page in live mode. The search input field contains "Robinson". The results table displays three rows of data, each representing an order:

Customer Name	Order Date	Order ID	Status	Total Amount
Norma	2022-08-13T15:43:33+00:00	F000000146	DRAFT	6895
Norma	2022-08-17T15:43:33+00:00	F000000147	DRAFT	8006
Norma	2022-08-20T15:43:33+00:00	F000000148	DRAFT	1953

16. Clear the Search field and enter 146.

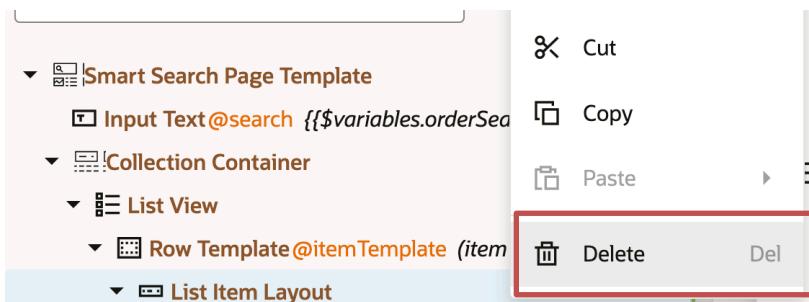
The screenshot shows the 'Manage Orders' page. At the top, there is a search bar with the placeholder 'Try a customer last name or order number'. Below the search bar, the order details for order number 146 are displayed. The details include the customer's name (Norma), the order date (2022-08-13T15:43:33+00:00), the order ID (F000000146), and the status (DRAFT). The entire search bar area is highlighted with a red box.

17. Click the **Design** button to return to Design mode.

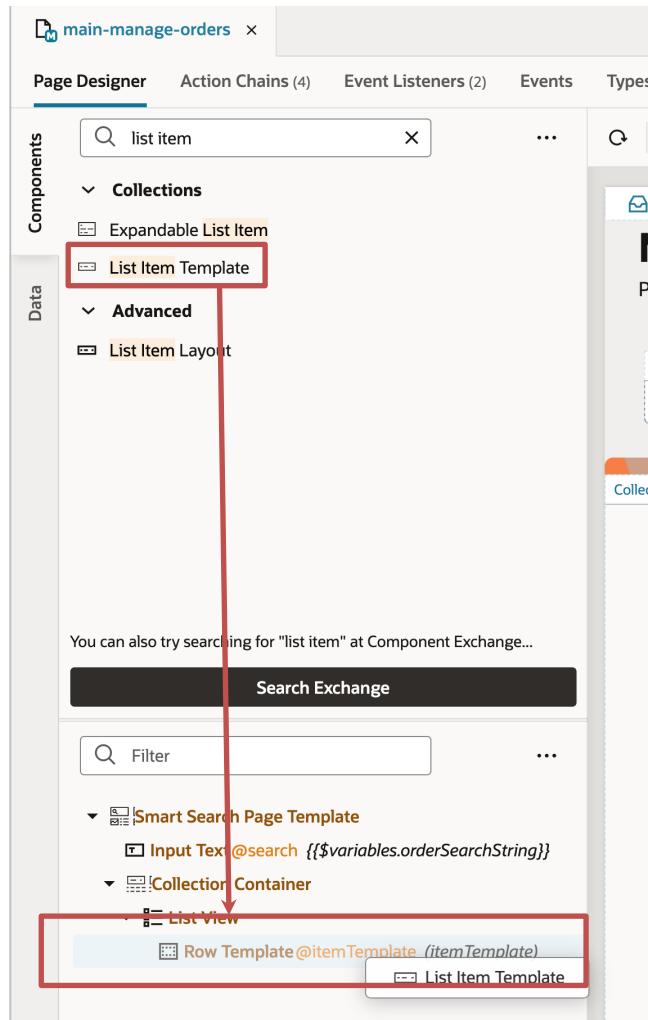


Configure the list to include separate columns, formatting, and actions.

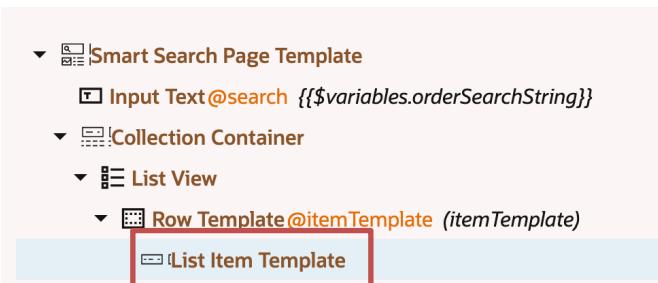
18. Select the **List Item Layout** in the **Structure** pane and right-click it to delete it.



19. Drop a **List Item Template** component onto the **Row Template**.



20. With the **List Item Template** selected, click the > for the **Badge** property.



The screenshot shows the 'List Item Template' configuration screen. The 'General' tab is selected. The 'ID' field is empty. The 'Column Layout' dropdown is set to 'auto'. To the right, there is a vertical 'Properties' panel.

21. For the **Text** property, bind **orderStatus**.

The screenshot shows the 'Badge' properties panel. The 'Text' field contains the expression `[[$current.data.orderStatus]]`, with a red box highlighting the 'fx (x)' button. Below it, the 'Variables' list shows various variables under the 'Current' scope, with 'orderStatus' highlighted by a red box. The 'Properties' panel is visible on the right.

22. Set the badge to display differently based on **orderStatus**. Enter the following expression for **Status** to test DRAFT:

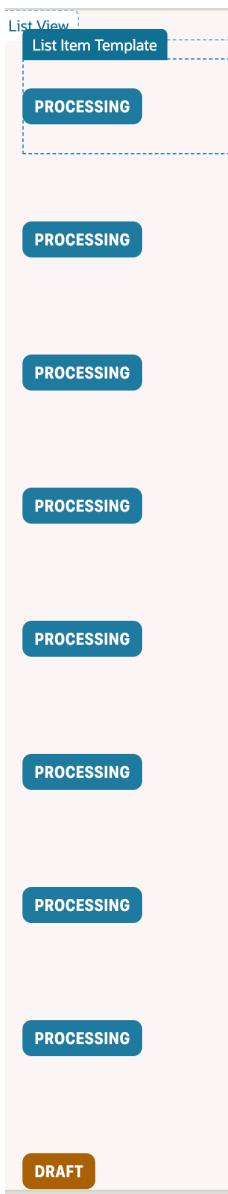
Note: The badge text will not display until you complete configuring its properties.

```
[[ $current.data.orderStatus === 'DRAFT' ? 'warning' : 'info' ]]
```

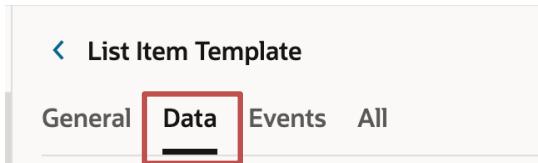
The screenshot shows the 'Badge' properties panel again. The 'Text' field contains the expression `[[$current.data.orderStatus]]`. The 'Status' field contains the conditional expression `[[$current.data.orderStatus === 'DRAFT' ? 'warning' : 'info']]`, which is highlighted with a red box. The 'Properties' panel is visible on the right.

23. For **Position**, select **Leading**.
24. For **Size**, select **Medium**.
25. Click the < **Badge** link to return to the **General** properties.

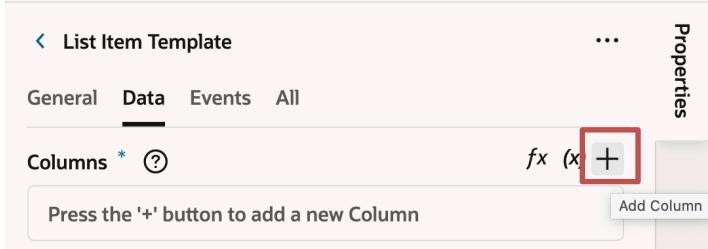
The screenshot shows the Oracle ADF Faces configuration interface. On the left, there is a navigation bar with a back arrow labeled '< Badge' and three dots (...). To the right of the navigation bar is a vertical sidebar labeled 'Properties'. The main area contains four configuration sections: 'Text' (with the value '[[\$current.data.orderStatus]]', 'Status' (with the value '[[\$current.data.orderStatus === \'DRAFT\' ? \'warning\' : \'info\']]'), 'Position' (set to 'Leading'), and 'Size' (set to 'Medium'). A red rectangular box highlights the 'Text', 'Status', 'Position', and 'Size' sections.



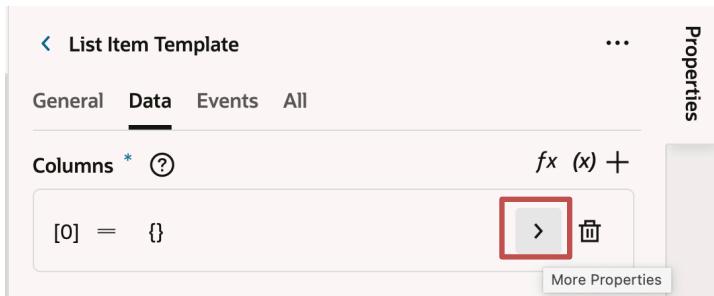
26. Click the **Data** tab to enter the rest of the field values.



27. Click the **+ Add Column** icon for the Columns property to add a new column.



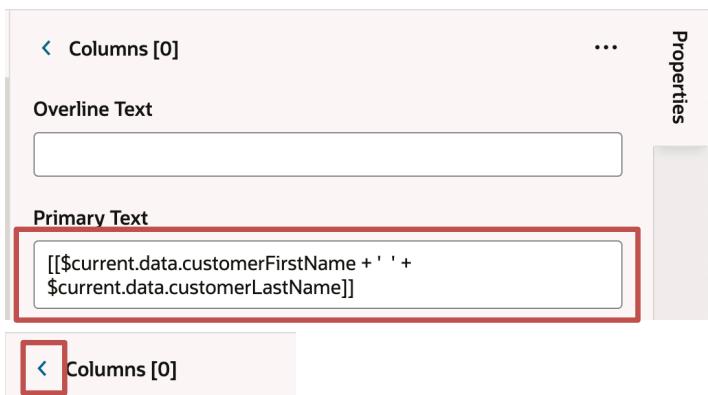
28. Click the **> More Properties** icon to define the first column.



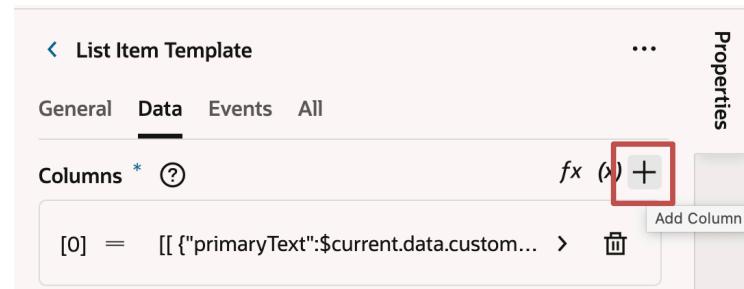
29. For **Primary Text**, enter:

```
[$current.data.customerFirstName + ' ' +
$current.data.customerLastName]
```

30. When done, click the **< Columns [0]** link to return to the Columns property.



31. Click the **+ Add Column** icon for the Columns property to add another column.



32. Click the **> More Properties** icon to define the next column.

General Data Events All

Columns *

[0] = [[{"primaryText":\$current.data.custom... >]

[1] = {}

33. For **Primary Text**, enter:

```
[$current.data.orderNumber]
```

34. For **Secondary Text**, enter:

```
[$flow.functions.formatDate($current.data.orderDate)]
```

35. When done, click the **< Columns [1]** link to return to the Columns property.

Columns [1] ... Properties

Overline Text

Primary Text

[[\${current.data.orderNumber}]]

Secondary Text

[[\${flow.functions.formatDate(\${current.data.orderDate})}]]

36. Click the **+ Add Column** icon for the Columns property to add another column. Click the **> More Properties** icon to define the next column.

General Data Events All

Columns *

[0] = [[{"primaryText":\$current.data.custom... >]

[1] = [[{"primaryText":\$current.data.orderN... >]

Add Column

37. For **Primary Text**, enter:

```
[$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode  
, $current.data.totalAmount ? $current.data.totalAmount : 0)]
```

List Item Template			
DRAFT	Kathryn Rogers	F000000313 8/7/2022	\$5,151.00
DRAFT	Kathryn Rogers	F000000314 8/11/2022	\$1,664.00
DRAFT	Kathryn Rogers	F000000315 8/21/2022	\$4,481.00
PROCESSING	Dennis Lopez	F000000316 5/24/2022	\$4,682.00

Hint: If the page does not display horizontally as shown, try toggling the Navigator panel tab closed. Also, if the formatted total amount does not display, this will be resolved next.

- When done, click the < **Columns [2]** link to return to the Columns property.

To ensure all columns display, you need to set the **Column Layout**, which you do next.

39. Click the **General** tab.

The screenshot shows the 'List Item Template' configuration interface. The 'General' tab is selected, highlighted with a red border. Below it are tabs for 'Data', 'Events', and 'All'. On the right, a vertical 'Properties' panel is visible. Under the 'General' tab, there's a section titled 'Columns *' containing three items labeled [0], [1], and [2], each with a code snippet and a trash icon. A plus sign (+) is located to the right of the list.

40. Set **List Item Template Column Layout** to: **3 columns 1/4 1/2 1/4**.

The screenshot shows a dropdown menu for 'Column Layout'. The current selection is 'auto', which is highlighted with a blue border. Other options listed include: '1 column' (one), '2 columns 1/4 3/4' (two-split3-9), '2 columns 3/4 1/4' (two-split9-3), '2 columns 1/2 1/2' (two-split6-6), '3 columns 1/4 1/4 1/2' (three-split3-3-6), '3 columns 1/4 1/2 1/4' (the option selected, highlighted with a red border), '3 columns 1/2 1/4 1/4' (three-split6-3-3), and '3 columns 1/3 1/3 1/3' (three-split4-4-4).

The screenshot shows the 'List Item Template' interface displaying three rows of data. Each row consists of three columns: a status button ('PROCESSING'), a name ('Judy Evans'), and a reference number ('F000000170'). To the right of the third column, there is a large red-bordered box containing the values '\$8,126.00', '\$12,217.00', and '\$4,788.00' respectively. The date '6/5/2022' is also visible in the third column of the first two rows.

41. Select **Smart Search Page Template** in the Structure pane. For **Primary Action Label**, enter: Create Order. The Primary action button appears.

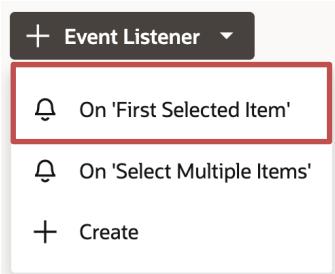
The screenshot shows the Oracle ADF Faces Designer interface. At the top, there's a 'Primary Action' dialog with a 'Primary Action Label' field containing 'Create Order'. Below it, a larger panel displays a 'Create Order' button. The background shows the 'Structure' pane with a tree view of the page template components.

42. Select the **List View** and set its **ID** to `orderList`. Set the **Selection Mode** to **Single**.

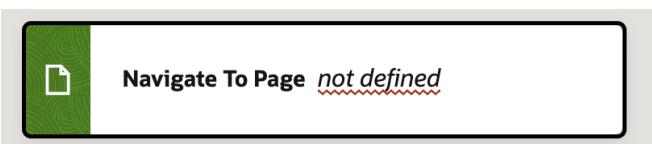
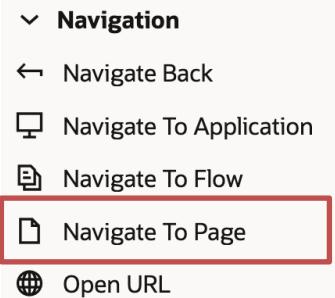
The screenshot shows the Oracle ADF Faces Designer interface. It displays the 'Properties' dialog for a 'List View' component. In the 'General' tab, the 'ID' is set to 'orderList'. In the 'Events' tab, the 'Selection Mode' is set to 'Single'. The 'Structure' pane shows the page template with the 'List View' component selected.

43. Set up navigation to an edit page to edit a selected order. Select the **List View's Events** tab. Add an **On 'First Selected Item'** event listener. The action chain opens.

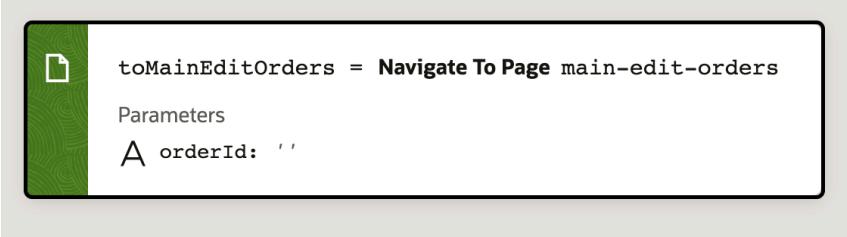
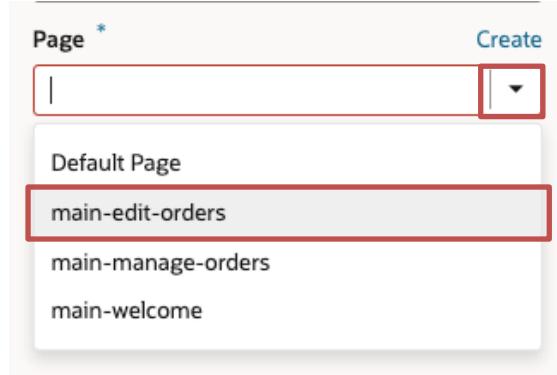
The screenshot shows the Oracle ADF Faces Designer interface. It displays the 'Properties' dialog for a 'List View' component. The 'Events' tab is selected. The tabs at the bottom are 'General', 'Data', 'Events' (selected), 'All', and 'Quick Start'.



44. Add a **Navigate to Page** action to the canvas.



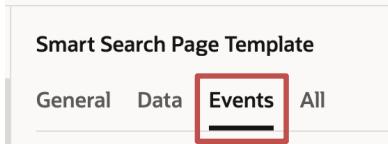
45. With the new action selected, select **main-edit-orders** for the page.



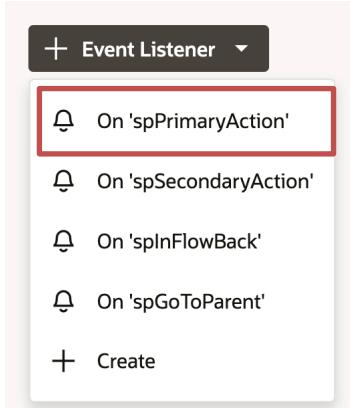
Note: The orderId has not been bound to a variable yet. You will do that later.

46. Click the **Page Designer** tab.

47. Select the **Smart Search Page Template**. Click its **Events** tab.



48. Add an **On 'spPrimaryAction'** event listener. The new action chain opens.



49. Add a **Navigate to Page** action to the canvas. Click the **Create** link to create a new page named `main-create-orders`. Use the **Advanced Create and Edit Page Template**.

The screenshot shows the Oracle ADF interface. At the top, there is a 'Navigate To Page' action on the canvas. Below it, a 'Create' link is highlighted with a red border. A modal dialog titled 'Create Page' is open, showing a 'Page ID' field containing 'main-create-orders', which is also highlighted with a red border. The 'Choose Page Content' section contains a 'Filter' input field. Two buttons are visible: 'Create an Empty Page' (with a plus icon) and 'Advanced Create and Edit Page Temp...' (with a document icon). The 'Advanced Create and Edit Page Temp...' button has a blue border around it.

Page *

Create

Go to Page

50. Click the **Page Designer** tab.
51. Test the Navigation works. Enter **Live** mode and click the **Create Order** button. Close the **main-create-orders** page after it opens.

The screenshot shows the Oracle Redwood Page Designer interface. At the top, there are tabs: 'Page Designer' (which is selected), 'Action Chains (9)', 'Event Listeners (10)', 'Events', 'Types', 'Variables (13)', and 'JavaScript'. Below the tabs is a search bar with placeholder text 'Filter, Alt+F' and a '... More' button. To the right of the search bar are icons for refresh, undo, redo, and a user profile, followed by a 'Fit to Canvas' button.

The left sidebar has several sections with icons: 'Components' (selected), 'Data' (selected), 'Actions', 'Style', and 'Behaviors'. Under 'Components', there is a collapsed section 'Redwood Patterns' and a expanded section 'Data' containing the following items:

- Bottom Drawer Template
- Calendar
- Card
- Completion Panel
- Create Edit Drawer Template
- Detail Panel
- Diagram Builder
- Diagram Node
- Empty State
- General Drawer Template
- Hierarchy Card
- Hierarchy Card Info
- Hierarchy Viewer Pattern

The main canvas area displays the 'main-create-orders' page in 'Live' mode. The page title is 'New Page' with a subtitle 'Page Subtitle'. It features two sections: 'Sample Section 1' and 'Sample Section 2', each with a dashed border indicating they are interactive components.

52. Return to the main-manage-orders page. Select a row and the **main-edit-order** page opens. Close the **main-edit-orders** tab.

	Judy Evans	F000000170 6/5/2022	\$8,126.00
PROCESSING	Judy Evans	F000000171 6/8/2022	\$12,217.00

53. Enter **Design** mode on the **main-manage-orders** page.
 54. Add secondary actions for the **List Item Template** to view orders and customers. Select the **List Item Template**. Click the **General** tab. Click the **+ Add Action** icon for **Secondary Actions**.

Primary Action >

Secondary Actions ⓘ fx (x) +

Press the '+' button to add a new Action

Add Action

55. For [0] click > to enter **More Properties**.

Id	Label
[0] =	>

56. Set the following properties:

- Action Id:** order
- Label:** View Order

- **Icon:** oj-ux-ico-file-view-details

Manage Orders ▼

Page Subtitle

Try a customer last name or order number

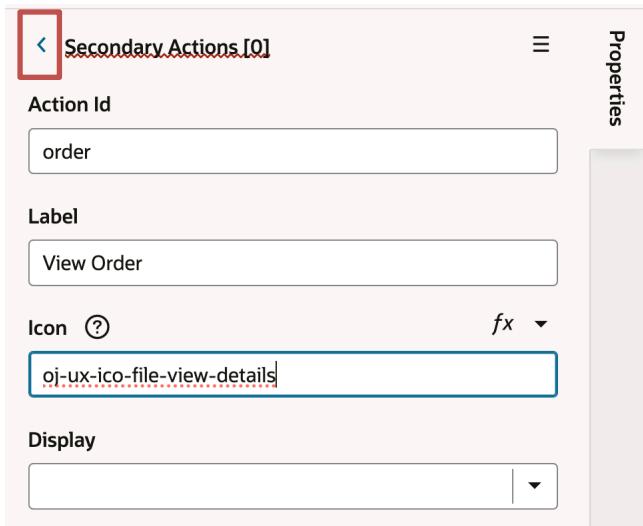
List Item Template

DRAFT	Kathryn Rogers	F000000313 8/7/2022	\$5,151.00	⋮
DRAFT	Kathryn Rogers	F000000314 8/11/2022	\$1,664.00	⋮

Note: More Redwood icons are available here:

<https://static.oracle.com/cdn/fnd/gallery/2410.0.3/images/preview/index.html>

57. Click < Secondary Actions[0].



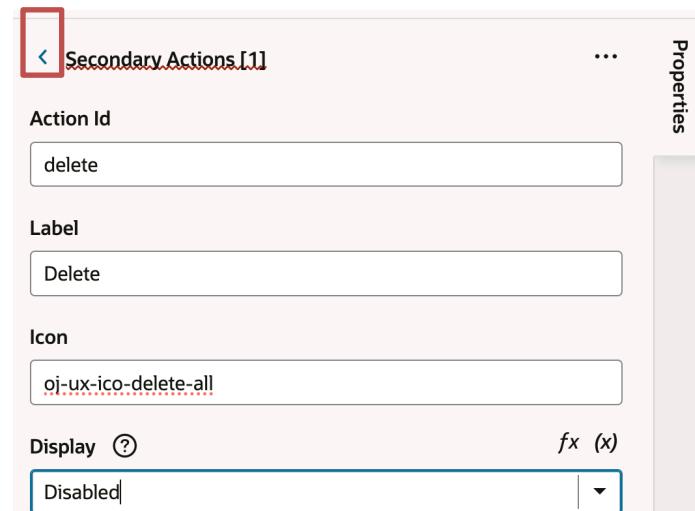
58. Click + to add another Secondary Actions [0].

59. For [1] click > to enter More Properties:



- **Action Id:** delete
- **Label:** Delete
- **Icon:** oj-ux-ico-delete-all

60. Set the Display property to Disabled. You will not implement delete functionality in this app.



61. Click < Secondary Actions [1].

62. Click + to add another **Secondary Action**.

63. For [2] click > to enter More Properties.

The screenshot shows a table titled "Secondary Actions". It has two columns: "Id" and "Label". Row [0] contains "order" and "View Order". Row [1] contains "delete" and "Delete". Row [2] is empty. To the right of the table is a toolbar with a "fx (x)" button and a red-bordered "+" button. Below the table is a "More Properties" button. The row for index [2] has a red border around its "More Properties" button.

Id	Label
[0] = order	View Order
[1] = delete	Delete
[2] =	

More Properties

- **Action Id:** customer
- **Label:** View Customer
- **Icon:** oj-ux-ico-customer-blog

64. Click < Secondary Actions [2].

The screenshot shows a "Properties" dialog for "Secondary Actions [2]". It has sections for "Action Id" (customer), "Label" (View Customer), "Icon" (oj-ux-ico-customer-blog), and "Display". A red box highlights the title bar of the dialog.

Properties

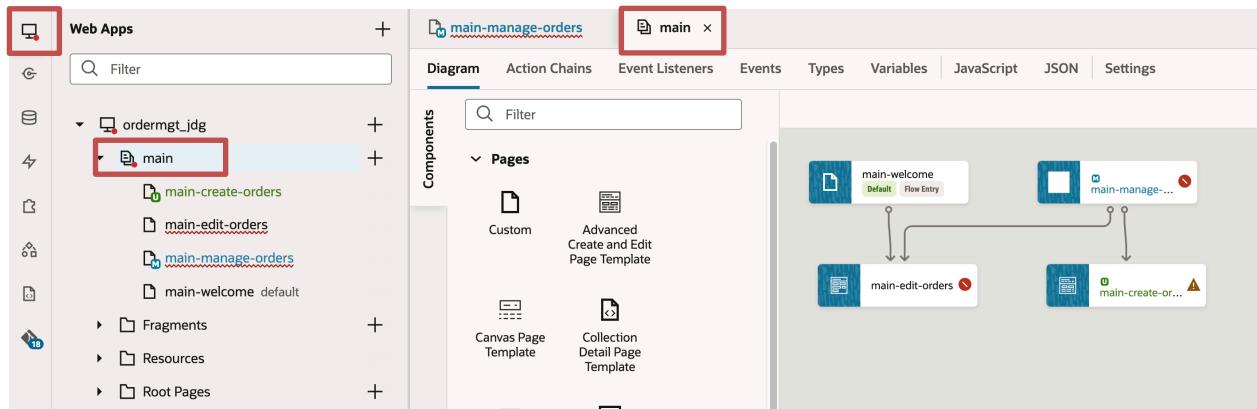
Action Id	customer
Label	View Customer
Icon	oj-ux-ico-customer-blog
Display	...

Notice the two icons were replaced with an action (...) icon. The page is responsive to how many actions are defined and automatically changes to display accordingly.

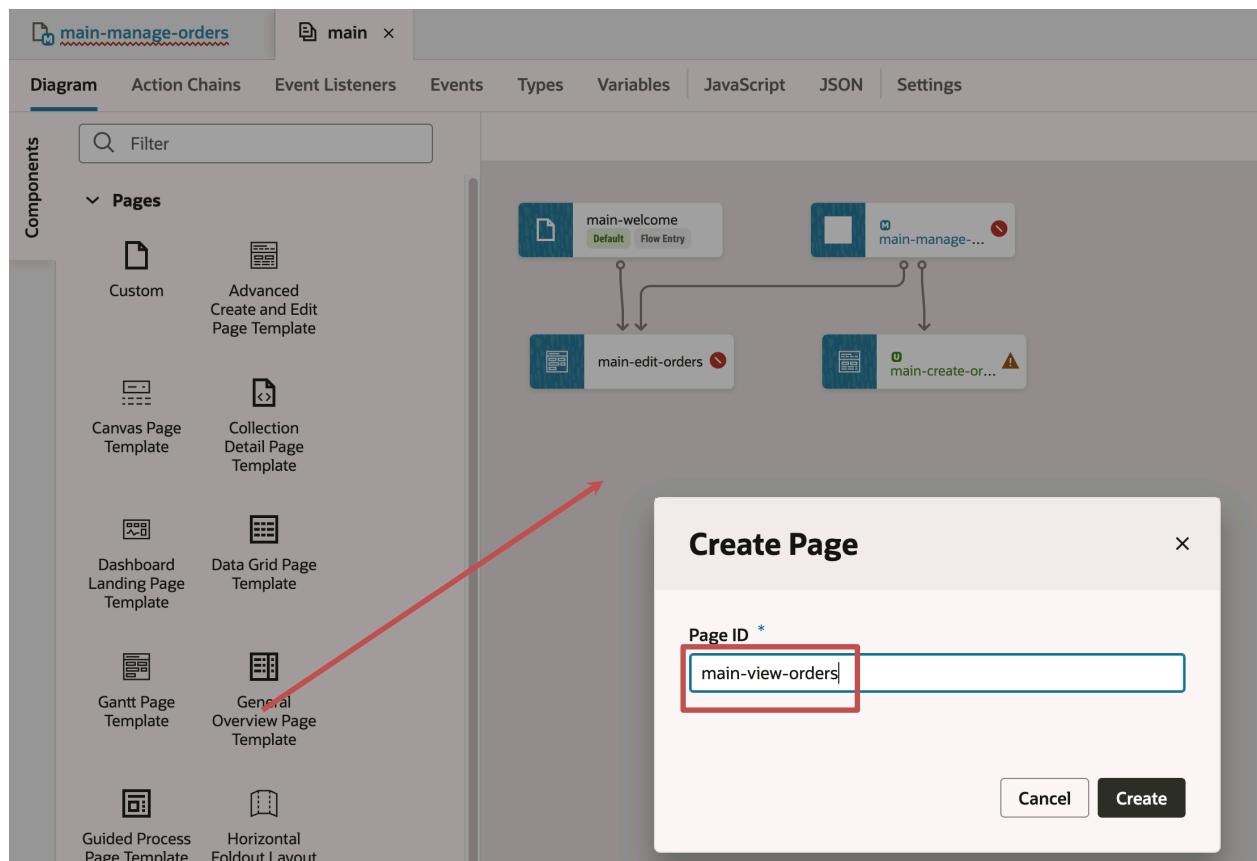
65. Enter **Live** mode and test that the ... button shows the three actions in the pop-up menu. Return to **Design** mode when done.

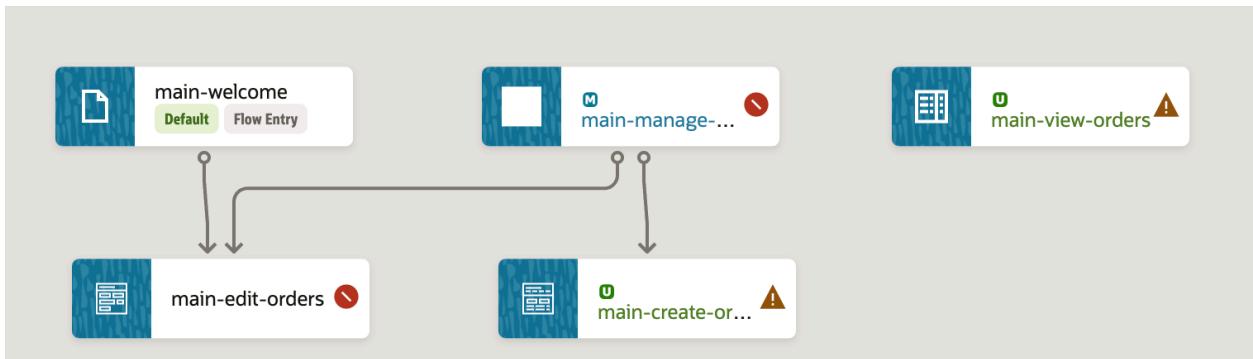
The screenshot shows a "Live" view of a page. A context menu is open at a specific location, featuring three items: "View Order", "Delete", and "View Customer". The "View Order" item has a small icon next to it. A red box highlights the "..." button on the left side of the menu, which is used to expand the list. The menu is displayed over a light gray background.

66. Use the main flow diagram to add the remaining pages. Click the **main** flow in the Web Apps pane. Click the **Diagram** tab.



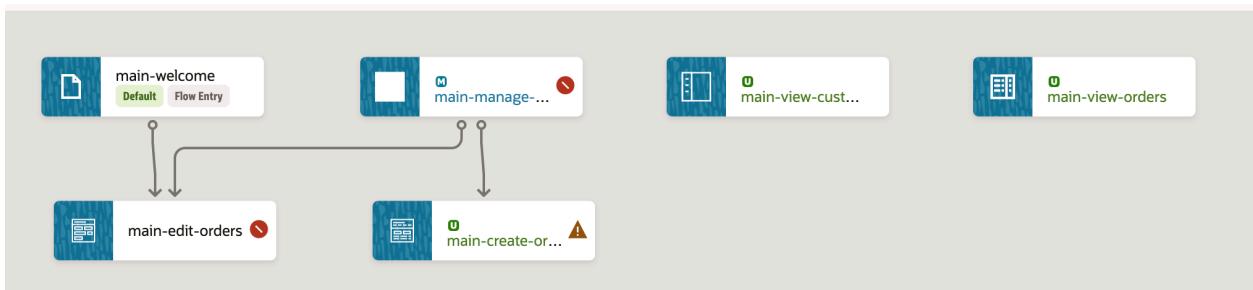
67. Drag the **General Overview Page Template** to the diagram. Name the page **main-view-orders** and click **Create**.





68. Drag an Item Overview Page Template to the diagram. Name it main-view-customers.

The screenshot shows the Oracle APEX 'main-manage-orders' application. On the left, the 'Components' sidebar is open, displaying various page templates under the 'Pages' category. An 'Item Overview Page Template' icon is highlighted with a red arrow pointing towards the central workspace. The workspace contains a process diagram with five pages: 'main>Welcome', 'main-manage-orders', 'main-view-orders', 'main-edit-orders', and 'main-create-orders'. A 'Create Page' dialog box is open in the foreground, prompting for a 'Page ID'. The input field contains the value 'main-view-customers', which is highlighted with a red border. Below the input field are 'Cancel' and 'Create' buttons.



69. Close the **main** tab.
70. On the **main-manage-orders** page, select **List Item Template**. Click its **Events** tab.

71. Add the **On 'spSecondaryAction'** event listener. The new action chain opens.

72. Drag a **Switch** action to the canvas. Select the Switch action and select **actionId** for the **Value**.

The screenshot shows the Oracle APEX workspace interface. At the top, there is a sidebar with a 'Logic' section containing actions: For Each, If, Return, Run in Parallel, Switch (which is highlighted with a red box), and Try Catch. Below this is a main panel titled 'switch (not defined)' which contains a 'case not defined' section with an 'Add Action' button and a 'default' section with an 'Add Action' button. To the right of the main panel is a toolbar with a red 'x' icon.

Below the main panel, there is a 'Variables' section. It includes a 'Filter' input field and two 'Create' buttons for 'Local' and 'Input Parameters'. Under 'Input Parameters', there is a list of variables: 'actionId' (highlighted with a red box), 'current', 'index', and 'key'. Further down, there is a 'Value *' input field containing the expression '[[actionPerformed]]'.

73. Select the case area and enter `order` for the **Value**.

The screenshot shows the configuration of a **Switch | case** component. On the left, there's a tree view showing the structure: `switch (actionId)`, `case 'not defined'`, and `default`. The `case` node has an `Add Action` button. On the right, the **Properties** panel is open, showing the **Value** field set to `order` (which is highlighted with a red box). Below the value field is a checked checkbox labeled `Break after`. The overall background has a wood-grain texture.

74. Drag a **Navigate To Page** action to the case **Add Action** area. Select **main-view-orders** for the **Page** property.

The screenshot shows the same **Switch | case** component structure as before. The `case 'order'` node now contains a green **Navigation To Page** icon followed by the text `not defined`. Below this node is a `break;` statement. The `default` node still has an `Add Action` button. The **Properties** panel is not visible in this specific screenshot, but the action was added to the case area.

```

switch (actionId)

case 'order'

    toMainViewOrders = Navigate To Page main-view-orders ✎

break;

default
    Add Action

```

Navigate To Page ...

⚠ Unexpected lexical declaration in case block.

ID:

Description:

Page * Create

main-view-orders ▼

Go to Page

Browser History: push

Store Result In:

Properties

75. Drag a **Navigate To Page** action to the **Create Case** area under the **break**.

```

switch (actionId)

case 'order'

    toMainViewOrders = Navigate To Page main-view-orders ✎

break; Create Case □ Navigate To Page

default
    Add Action

```

76. Select the case and set its **Value** to `customer`. Select the **Navigate To Page** action and set its **Page** property to **main-view-customers**.

The screenshot shows the Oracle ADF Faces Designer interface. At the top, there is a "Switch | case" component with a "Value" field containing the value "customer". A red box highlights the "customer" value. Below the component is a "Properties" panel. The main area contains Java-like code for a switch statement:

```

switch (actionId)
{
    case 'order'
        toMainViewOrders = Navigate To Page main-view-orders
        break;

    case 'customer'
        toMainViewCustomers = Navigate To Page main-view-customers
        Parameters
        bidirectionalNavigation: true
        inFlowBack: false
        overviewExpanded: false
        break;

    default
        Add Action
}

```

A red box highlights the "toMainViewCustomers" action and its parameters. The "Parameters" section includes three checked checkboxes: "bidirectionalNavigation: true", "inFlowBack: false", and "overviewExpanded: false".

Notice this type of page is coded to accept an **itemId**. You must pass the selected **customer Id** as the itemId to the **main-view-customers** page so it can retrieve the selected customer data.

77. Select the **toMainViewCustomers** action. Enter `{ current.data.customerId }` for the **itemId** property.

```

switch (actionId)

case 'order'

    toMainViewOrders = Navigate To Page main-view-orders ✎

break;

case 'customer'

    toMainViewCustomers = Navigate To Page main-view-customers ✎
        Parameters
        ☐ bidirectionalNavigation: true
        ☐ inFlowBack: false
        A itemId: current.data.customerId
        ☐ overviewExpanded: false

break;

default

    Add Action

```

78. You are missing the **customerId** field from the data source. You need to add the customerId field to your data source definition. Select the **Types** tab to add the field.

main-manage-orders x

Page Designer Action Chains (7) Event Listeners (5) Events Types (4) Variables (21) JavaScript JSON Settings

Action Chains > ListItemTemplateSpSecondaryActionChain > Diagram Tests

79. Click the **Edit from Endpoint** link.

```

▼ { } getallRraOrdersResponse Edit from Endpoint ,
  ▶ [ ] items

```

80. Select **customerId** and click **Finish**.

The screenshot shows a list of fields for a data source type definition. Under the heading 'customerFirst Name', there is a section for '# customerId'. The '# customerId' field is checked and highlighted with a red border.

- A creationDate
- A currencyCode
- # customerAddressId
- # customerAddressId1
- A customerFirstName
- # customerId
- # customerId1
- ▶ { } customerIdObject
- A customerLastName

The customerId field is added to the data source type definition. Now when the navigation is activated, the selected customerId will be passed to the **main-view-customers** page. You will develop that page in a later practice.

Note: For the purpose of this practice, we deliberately had you leave out adding the customerId when you defined the List View data source so that you could add it later and see how easy it is to add or remove fields for the data source type definition as needed. It's very common to have to add data source fields later as requirements change or evolve.

The view-orders page does not have an input parameter defined for it yet. You will configure that later when you develop that page.

81. Test if the new navigation works. Click the **Page Designer** tab and enter **Live** mode. Select one of the ... buttons for an order. Select **View Customer**. The page navigates to the **main-view-customer** page.

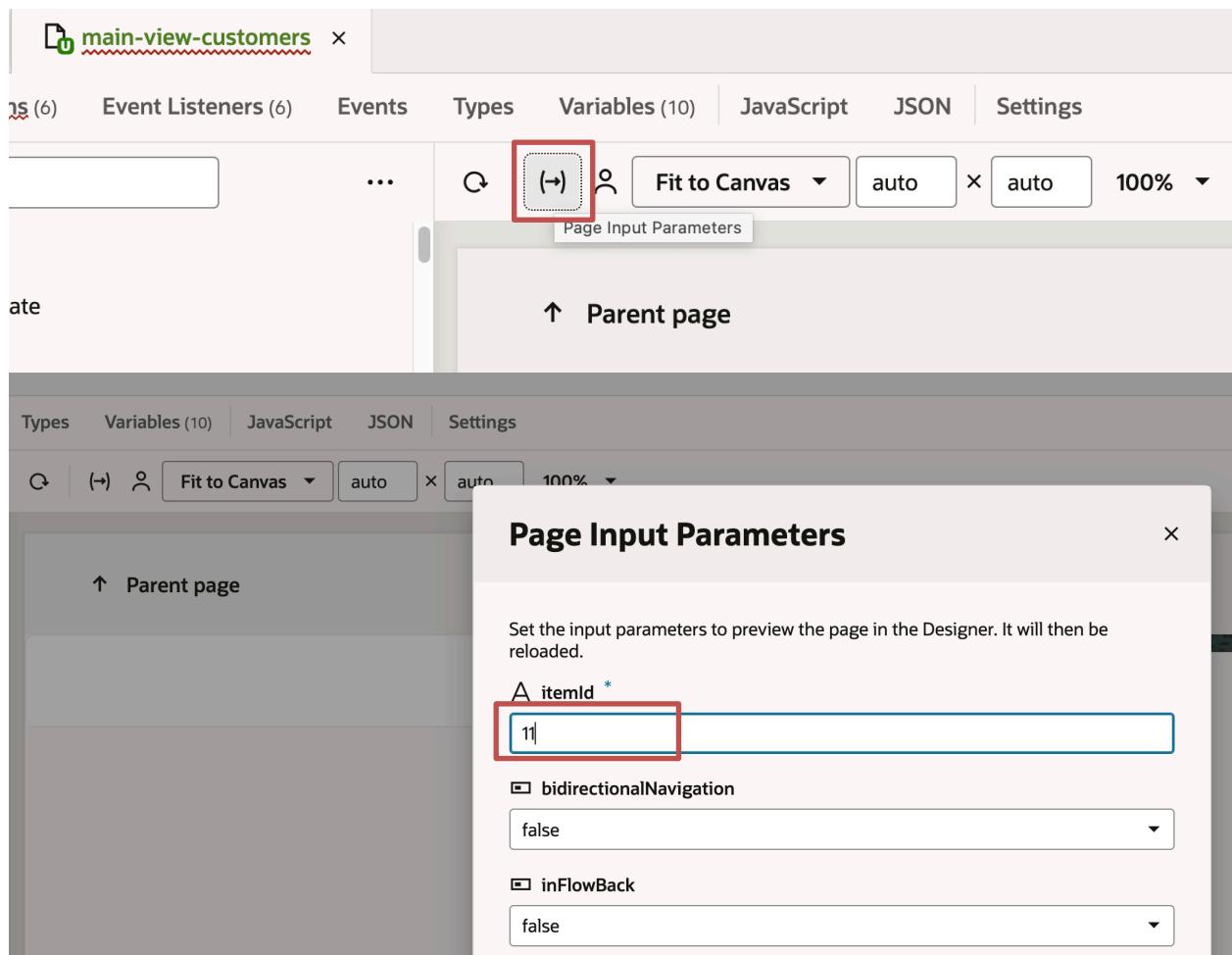
The screenshot shows the 'Manage Orders' page in live mode. It lists three orders. The third order (F000000315, 8/21/2022) has a context menu open, with the 'View Customer' option highlighted with a red box.

	Kathryn Rogers	F000000313 8/7/2022	\$5,151.00	
DRAFT	Kathryn Rogers	F000000314 8/11/2022	\$1,664.00	...
DRAFT	Kathryn Rogers	F000000315 8/21/2022	\$4,481.00	View Order Delete View Customer

82. Confirm the CustomerId was passed correctly. Click the **Page Input Parameters** button. This displays any parameters that were passed to the page.

83. Confirm that the **itemId** has a value. Click **Cancel** when done.

Note: Your itemId number may be different depending on which order you used.



84. Return to the **main-manage-orders** tab. Test the **View Order** navigation. The **main-view-orders** page should display. Return to Design mode.

Manage Orders ▾

Create Order

Page Subtitle

Try a customer last name or order number

DRAFT	Kathryn Rogers	F000000313 8/7/2022	\$5,151.00	...
DRAFT	Kathryn Rogers	F000000314 8/11/2022	\$1,664.00	...
DRAFT	Kathryn Rogers	F000000315 8/21/2022	\$4,481.00	...
PROCESSING	Dennis Lopez	F000000316 5/24/2022	\$4,682.00	...
PROCESSING	Dennis Lopez	F000000317 5/27/2022	\$3,357.00	View Order
PROCESSING	Dennis Lopez	F000000318 5/31/2022	\$7,884.00	Delete View Customer

85. Close all open tabs.

This completes this practice.

Practice 3-2: Create the View Orders Page

Overview

In this practice, you create the View Orders page, which displays read-only information about an order. The page presents information about the customer, the order, and the order's line items. Also, the product number for the line item is selectable and, when selected, displays the product details in a slide-out drawer.

Order F000000314

Your Page Subtitle here

Customer Kathryn Rogers Date 8/11/2022 Total Amount \$1,664.00

CustomerName	EmailAddress	Input Phone Number
Kathryn Rogers	krogers@internalmail.com	+1 650 506 0019

Shipping Address

919 Pudding Street
15op837
95727
US

ProductName	ProductNumber	ImageUrl	ListPrice	Quantity
Orion Fitness Wearable Gen3	RWD000103		99	9
AV Noise Cancelling Pro Headphones	RWD001003		299	1
Fastrack Wired Keyboard	RWD004002		79	6

Order F000000170

Your Page Subtitle here

Customer Judy Evans Date 6/5/2022 Total Amount \$8,126.00

CustomerName Judy Evans	EmailAddress jevans@internalmail.com	Input Phone Number +1 650 506 0011
Shipping Address		
364 Chocolate Blvd 6op998 91150 US		

ProductName	ProductNumber	ImageUrl	Link
AV Noise Cancelling Pro Headphones	RWD001003		\$129.99
Alpha Logic Wireless Optical Ergo Mouse	RWD005002		\$24.99
Oro Gold Wearable, Black	RWD000201		\$199.99
Fuzzy Slimline Tablet	RWD002002		\$149.99
Silicon Smart Active Smartwatch, Blue	RWD000402		\$149.99

Product detail

1 product



Fuzzy Slimline Tablet

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 Project Home, Workspaces, Git, Merge Requests, Maven, NPM, and Docker. The main area is titled "Organization" and shows a list of "Projects: 3". The first project listed is "OrderMgt_JDG Current", which is highlighted with a red box. Below it are "Order Management Applic..." and another partially visible project. A search bar and filter buttons for Member, Favorites, Owner, Shared, and All are at the top. A navigation bar at the bottom includes Projects, OCI Account, Build Executor Templates, Build Executors, Component Exchange, and a gear icon.

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 with the "Workspaces" tab selected. The workspace "OrderMgt_JDG" is highlighted with a red box. A context menu is open over this workspace, with the "Open in New Tab" option also highlighted with a red box. Other options in the menu include Rename, Export Visual Application, Switch Environment, Change Ownership, and Delete. A search bar and filter buttons for Mine and Others are at the top. A navigation bar at the bottom includes Workspace, Repository, and Current Branch.

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-view-orders** page. It opens in Page Designer.

5. Click the **Variables** tab to define the input parameter that will receive the passed order number from the **main-manage-orders** page.

6. Add a new String variable. Change its ID to `selectedOrderId`. Set it to be a **Required Input Parameter**.

7. Click the **+ Variable** button to create another variable named `contextualInfo` that will be used for contextual information displayed at the top of the screen. Make this of **Array** type.

Create Variable

Variable Constant

ID *

contextualInfo

Type *

Array

Cancel

Create & New

Create

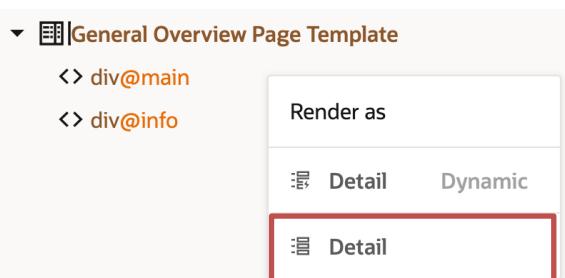
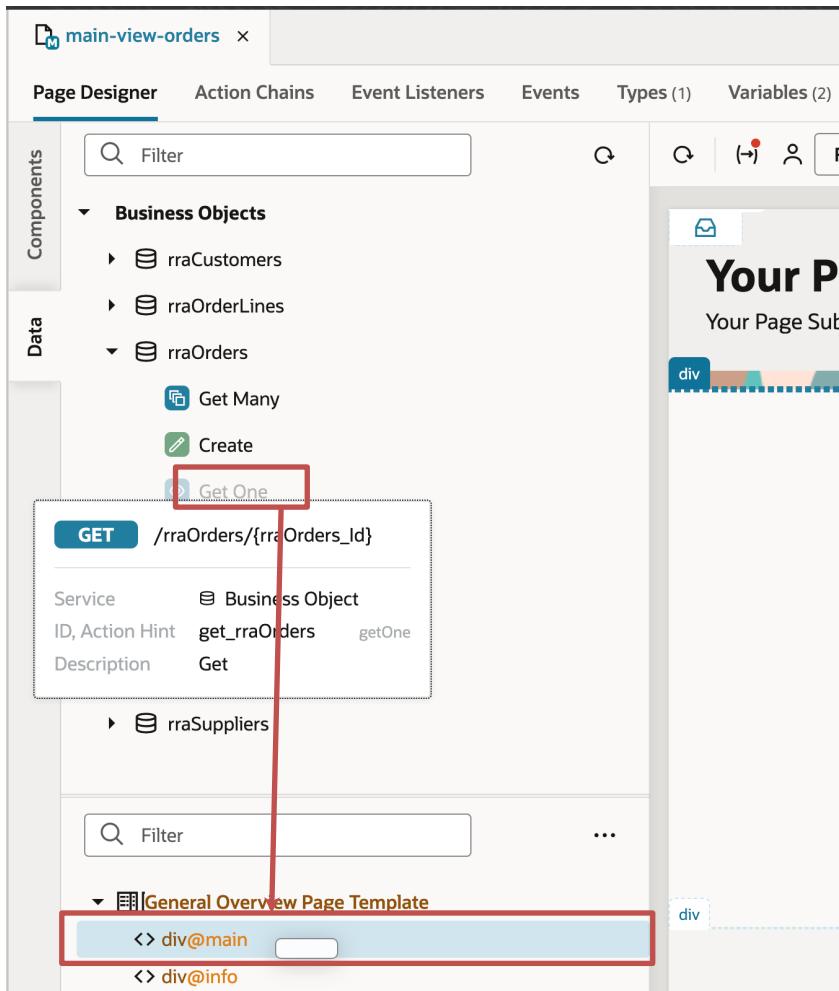
Variables

[] contextualInfo

{ } item[i]

A selectedOrderId *

8. Click the **Page Designer** tab to define the page.
9. Click the **Data** tab and expand **Business Objects > rraOrders** and drag **Get One** to the **div@main** element in the Structure pane. Select **Detail** (not Dynamic) when prompted.
Thought question: *Why Get One?* Because the page needs to retrieve a single record – not a collection - using the **selectedOrderId** that is passed to the page from the selected order on the **view-manage-orders** page.



10. From the Endpoint Structure, select the following fields and click **Next**.

- customerFirstname
- customerLastName
- orderDate
- orderId
- orderNumber
- orderStatus
- customerIdObject> items > item[i] > customerName, emailAddress, phone
- orderLinesCollection > items > item[i] > imageUrl, orderLineId, productName

- shippingAddressCity
- shippingAddressCountry
- shippingAddressLine11
- shippingAddressLine21
- shippingAddressPostalCode
- shippingAddressState
- shipToAddressId
- totalAmount

Endpoint Structure

Filter

▼ □ { } response

- A attachmentEntityName
- A basketFlag
- A billingAddressCity
- A billingAddressCountry
- A billingAddressLine11
- A billingAddressLine21
- # billingAddressPostalCode
- A billingAddressState
- # billToAddressId
- A comments
- # contextKey
- A createdBy
- { } creationDate
- A currencyCode
- # customerAddressId
- # customerAddressId1
- A customerFirstName
- ↗ customerId
- # customerId

□ # customerId1

▼ □ { } customerIdObject

- # count
- hasMore

▼ □ [] items

▼ □ { } item[i]

 - A billingAddressCity
 - A billingAddressCountry
 - A billingAddressLine1
 - A billingAddressLine2
 - # billingAddressPostalCode
 - A billingAddressState
 - # contextKey
 - A createdBy
 - { } creationDate
 - # customerAddressId
 - # customerAddressId1
 - # customerId
 - A customerName
 - A customerNumber
 - A emailAddress
 - A firstName
 - # id
 - A lastName

```

□ { } lastUpdateDate
□ A lastUpdatedBy
▶ □ [] links
□ # objectVersionNumber
□ # objectVersionNumber1
□ # objectVersionNumber2
 # phone
□ # primaryBillingAddressId
□ # primaryShippingAddressId
□ A shippingAddressCity
□ A shippingAddressCountry
□ A shippingAddressLine1
□ A shippingAddressLine2
□ A shippingAddressState
□ # shippingPostalCode

□ # limit
▶ □ [] links
□ # offset
□ # totalResults
 A customerLastName
□ A displayStatus
□ # id
□ { } lastUpdateDate
□ A lastUpdatedBy

```

```

□ A lastUpdatedBy
▶ □ [] links
□ # objectVersionNumber
□ # objectVersionNumber1
□ # objectVersionNumber2
□ # objectVersionNumber3
 A orderDate
 # orderId
▼ □ { } orderLinesCollection
□ # count
□ ↗ hasMore
▼ □ [] items
▼ □ { } item[i]
□ # amount
□ # contextKey
□ A createdBy
□ { } creationDate
□ A currencyCode
□ 📅 deliveredDate
□ # id
 A imageUrl
□ { } lastUpdateDate
□ A lastUpdatedBy
▶ □ [] links

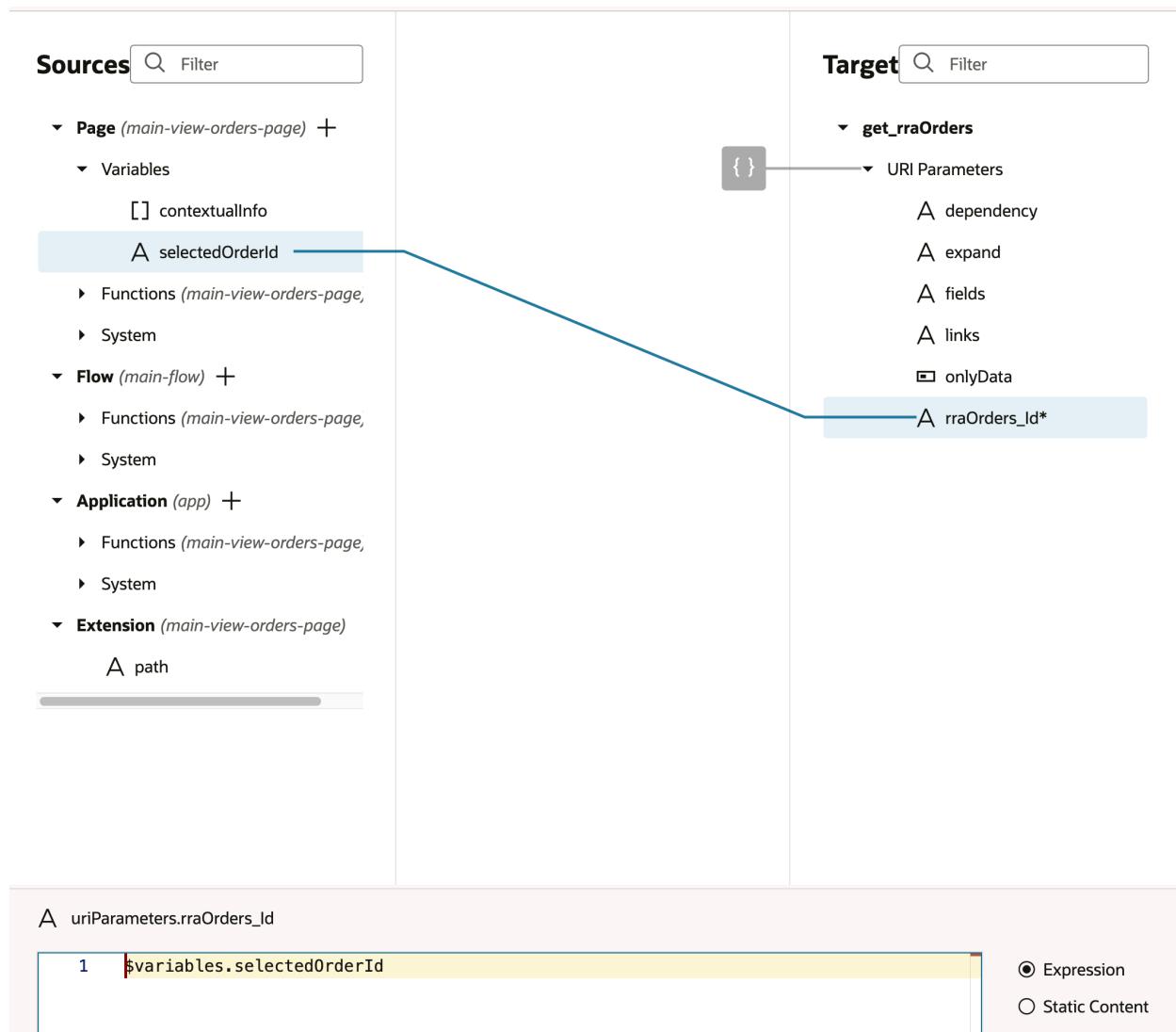
```

```
-- ## objectVersionNumber1
 # objectVersionNumber1
 ↳ orderId
 # orderLineId
 # productId
 # productId1
 A productName
 A productNumber
 # quantity
 # limit
▶  [] links
 # offset
 # totalResults
 A orderNumber
 A orderStatus
 # primaryBillingAddressId
 # primaryShippingAddressId
 A shippingAddressCity
 A shippingAddressCountry
 A shippingAddressLine1
 A shippingAddressLine21
 # shippingAddressPostalCode
 A shippingAddressState
 # shipToAddressId
 # totalAmount
```

Fields

A customerFirstName X	Input Text ▾
A customerLastName X	Input Text ▾
A orderDate X	Input Text ▾
# orderId X	Input Number ▾
A orderNumber X	Input Text ▾
A orderStatus X	Input Text ▾
A customerName X	Input Text ▾
A emailAddress X	Input Text ▾
# phone X	Input Number ▾
A imageUrl X	Input Text ▾
# orderLineId X	Input Number ▾
A productName X	Input Text ▾
A shippingAddressCity X	Input Text ▾
A shippingAddressCountry X	Input Text ▾
A shippingAddressLine1 X	Input Text ▾
A shippingAddressLine2 X	Input Text ▾
# shippingAddressPostalCode X	Input Number ▾
A shippingAddressState X	Input Text ▾
# shipToAddressId X	Input Number ▾
# totalAmount X	Input Number ▾

11. Drag Sources > Page > selectedOrderId onto Target > get_rraOrders > rraOrders_Id and click **Finish**.



You are returned to the page and the fields have been added to the Form Layout.

```
<div>
    <div>
        <div>
            <div>
                <div>
                    <div>
                        <div>
                            <div>
                                <div>
                                    <div>
                                        <div>
                                            <div>
                                                <div>
                                                    <div>
                                                        <div>
                                                            <div>
                                                                <div>
                                                                    <div>
                                                                        <div>
                                                                            <div>
                                                                                <div>
                                                                                    <div>
                                                                                        <div>
                                                                                            <div>
                                                                                                <div>
                                                                                                    <div>
                                                                                                        <div>
                                                                                                            <div>
                                                                                                                <div>
                                                                                                                    <div>
                                                                                                                        <div>
                                                                                                                            <div>
                                                                                                                                <div>
                                                                                                                                    <div>
                                                                                                                                        <div>
                                                                                                                                            <div>
                                                                                                                                                <div>
                                                                                                                                                    <div>
................................................................
```

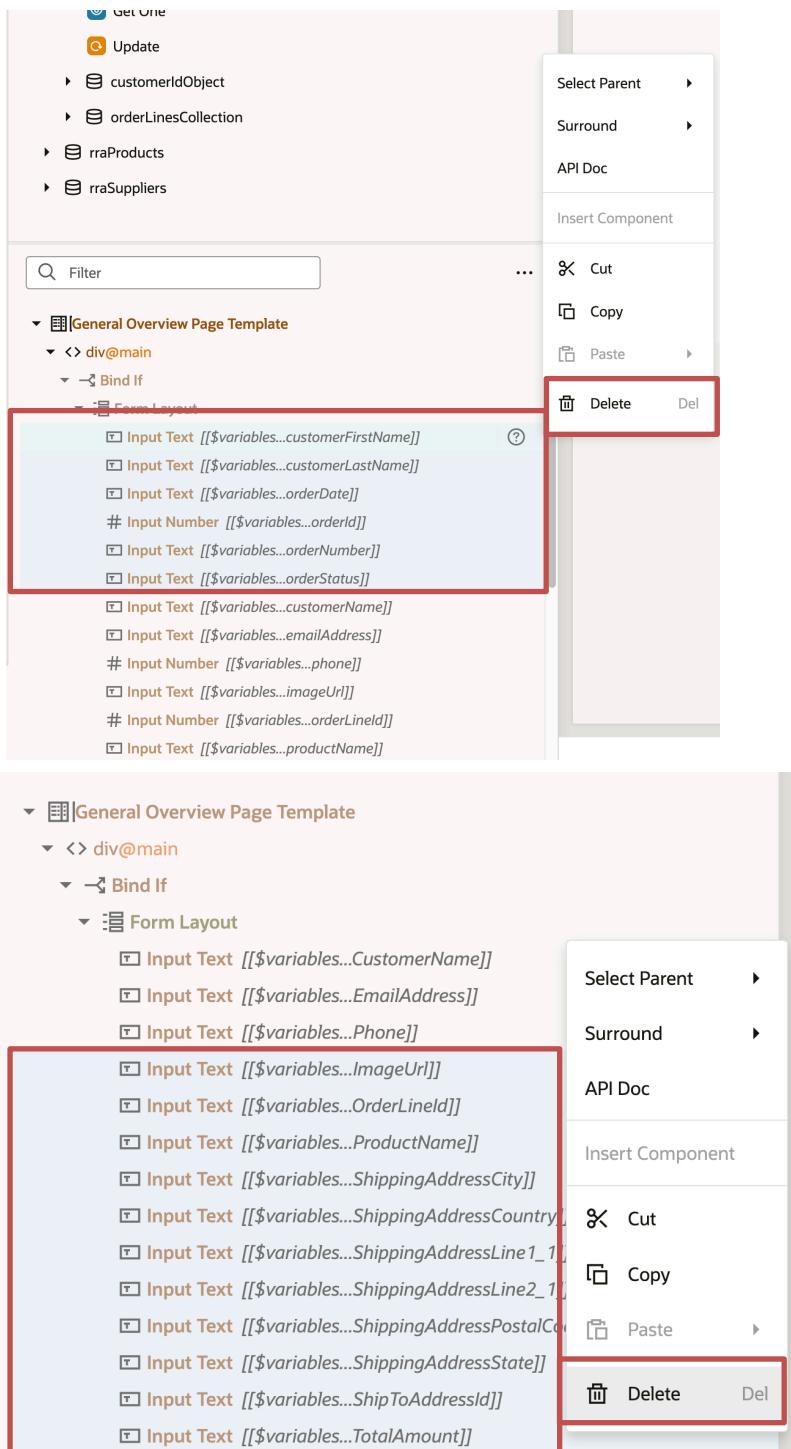
The screenshot shows the Oracle ADF Faces Designer interface with a tree view of components. The 'Form Layout' node is expanded, revealing several 'Input Text' and 'Input Number' components. Some of these components have red boxes drawn around them, specifically highlighting the following fields:

- # Input Number [[\${variables...orderId}]]
- # Input Number [[\${variables...phone}]]
- # Input Number [[\${variables...orderLineId}]]
- # Input Number [[\${variables...shippingAddressPostalCode}]]
- # Input Number [[\${variables...shipToAddressId}]]
- # Input Number [[\${variables...totalAmount}]]

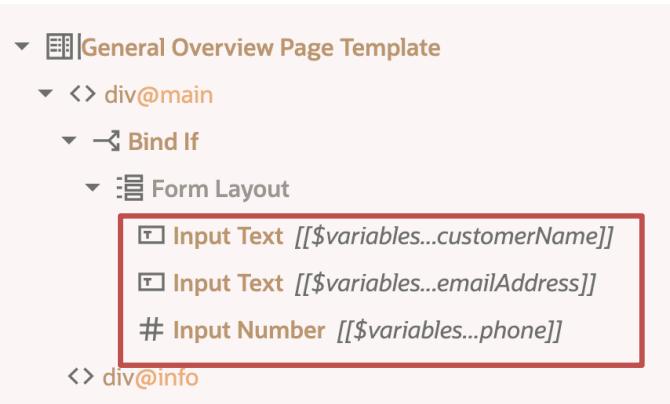
Note: Some fields were automatically created using Input Number components by VBS.

12. Delete all fields from the Form Layout **except customerName, emailAddress, and phone**.

Tip: You can multi-select more than one field and delete them as a group, as follows.



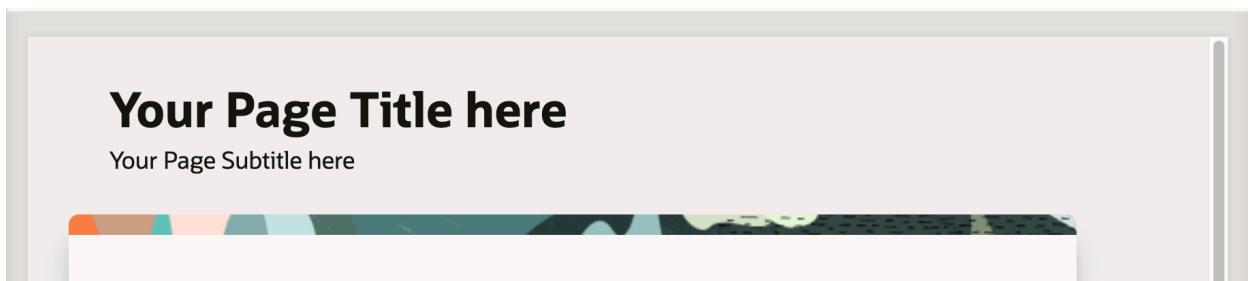
Why delete all these fields after you just added them? You won't use the data fields you are deleting here in this Form Layout, but you'll use them elsewhere, and they will be available to reference later because you added them while going through the Detail wizard. Using the Detail form wizard, VBS automatically creates the action chain and event listener (vbEnter) that will call the Business Object endpoint using selectedOrderId and retrieve the record when the page first loads. This saves you much time and effort in creating the necessary action chain and REST actions and assignments to variables.



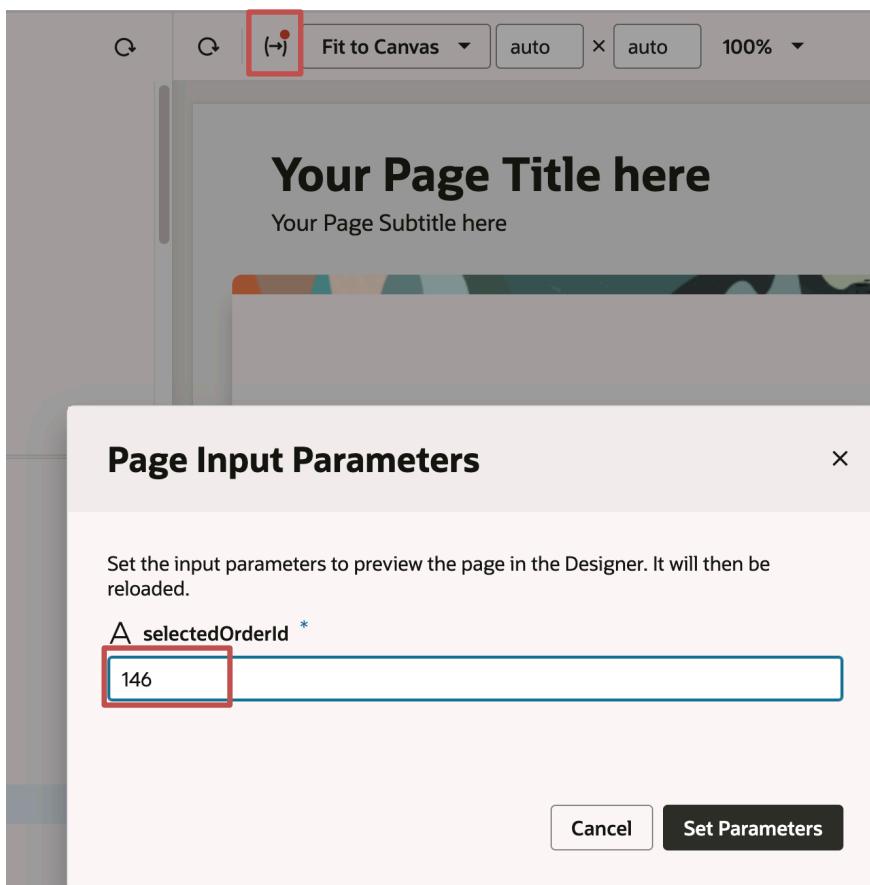
Optional: If you are curious, you can examine the event listener and click the Go to Action Chain link to examine how the record is retrieved. This can be useful as a guide and template for when you want to create your own REST load action chains. Return to Page Designer when you are done.

Page Designer	Action Chains (1)	Event Listeners (1)
		Lifecycle Event Listeners <ul style="list-style-type: none"> vbEnter <ul style="list-style-type: none"> → loadRraOrdersChain Custom Event Listeners <p>No Custom Event Listeners defined.</p> Component Event Listeners <p>No Component Event Listeners defined.</p>

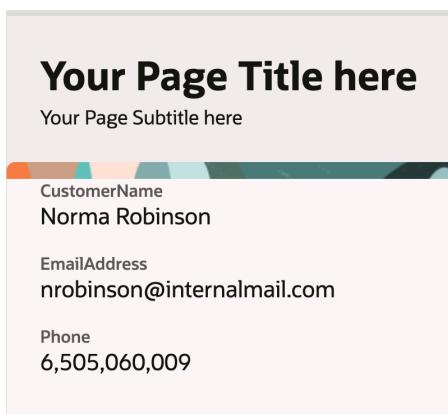
13. No data appears yet, as no record has been retrieved. You'll resolve that next.



14. Notice the red dot next to the **Page Input Parameters** button. This indicates that required page input parameters are missing. Click the **Page Input Parameters** button and enter an order number: 146 for the **selectedOrderId** parameter and click **Set Parameters**.



Note: When using the app, you start on the main-manage-orders page and then select the View Order action, which would pass the order number and navigate to this page, which would execute the REST call to load the record. You'll test this later.



15. Select **General Overview Page Template** in the **Structure** pane.

The screenshot shows the Oracle ADF Structure pane. A red box highlights the "General Overview Page Template" node under the "div@main" node. The "div@main" node also has a red box around it. Below "div@main", there are nodes for "Bind If", "Form Layout", and three "Input Text" components with variable bindings: "\$variables...customerName", "\$variables...emailAddress", and "\$variables...phone". At the bottom, there is a "div@info" node.

16. For its **Page Title** property, enter this code:

```
[[ "Order " + $variables.raaOrders.orderNumber ]]
```

The screenshot shows the Oracle ADF Properties pane for the "General Overview Page Template". The "Page Title" property is highlighted with a red box and contains the dynamic expression "[["Order " + \$variables.raaOrders.orderNumber]]". Other properties shown include "ID" (oj_gop1) and "Page Subtitle".

You should see the title change to reflect the order number.

The screenshot shows a preview of the "Order F000000146" page template. The main title is "Order F000000146" in a large bold font, with a red box around it. Below it is a subtitle placeholder "Your Page Subtitle here". Underneath, there is a decorative bar with orange, teal, and blue segments. The page displays customer information: "CustomerName Norma Robinson", "EmailAddress nrobinson@internalmail.com", and "Phone 6,505,060,009".

17. Ensure **Contextual Info Label** is selected under **Display Options**.

The screenshot shows the 'General Overview Page Template' configuration. On the left, there are tabs for 'General', 'Events', and 'All'. The 'General' tab is selected. The 'ID' field contains 'oj_gop1'. The 'Page Title' field contains the expression '[["Order "+ \$variables.rraOrders.orderNumber]]' which is highlighted with a red border. The 'Page Subtitle' field is empty. Below these, there is a section for 'The display mode for the header.' with 'Light' and 'Mixed' options, where 'Light' is selected. On the right, a vertical 'Properties' panel is visible.

Click the **All** tab to enter the contextual info code. For the **Contextual Info** property, enter this code. You can copy the code from the code text file.

```
[[ [{"label":"Customer","value":$variables.rraOrders.customerFirstName  
+ ' ' +$variables.rraOrders.customerLastName},  
 {"label":"Date","value":  
 $flow.functions.formatDate($page.variables.rraOrders.orderDate)}, {"label":"Total  
Amount","value":$flow.functions.formatCurrency($flow.constants.default  
CurrencyCode, $page.variables.rraOrders.totalAmount ?  
 $page.variables.rraOrders.totalAmount : 0)}] ]]
```

18. The textual information is displayed in the header.

General Overview Page Template

General Events **All**

Filter

ID
oj_gop1

General Overview Page Template Attributes

Avatar Image

Badge

Contextual Info

```
[[{"label": "Customer", "value": "$variables.raOrders.customerFirstName + ' ' + variables.raOrders.customerLastName"}, {"label": "Date", "value": "$flow.functions.formatDate($page.variables.raOrders.orderDate)"}, {"label": "Total Amount", "value": "$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode, $page.variables.raOrders.totalAmount ? $page.variables.raOrders.totalAmount : 0)"}]]
```

General Overview Page Template

General Events All

Filter

ID
oj_gop1

General Overview Page Template Attributes

Avatar Image

Badge

Contextual Info

```
[[{"label": "Customer", "value": "$variables.raOrders.customerFirstName + ' ' + variables.raOrders.customerLastName"}, {"label": "Date", "value": "$flow.functions.formatDate($page.variables.raOrders.orderDate)"}, {"label": "Total Amount", "value": "$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode, $page.variables.raOrders.totalAmount ? $page.variables.raOrders.totalAmount : 0)"}]]
```

19. Select the **Form Layout** in the Structure pane and change its **Columns** property to 4.

The screenshot shows the Oracle ADF interface with two main panes: the Structure pane at the top and the Properties pane below it.

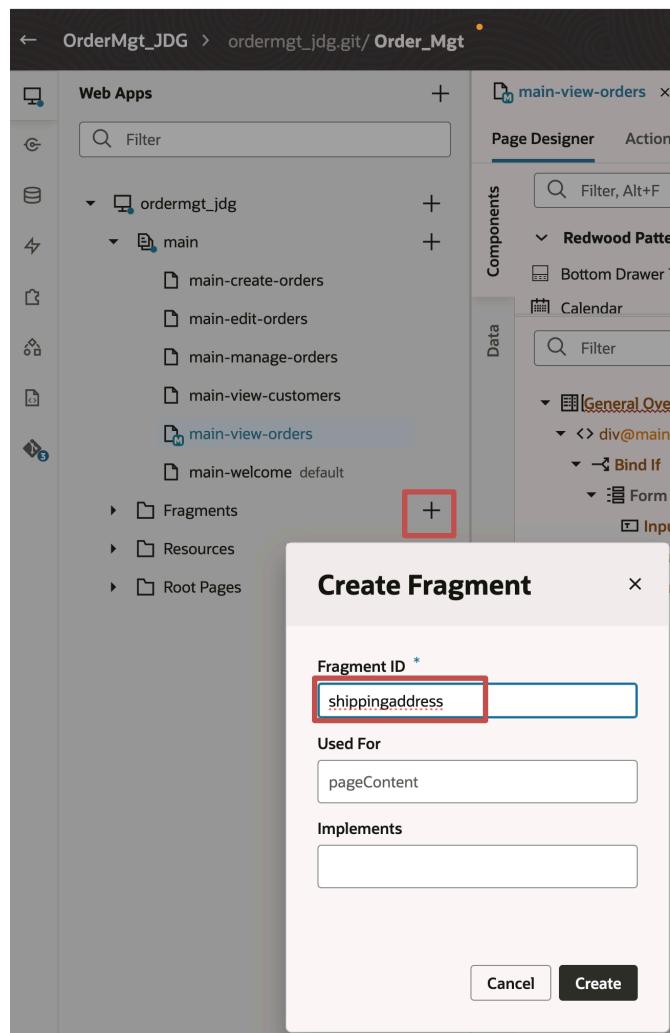
Structure Pane:

- General Overview Page Template
- div@main
- Bind If
- Form Layout (highlighted with a red box)
- Input Text [[variables...customerName]]
- Input Text [[variables...emailAddress]]
- # Input Number [[variables...phone]]
- div@info

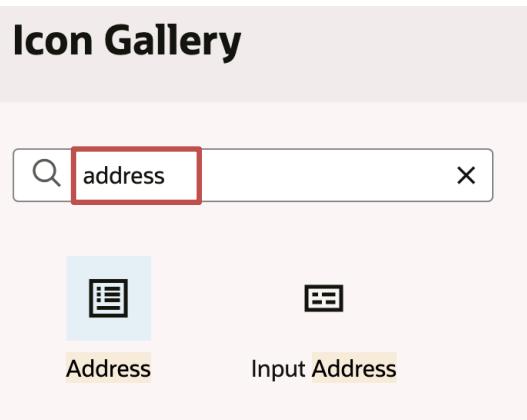
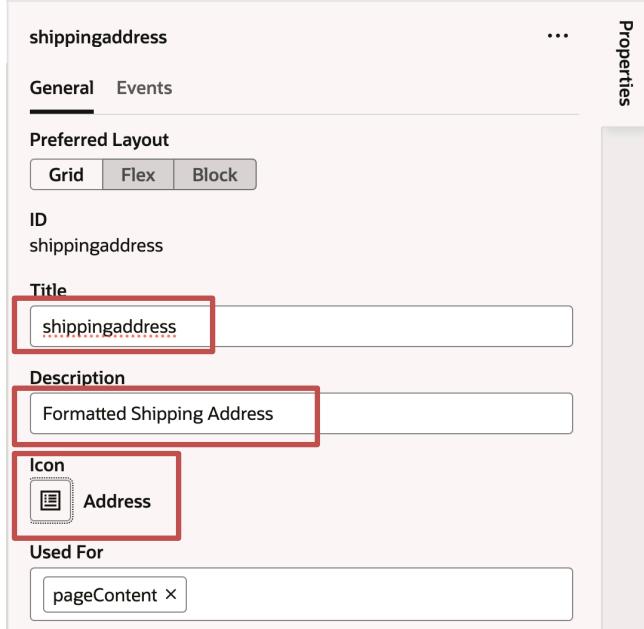
Properties Pane:

- Form Layout
- General Events All Quick Start
- Filter
- Class (empty)
- ID (empty)
- Slot (empty)
- Form Layout Attributes
 - Colspan Wrap (empty)
 - Columns (value: 4, highlighted with a red box)

20. Create a **fragment** for the shipping address. Click the **+** to create a new fragment named **shippingaddress** in the Web Apps pane. Click **Create**. The Fragment Designer opens.



21. In the Property pane for the fragment, enter a **Title** of `shippingaddress`, a **Description** of `Formatted Shipping Address` and set the icon to the **Address** icon: click the **Icon** button and enter `address` in the search field, select **Address**, and click **Select**.



22. Click the **Variables** tab to create the variables shown in the following screenshot to hold values passed to the fragment from the containing page, all of type String. Set them all to be **Enabled Input Parameters**.

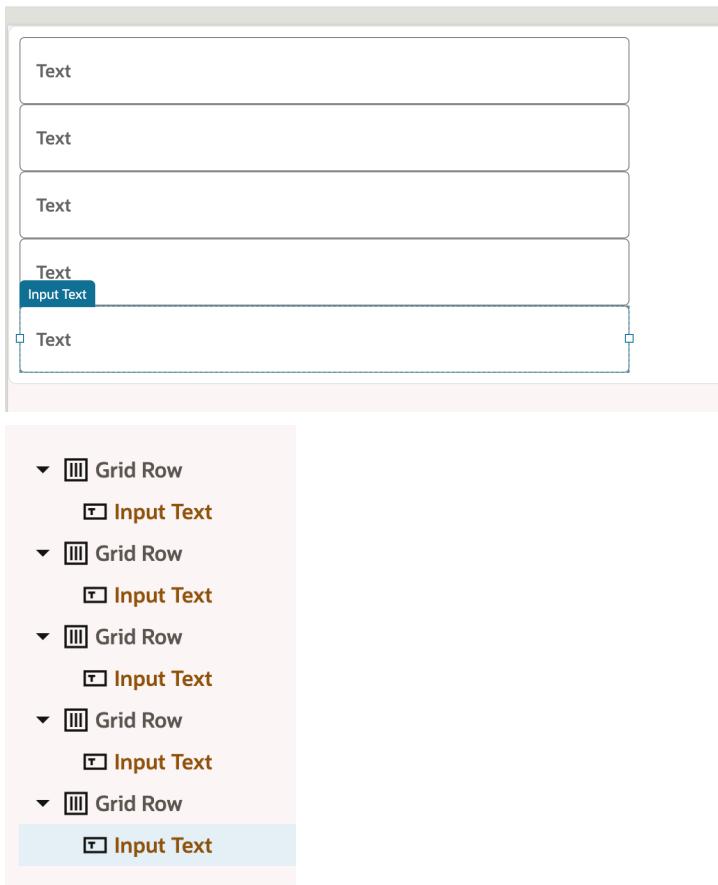
Variables: city, country, postalCode, state, street_1, street_2

The screenshot shows the Fragment Designer interface with the Variables tab selected. The Variables section lists six variables: city, country, postalCode, state, street_1, and street_2. The street_2 variable is selected, and its properties are displayed on the right side of the screen. The properties include:

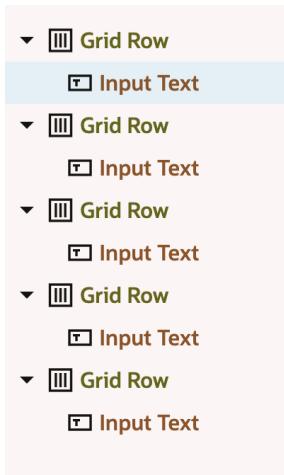
- ID: street_2
- Type: String
- Label: (empty)
- Description: (empty)
- Input Parameter: Enabled (highlighted with a red box)

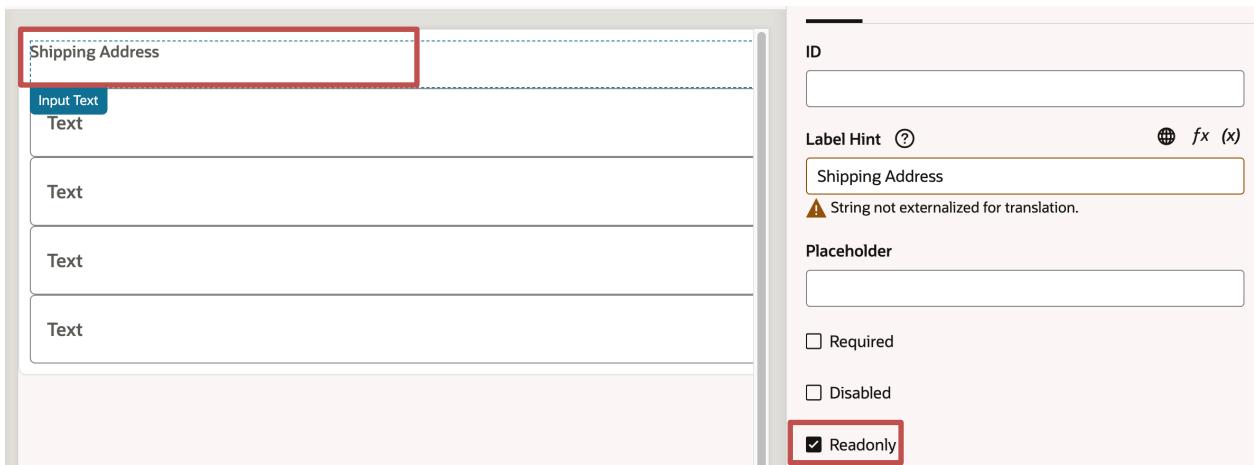
23. Click the **Fragment Designer** tab and add **five (5) Input Text** components to the canvas, stacked one below/after the previous.

The screenshot shows the Fragment Designer interface with the Fragment Designer tab selected. The Components panel on the left shows the 'Forms & Inputs' category with 'Input Text' selected. A component is being dragged from the Components panel onto the canvas. The canvas has five 'Input Text' components stacked vertically, each highlighted with a red box. The components are arranged in a vertical stack, with the first component at the top and the fifth component at the bottom.



24. Select the **first Input Text** component, set it to **Readonly** and set its **Label Hint** property to Shipping Address.





Select the next **Input Text** component, set it to be **Readonly**, delete the **Label Hint** text and, on the **Data** tab, bind the **Value** to `{ ${variables.street_1} }`.

Input Text

General Data Events All

Value [\(?\)](#) [fx \(x\)](#) Select Variable

Variables

Filter

- Fragment
- Variables
- city
- country
- postalCode
- state
- street_1**
- street_2

main-view-orders shippingaddress

Fragment Designer Action Chains Event Listeners Events Types Variables (6) Design Time JavaScript JSON Settings

Components

Forms & Inputs

Input Text

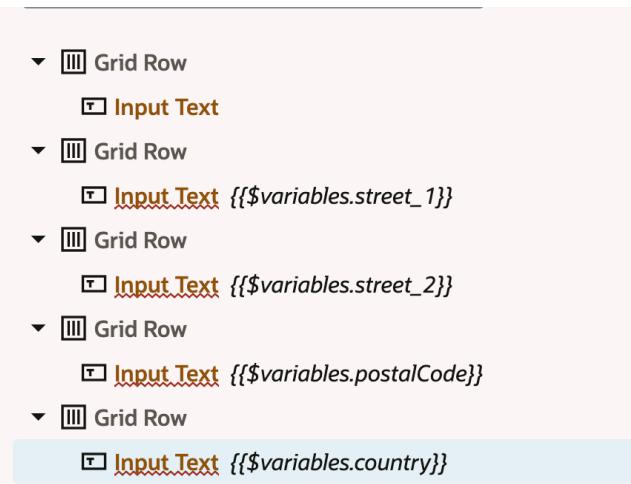
Shipping Address

Value `{ ${variables.street_1} }`

Components Tree

Input Text `{ ${variables.street_1} }`

25. Set the next Input Text component to **Readonly**, delete the **Label Hint** text, and bind the **Value** to `{$variables.street_2}` .
26. Set the next Input Text component to **Readonly**, delete the **Label Hint** text, and bind the **Value** to `{$variables.postalCode}` .
27. Set the next Input Text component to **Readonly**, delete the **Label Hint** text, and set the **Value** to `{$variables.country}` .



28. Close the **shippingaddress** tab to return to the **main-view-orders** tab.
29. Enter `fragment` in the Components Filter. Then, drag the **shippingaddress** fragment onto the **div@main** element and it is appended to the div.

The screenshot shows the Oracle ADF Page Designer interface. The top navigation bar includes tabs for "Page Designer", "Action Chains (1)", "Event Listeners (1)", "Events", and "Types". The left sidebar has sections for "Components" and "Data". Under "Components", there are "Layouts" (with "Fragment Container") and "Fragments" (with "page-header" and "shippingaddress", which is highlighted with a red box). A search bar at the top of the main area shows "Fragment". Below the search bar, a message says "You can also try searching for 'Fragment' at Component Exchange...". A "Search Exchange" button is present. The right side shows a preview of a page titled "Order" with a "Your Pag" section and a "Customer" section containing a form with fields like "Name" (Norma) and "Phone" (Robinsc). A detailed view of the "General Overview Page Template" is shown in the bottom right, where the "shippingaddress" fragment is defined within a "Bind If" block under a "Form Layout".

Hint: If you do not see `shippingaddress` in the Components list, close the page and reopen it.

```

▼ ┌─[General Overview Page Template]
  ─<> div@main
    ─→ Bind If
      ─ Form Layout
        └─ Input Text [[$variables...customerName]]
        └─ Input Text [[$variables...emailAddress]]
        └# Input Number [[$variables...phone]]
      <> div@info
      └─ shippingaddress
  <> div@info

```



30. With the **shippingaddress** fragment selected, bind its **city** parameter to:
`[[$variables.rraOrders.shippingAddressCity]]`

Input Parameters

A city ?	fx (x)	Select Variable
---	--	-----------------

Variables

A city	Edit	
A country	Edit	
A state	Edit	
A street_	Edit	
A shippingAddressCity	Edit	

A city

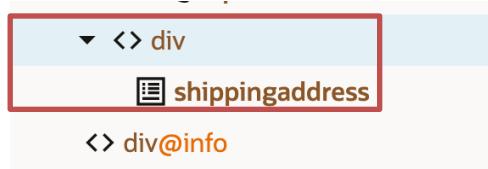
`[[$variables.rraOrders.shippingAddressCity]]`

31. Repeat this for the following parameters:

Note: City does not display as it does not have a field for it in the fragment. We left that off to avoid too much repetition. But, it does illustrate you can pass data to fragments that might not be displayed and can be referenced in other ways within the fragment.

Optional: If you so choose, you can go back and add the field to the fragment.

32. Fragments don't get styles, so surround the fragment in a div so you can apply a class to pad the fragment in from the left. Right-click the **shippingaddress** fragment and select **Surround > Div**.



33. Select the **div** and enter `oj-sm-margin-10x-start` for the **Class**. The fragment margin is changed and is indented from the left.

The screenshot shows the properties panel for a 'div' component. The 'General' tab is selected. In the 'Class' field, the value 'oj-sm-margin-10x-start' is entered and highlighted with a red box. There is also a small 'x' icon next to the class name. The 'Properties' sidebar is visible on the right.

The screenshot shows a form layout with several fields. On the left, there is a 'CustomerName' field with the value 'Norma Robinson'. Next to it are 'EmailAddress' and 'Phone' fields with values 'nrobinson@internalmail.com' and '6,505,060,009' respectively. Below these is a 'div' component with a dashed border, which contains a 'Shipping Address' section. This section includes an address line '909 Chocolate Ave', a zip code '18op959', a city '90747', and a country 'US'. A red box highlights this 'div' component.

34. Notice the phone number needs to be formatted. Replace the Input Number component with an Input Phone Number component. Drag an **Input Phone Number** component onto the **Form Layout** and it will be appended after the Input Number component.

main-view-orders

Page Designer Action Chains (1) Event Listeners (1) Events Types (2) Variables (3) Java

Components Data

Forms & Inputs

Input Phone Number

Phone

You can also try searching for "phone" at Component Exchange...

Search Exchange

Filter

General Overview Page Template

Bind If

Form Layout

Input Text Name
Norma Robinson

Shipping Address
909 Chocolate Ave
180p959
90747
US

Input Text [[variables...emailAddress]]
Input Number [[variables...phone]]

shippingaddress

div@info

General Overview Page Template

Bind If

Form Layout

Input Text [[variables...customerName]]
Input Text [[variables...emailAddress]]
Input Number [[variables...phone]]

Input Phone Number

shippingaddress

div@info

35. Select the **Input Number** component and click its **Data** tab. Copy the Value that is there.

The screenshot shows the Oracle ADF interface. At the top, a tree view displays the 'General Overview Page Template' structure. A red box highlights the '# Input Number [[variables...phone]]' component under the 'Form Layout' section. Below this, a detailed view of the '# Input Number' component is shown. The 'Data' tab is selected. In the 'Value' field, the expression '[[\$variables.raaOrders.customerIdObject.items[0]? .phone]]' is displayed, which is also highlighted with a red box.

`[[$variables.raaOrders.customerIdObject.items[0]? .phone]]`

36. Select the new **Input Phone Number** component, click its **Data** tab and paste the copied value into the component's **Subscriber Number** property.

The screenshot shows the Oracle ADF interface. A red box highlights the 'Input Phone Number' component. Below it, the 'Input Phone Number' component's Data tab is selected. In the 'Value' field, the text 'International Dial Code' is present. In the 'Subscriber Number' field, the expression '[[\$variables.raaOrders.customerIdObject.items[0]? .phone]]' is displayed and highlighted with a red box.

Customer Norma Robinson Date 8/13/2022 Total Amount \$6,895.00

CustomerName Norma Robinson	EmailAddress nrobinson@internalmail.com	Phone 6505,060,009	Input Phone Number +1 650 506 0009
--------------------------------	--	-----------------------	---------------------------------------

37. The phone number is formatted correctly. You can now delete the Input Number component.

The screenshot shows the Oracle ADF Structure pane. A context menu is open over the '# Input Number' component, which is highlighted with a red box. The menu options include Cut, Copy, Paste, and Delete, with 'Delete' also highlighted with a red box.

CustomerName Norma Robinson	EmailAddress nrobinson@internalmail.com	Input Phone Number +1 650 506 0009
Shipping Address		
909 Chocolate Ave 18op959 90747 US		

Add a table of Order Lines. Filter all the Order Lines for the specific order by the orderNumber.

38. Click the **Data** tab. Expand **Business Objects > rraOrders > orderLinesCollection** and drag **Get Many** below the **div@main** but **before** the **div@info**. Pay attention to the horizontal placement indicator line. Select **Table** to **Render as** when prompted.

Hint: If you make a mistake, use Undo to correct your work. If the table is not in the correct place, you can drag and drop it within the Structure pane.



main-view-orders

Page Designer Action Chains (1) Event Listeners (1) Events

Components Data

Filter

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

GET /rraOrders/{rraOrders_Id}/child/orderLinesCollection

Service	Business Object
Action Hint	getall_rraOrders-orderLinesCollection getMany
Description	Get all

rraSuppliers

Filter

- General Overview Page Template
 - <> div@main
 - Bind If
 - Form Layout
 - Input Text [[variables...customerName]]
 - Input Text [[variables...emailAddress]]
 - Input Phone Number
 - <> div
 - shippingaddress
 - div.info