

1. Select **productName**, **productNumber**, **imageUrl**, **listPrice**, **quantity**, and **amount** in that order. Change the type for **imageUrl** to **Image** and change the type for **productNumber** to **Hyperlink** and change **listPrice** and **amount** to **Currency** and click **Next**.

Columns

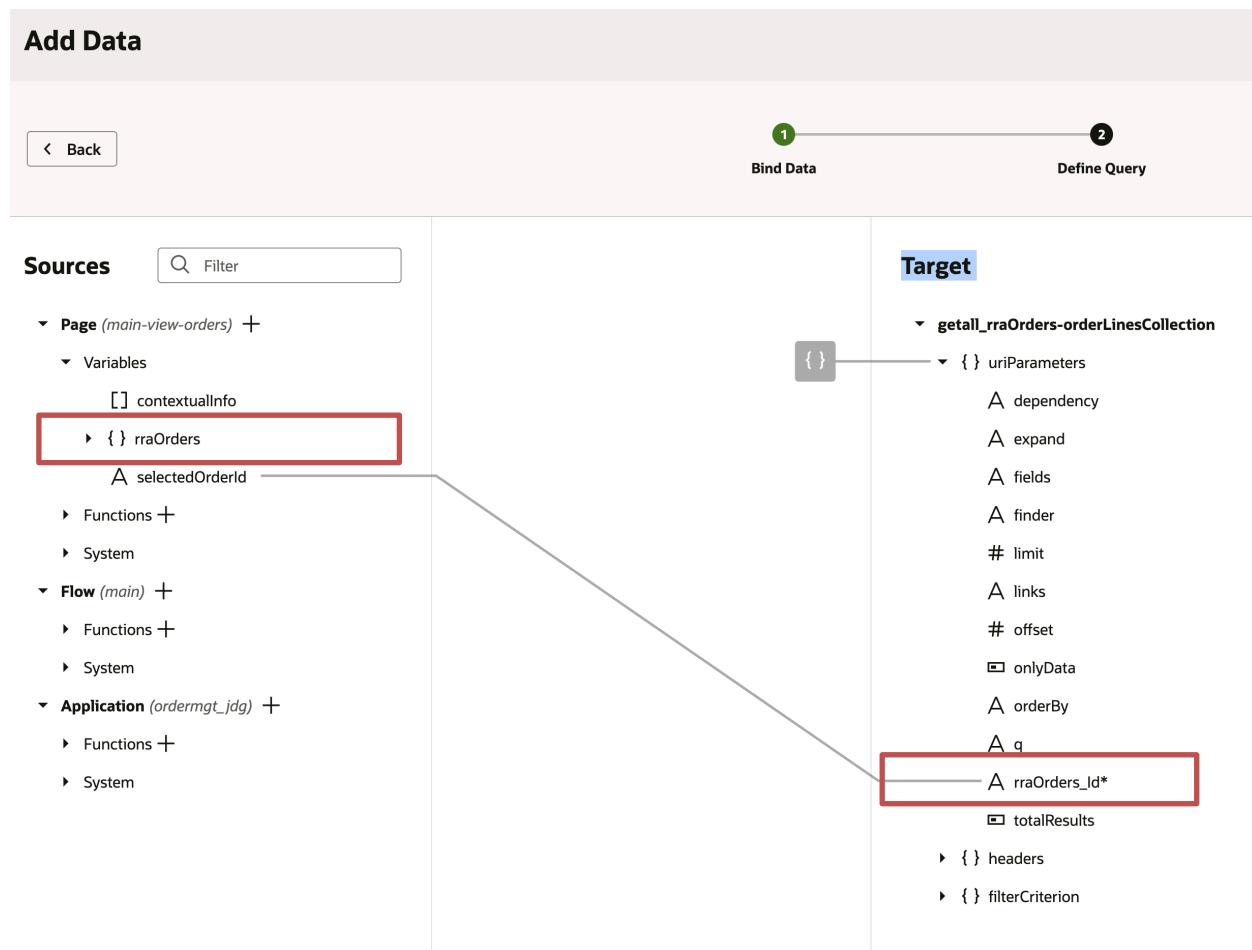
A productName X	Text ▾
A productNumber X	Text ▾
A imageUrl X	Text ▾
# listPrice X	Text ▾
# quantity X	Text ▾
# amount X	Text ▾
Drop a field from Endpoint	

- Text
- Hyperlink
- Heading
- Paragraph
- Image
- Input Text
- Text Area

Columns

A productName X	Text ▾
A productNumber X	Hyperlink ▾
A imageUrl X	Image ▾
# listPrice X	Currency ▾
# quantity X	Text ▾
# amount X	Currency ▾

2. Drag the **Sources > Page selectedOrderId** variable to the **Target > getall_rraOrders-orderLinesCollection > rraOrders_Id** and click **Finish**.



3. You don't see the table display. Change to **Code** view type in `slot="main"` into the table element where indicated in the screenshot. Click the **Code** button to edit the code. After pasting the code, click back to Design mode and the table will display.



```

1 <oj-sp-general-overview-page id="oj_gop1" display-options.responsive-padding="on"
2   page-title='[[ "Order " + $variables.raaOrders.orderNumber ]] '
3   contextual-info='[[ [{"label": "Customer", "value": $variables.raaOrders.customerFirstName}]]'
4   <div slot="main">
5     <!-- Main Content goes here... It is recommended to have a heading element title a>
6     <oj-bind-if test="[[ ($variables.selectedOrderId) !== undefined ]]">
7       <oj-form-layout readonly="true" columns="4">
8         <oj-input-text value="[[ $variables.raaOrders.customerIdObject.items[0]]"
9           |   readonly="true" label-hint="CustomerName"></oj-input-text>
10        <oj-input-text value="[[ $variables.raaOrders.customerIdObject.items[0]]"
11           |   readonly="true" label-hint="Email Address"></oj-input-text>
12        <oj-sp-input-phone-number label-hint="Input Phone Number"
13           |   value.subscriber-number="[[ $variables.raaOrders.customerIdObject.it]>
14         </oj-form-layout>
15       </oj-bind-if>
16       <div class="oj-sm-margin-10x-start">
17         <oj-vb-fragment bridge="[[vbBridge]]" name="shippingaddress">
18           <oj-vb-fragment-param name="city" value="[[ $variables.raaOrders.shippi]>
19           <oj-vb-fragment-param name="country" value="[[ $variables.raaOrders.sh]>
20           <oj-vb-fragment-param name="state" value="[[ $variables.raaOrders.ship]>
21           <oj-vb-fragment-param name="street_1" value="[[ $variables.raaOrders.s]>
22           <oj-vb-fragment-param name="street_2" value="[[ $variables.raaOrders.s]>
23           <oj-vb-fragment-param name="postalCode"
24             |   value="[[ $variables.raaOrders.shippingAddressPostalCode ]]"></oj-vb->
25         </oj-vb-fragment>
26       </div>
27     </div>
28     <oj-table scroll-policy="loadMoreOnScroll" slot="main" data="[[ $variables.orderLinesColl]>
29       columns='[{"headerText": "ProductName", "field": "productName"}, {"headerText": "ProductNum}>
30       <template slot="multiFieldCellTemplate2">
31         <a href="#" target="_blank" class="oj-link">
32           <oj-bind-text value="[[ $current.row.productNumber ]]"></oj-bind-text>
33         </a>
34       </template>

```

4. The image does not appear correctly. Select the **Image** component in the Structure pane to fix this.

General Overview Page Template

Order F000000146

Your Page Subtitle here

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

CustomerName Norma Robinson	EmailAddress nrobinson@internalmail.co m	Input Phone Number +1 650 506 0009		
Shipping Address				
909 Chocolate Ave 18op959 90747 Table US				
ProductName	ProductNumber	ImageUrl	ListPrice	Quantity
Silicon Smart Active Smartwatch, Pink	RWD000403		\$249.00	6
Silicon Smart Active Smartwatch, Pink	RWD000403		\$249.00	9
AV Noise Cancelling Headphones	RWD001001		\$199.00	5
Green Gate LaserJet Printer	RWD007003		\$355.00	4
Oro Gold Wearable, White	RWD000204		\$149.00	5

General Overview Page Template

- div@main
- Bind If
- Form Layout
 - Input Text [[variables...customerName]]
 - Input Text [[variables...emailAddress]]
 - Input Phone Number
- div
 - shippingaddress
- Table@main
 - Cell Template@multiFieldCellTemplate2
 - Hyperlink (Bind Text)
 - Cell Template@multiFieldCellTemplate3
 - Image
 - Cell Template@multiFieldCellTemplate4

5. Click its **Data** tab. Replace the **Source URL** with:

```
[[ $application.path + 'resources/' + $current.row.imageUrl ]]
```

The screenshot shows the Oracle ADF Data Editor interface. On the left, there's a navigation bar with tabs: General, **Data**, Events, All. The **Data** tab is highlighted with a red border. To the right of the tabs is a large text input field labeled "Source URL". Inside the input field, the expression `[[$application.path + 'resources/' + $current.row.imageUrl]]` is entered and also highlighted with a red border. On the far right, there's a vertical "Properties" panel.

The images now display

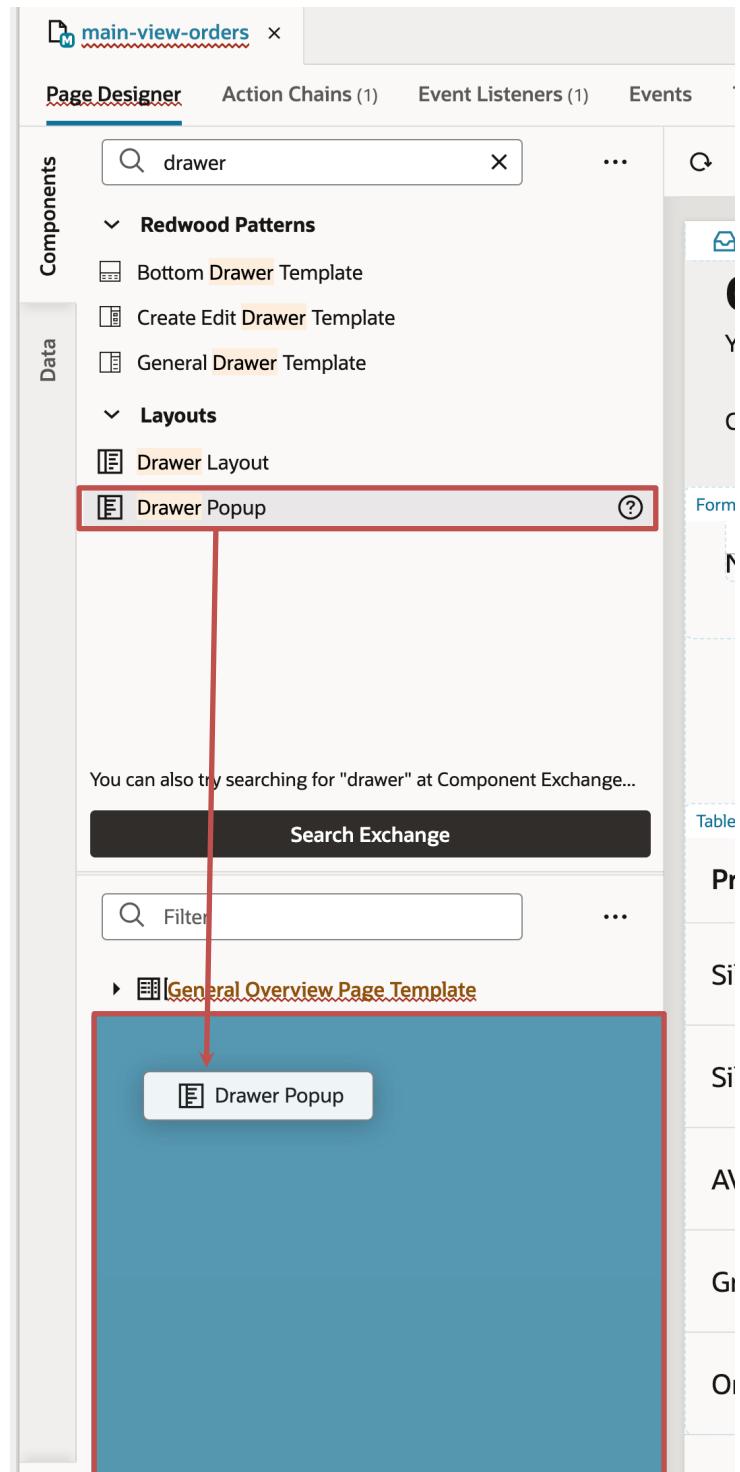
US						
Product Name	Product Number	Image URL	List Price	Minimum Order Quantity	Total Amount	
Oro Gold Wearable, White	RWD000204		149	5	745	
Silicon Smart Active Smartwatch, Pink	RWD000403		249	6	1494	
Silicon Smart Active Smartwatch, Pink	RWD000403		249	9	2241	
All Noise Cancelling Headphones	DWD001001		100	5	500	

The page is complete. Now, you need to add the drawer to display the product data for a selected product number.

Add a drawer component to display product data.

6. Drag a **Drawer Popup** component below the **General Overview Page Template** in the Structure pane.

Tip: Collapse the General Overview Page Template before adding the drawer.



- ▶ General Overview Page Template
- ▶ Drawer Popup

7. Select the **Drawer Popup** and enter `productDrawer` for its ID in its Property pane.

- ▶ General Overview Page Template
- ▶ **Drawer Popup**

Drawer Popup

...

General Events All

ID

fx (x)

productDrawer

Properties

8. Set the **Edge** property to end.

Edge

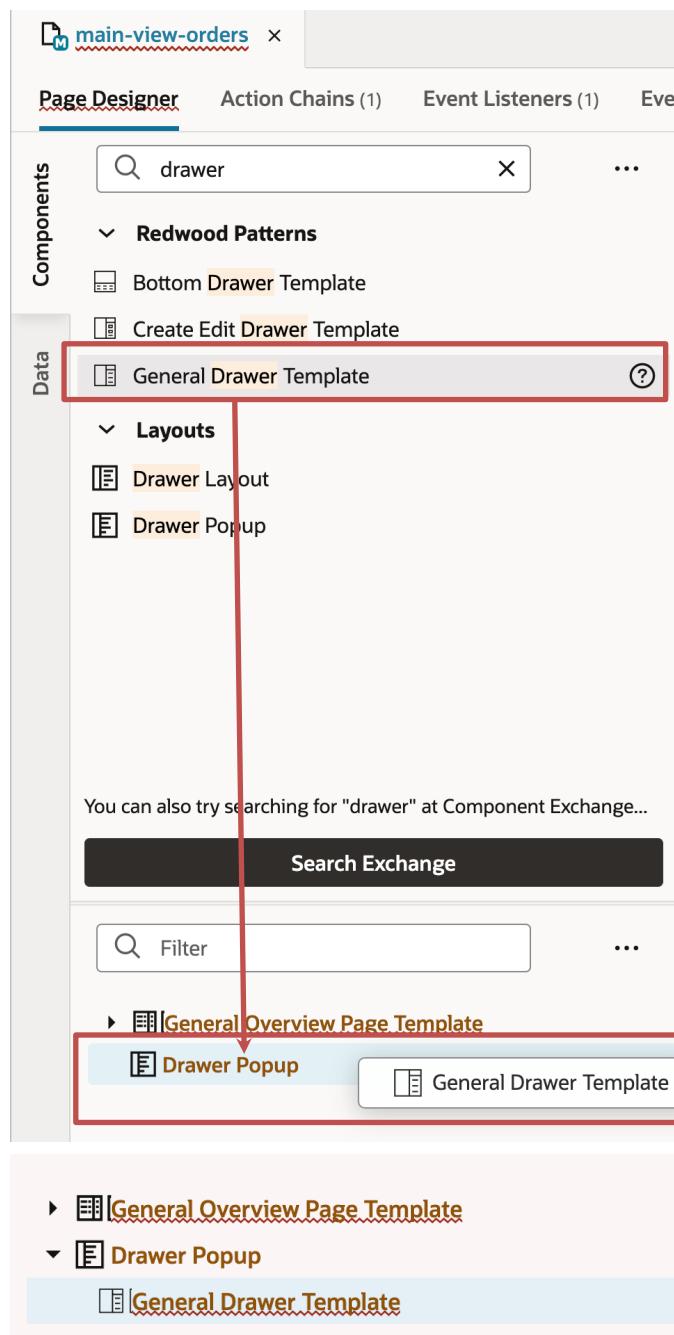
end

start

bottom

9. Drag a **General Drawer Template** component onto the **Drawer Popup**. This provides the layout area functionality.

Note: The drawer does not display by default. You will resolve that in later steps.



- With the **General Drawer Template** selected, set its **Drawer Title** property to **Product detail**. Set its **Drawer Subtitle** property to **1 product**.

The screenshot shows the 'General Drawer Template' properties panel. The 'General' tab is selected. The 'Drawer Title' field contains 'Product detail' and the 'Drawer Subtitle' field contains '1 product', both of which are highlighted with a red border.

You can work with the Drawer Popup open (displayed) for development by setting the **Drawer Popup Opened Temporary Override** property to **True**. You'll create a variable and event to programmatically manage the drawer opening and closing later. For now, you will add the needed fields for its contents.

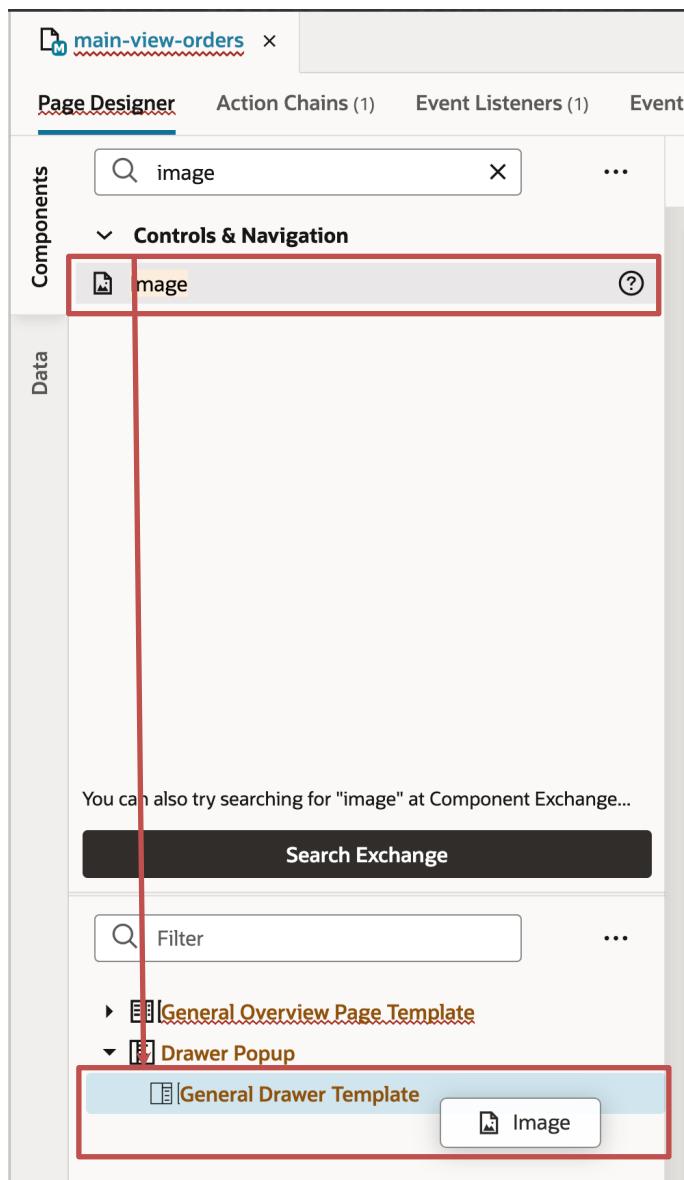
- Select the **Drawer Popup** and set its **Temporary Override** to **True**. The drawer displays.

The screenshot shows the page structure. The 'General Overview Page Template' is expanded, revealing its components. The 'Drawer Popup' component is highlighted with a red border.

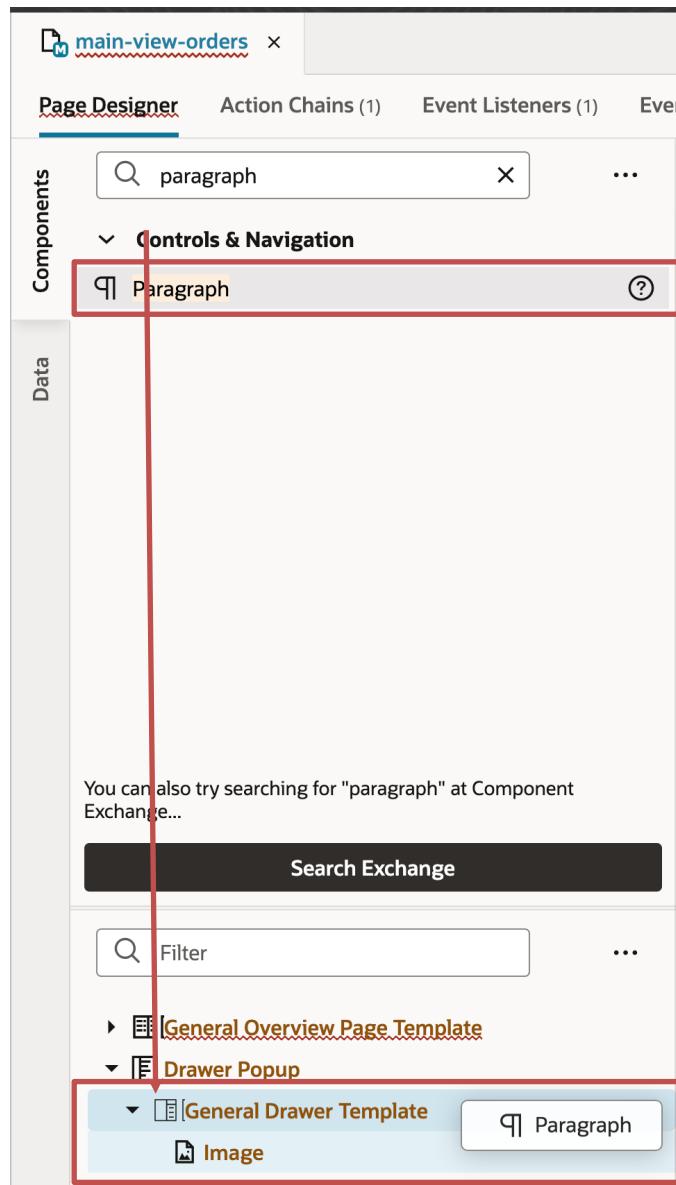
The screenshot shows the 'Drawer Popup' properties panel. The 'General' tab is selected. The 'Temporary Override' property is set to 'True', which is highlighted with a red border.

- Add an **Image** component to the **General Drawer Template > Default slot**.

Hint: The drawer tends to close after adding things to its children e.g., General Drawer Template. It may be easier to complete the following instructions and add all the necessary components, and then click the **True** setting.



13. Add a **Paragraph** control to the **General Drawer Template** > **Default slot**. This will help with formatting.



14. Add a second **Paragraph** below the first; drop it on the **General Drawer Template > Default slot** and it will be appended.

The screenshot shows the Oracle ADF Page Designer interface. At the top, there's a header bar with tabs for 'Page Designer' (which is selected), 'Action Chains (1)', 'Event Listeners (1)', and 'Events'. Below the header is a search bar with the placeholder 'paragraph' and a magnifying glass icon. To the right of the search bar is a three-dot menu icon. On the left side, there's a sidebar with sections for 'Components' (which is expanded) and 'Data'. Under 'Components', the 'Controls & Navigation' section is expanded, showing a list of components. One component, 'Paragraph', is highlighted with a red border. A vertical red line with an arrow points from this 'Paragraph' component down to the 'General Drawer Template' component in the component tree below. The main workspace shows a message: 'You can also try searching for "paragraph" at Component Exchange...'. Below this message is a 'Search Exchange' button. Further down is a 'Filter' search bar. The component tree on the right shows a hierarchy: 'General Overview Page Template' (expanded), 'Drawer Popup' (expanded), 'General Drawer Template' (highlighted with a red border), 'Image' (under General Drawer Template), and two 'Paragraph' components (both labeled '(Paragraph)'). The bottom portion of the screenshot shows a preview pane with the same component structure, where the second 'Paragraph' component is visible below the first.

15. Drag a **Text** component onto the **first Paragraph** control.

main-view-orders x

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

Components

Text

Controls & Navigation

Highlight Text

Forms & Inputs

Email

Input Rich Text 2

Input Text

Phone

Text Area

URL

Collections

List View

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

Search Exchange

Filter

General Overview Page Template

Drawer Popup

General Drawer Template

Image

Paragraph (Paragraph)

Text

Paragraph (Paragraph)

Drawer Popup

General Drawer Template

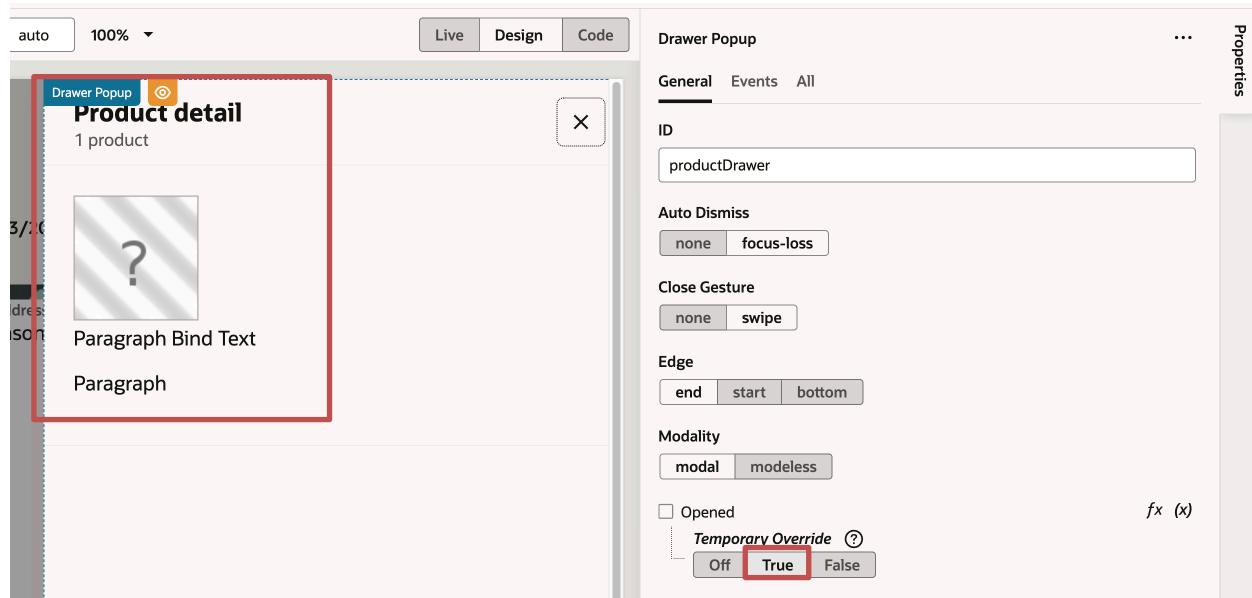
Image

Paragraph (Paragraph)

Bind Text (Bind Text)

Paragraph (Paragraph)

16. Select the Drawer Popup and set the Drawer Popup Temporary Override property to True to display the drawer.



17. Select the **Image** and click its **Data** tab. Enter this for Source URL:

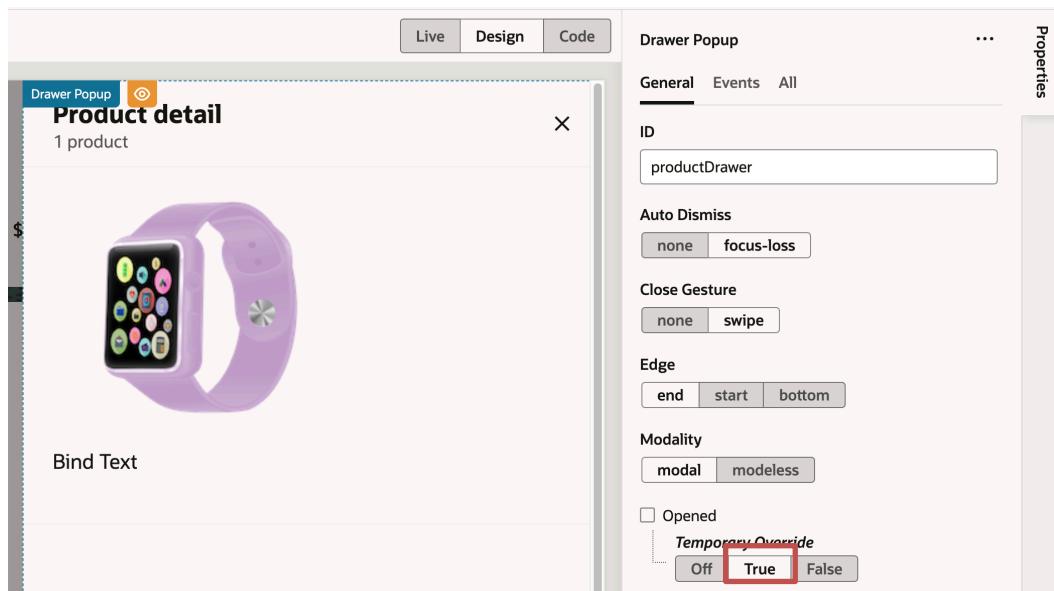
```
[[ $application.path + 'resources/' +
$variables.rraOrders.orderLinesCollection.items[0]?.imageUrl ]]
```

The screenshot shows the Oracle ADF Designer interface. On the left, there is a visual representation of a 'Image' component within a 'Drawer Popup' container. The image shows a smartwatch with a purple strap. Below the image, there is some text: 'Paragraph Bind Text' and 'Paragraph'. On the right, the 'Properties' panel is open for the 'Image' component. Under the 'Data' tab, there is a 'Source IRI' field containing the expression: [[\$application.path + 'resources/' + \$variables.rraOrders.orderLinesCollection.items[0]?.imageUrl]]. This expression is highlighted with a red box. Other tabs in the properties panel include 'General' and 'Events'.

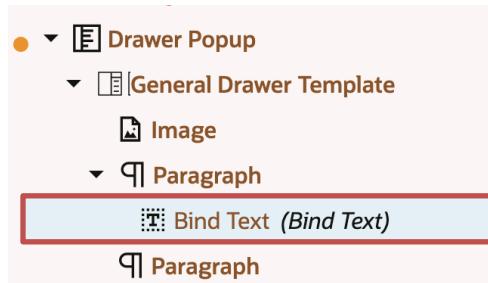
18. Click the **Code** button and remove the text “Paragraph” from both paragraph elements.

```
47 <div slot="info">
48 | | | <!-- Complementary Content goes here... It is recommended to have a heading element title as well -->
49 <oj-drawer-popup></oj-drawer-popup>
50 </oj-sp-general-overview-page>
51 <oj-drawer-popup id="productDrawer" edge="end">
52   <oj-sp-general-drawer-template drawer-title="Product detail" drawer-subtitle="1 product">
53     Paragraph
56       | <oj-bind-text value="Bind Text"></oj-bind-text>
57     </p>
58     <p class="oj-p">Paragraph</p>
59   </oj-sp-general-drawer-template>
60 </oj-drawer-popup>
61
~~
47 <div slot="info">
48 | | | <!-- Complementary Content goes here... It is recommended to have a heading element title as well -->
49 <oj-drawer-popup></oj-drawer-popup>
50 </oj-sp-general-overview-page>
51 <oj-drawer-popup id="productDrawer" edge="end">
52   <oj-sp-general-drawer-template drawer-title="Product detail" drawer-subtitle="1 product">
53     
56       | <oj-bind-text value="Bind Text"></oj-bind-text>
57     </p>
58     <p class="oj-p"></p>
59   </oj-sp-general-drawer-template>
60 </oj-drawer-popup>
```

19. Return to **Design** mode. Set the **Temporary Override** property for the **Drawer Popup** to **True** to open it again.

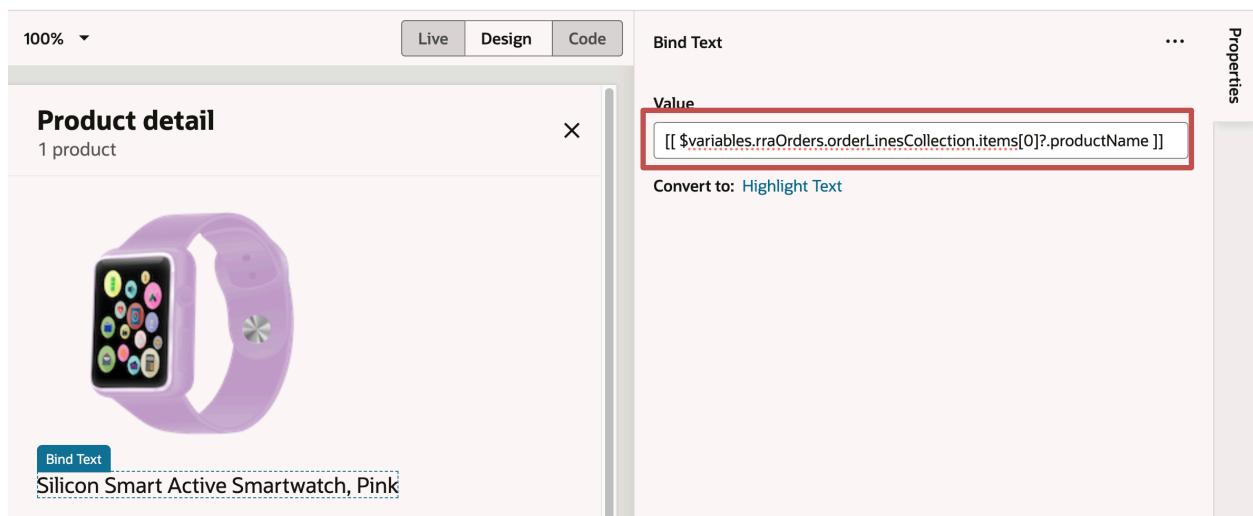


20. Expand the first **Paragraph** element and select the **Bind Text**.



21. Change the **Value** for the **Bind Text** to:

```
[[ $variables.rraOrders.orderLinesCollection.items[0]?productName ]]
```



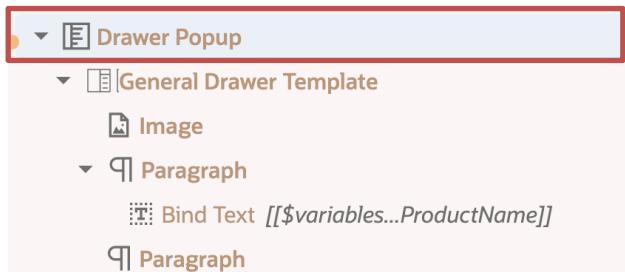
22. Lastly, add a variable that will control opening and closing the drawer. Click the **Variables** tab.



23. Click **+ Variable** to create a new variable named `isDrawerOpen` of type **Boolean**. Since the drawer is closed by default, set its **Default Value** to `false`.

A screenshot of the Variable creation dialog. The General tab is selected. The ID field contains `isDrawerOpen`, which is highlighted with a red box. The Type field contains `Boolean`, also highlighted with a red box. The Default Value field contains `false`, which is highlighted with a red box. A warning message below the ID field says: "⚠ Variable 'isDrawerOpen' is defined but not used".

24. Return to the **Page Designer** tab and select the **Drawer Popup**.



25. Bind the **Opened** property to the **isDrawerOpen** variable and click **Off** to close the drawer.

The screenshot shows the 'Variables' dialog box. On the left, there's a tree view with 'Page' expanded, showing 'Variables'. Under 'Variables', there are two entries: 'contextualInfo' and 'isDrawerOpen'. The 'isDrawerOpen' entry is highlighted with a red box. On the right, the 'fx (x)' panel shows the expression `{{ \$variables.isDrawerOpen }}` and a 'Temporary Override' section with 'Off' selected.

This screenshot shows the 'Opened' property configuration. The expression `{{ \$variables.isDrawerOpen }}` is shown in the 'Value' field. Below it, the 'Temporary Override' section is visible, with the 'Off' button also highlighted with a red box.

26. Select **Hyperlink** in the table. Click its **Data** tab. Delete the URL #.

The screenshot shows the 'Data' tab for a 'Hyperlink' component. The component is part of a 'Cell Template' under a 'Table' structure. The 'URL' field is empty, indicated by a red box. The entire 'Hyperlink' component is also highlighted with a red box.

27. Add an event to change the **isDrawerOpen** to **true** when the link is clicked. Click its **Events** tab and create a new **On 'click'** event listener. The action chain opens.



28. Drag an **Assign Variable** action onto the canvas. With it selected, choose **isDrawerOpen** for the **Variable** and set its **Value** to `{{ !$page.variables.isDrawerOpen }}`, which returns the opposite of its current value.

Note the !



29. Return to **Page Designer**. Select the **General Drawer Template** and click its **Events** tab. Add an event that will close the drawer when the Close icon is clicked.

The screenshot shows the Oracle Page Designer interface. In the left sidebar, under 'General Overview Page Template' and 'Drawer Popup', the 'General Drawer Template' item is selected and highlighted with a red border. In the top navigation bar, the 'Events' tab is also highlighted with a red border. Below the sidebar, the title 'General Drawer Template' is visible, followed by tabs for 'General', 'Events' (which is active), and 'All'.

30. Add a new **On 'spClose'** event listener. The action chain opens.

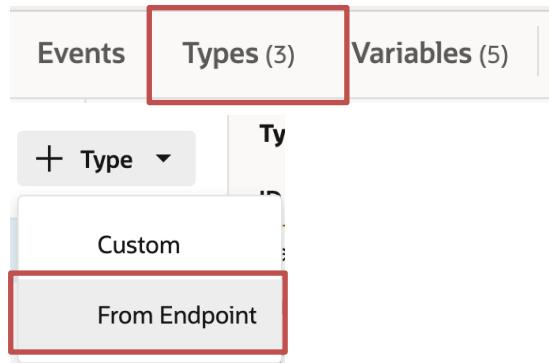
The screenshot shows the 'Event Listener' dialog box. It lists several event options: 'On 'spPrimaryAction'', 'On 'spSecondaryAction'', 'On 'spClose'' (which is highlighted with a red box), 'On 'spPrevious'', 'On 'spNext'', and '+ Create'. The '+ Create' option is preceded by a plus sign.

31. Add an **Assign Variable** action to the canvas. Select **isDrawerOpen** for the **Variable** and **false** for the **Value**.

The screenshot shows the 'Assign Variable' dialog box. The 'Variable' field is set to 'isDrawerOpen' (highlighted with a red box). The 'Value' field is set to 'false' (also highlighted with a red box). A 'Properties' panel is visible on the right side of the dialog.

```
(→) $page.variables.isDrawerOpen = false
```

32. Return to **Page Designer** to test the drawer.
33. Enter **Live** mode and click one of the Product Number links.
Notice the drawer opens and closes when you click the X, but it does not display the correct product info in the drawer. It only shows the first record's data from the table. To fix this, you'll add a row selection event for the table that will load that row's data into a variable and reference that data in the drawer.
34. Return to Design mode.
35. Start by clicking the **Types** tab to define a new type as the basis for a variable that will hold the selected table row data. Create a new type **From Endpoint**.



36. Expand **Business Objects > rraOrders > orderLinesCollection** and select **Get /[...]orderLinesCollection/{rraOrders_orderLinesCollection_Id}** to define a single object type and click **Next**.

Create Type From Endpoint

Select an endpoint to create a type.

 Filter

▼ Business Objects

▶ rraCustomers

▶ rraOrderLines

▼ rraOrders

GET /rraOrders

POST /rraOrders

GET /rraOrders/{rraOrders_Id}

PATCH /rraOrders/{rraOrders_Id}

DELETE /rraOrders/{rraOrders_Id}

▶ customerIdObject

▼ orderLinesCollection

GET /[...]/child/orderLinesCollection

POST /[...]/child/orderLinesCollection

GET /[...]/orderLinesCollection/{rraOrders_orderLinesCollection_Id}

PATCH /[...]/orderLinesCollection/{rraOrders_orderLinesCollection_Id}

DELETE /[...]/orderLinesCollection/{rraOrders_orderLinesCollection_Id}

37. Select **imageUrl**, **productName**, and **productNumber** and click **Finish**.

Create Type From Endpoint

Define Type

Select attributes from the endpoint you want to use in your application.

get_rraOrders_orderLinesCollection

Create From Endpoint Root

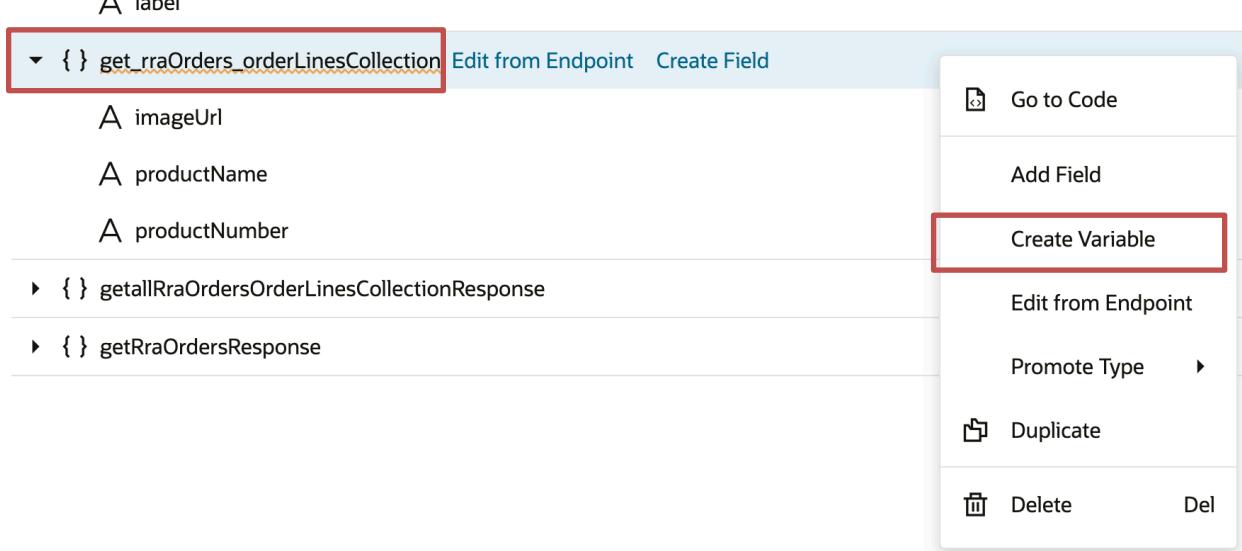
Endpoint Structure

- A createdBy
- A creationDate
- A currencyCode
- A deliveredDate
- # id
- A imageUrl
- A lastUpdateDate
- A lastUpdatedBy
- ▶ [] links
 - # listPrice
 - # objectVersionNumber
 - # objectVersionNumber1
 - # orderId
 - # orderLineId
 - # productId
 - # productId1
 - A productName
 - A productNumber
- # quantity

Type

- A imageUrl
- A productName
- A productNumber

38. Right-click the new type and select **Create Variable**.



39. The **Variables** tab opens. Rename **get_rraOrders_orderLinesCollectionVar** to **rraLineItem**.

The Variables tab shows a single variable entry:

- ID ***: rraLineItem (highlighted with a red box)
- Type ***: get_rraOrders_orderLinesCollection
- A tooltip message: 'Variable 'get_rraOrders_orderLinesCollectionVar' is defined but not used'

40. Return to the **Page Designer**. Select the **Table** and click its **Events** tab. Add a new **On 'First Selected Row'** event listener. The action chain opens.

The screenshot shows two panels. The top panel displays the hierarchical structure of a page template named "General Overview Page Template". It includes sections for "Bind If", "Form Layout", and a "shippingaddress" section which contains a "Table@main" component. The "Table@main" component is highlighted with a red box. The bottom panel shows the "Events" tab of the "Table" configuration, with the "Event Listener" dropdown menu open. An item in the dropdown, "On 'First Selected Row'", is also highlighted with a red box.

```

- General Overview Page Template
  - Bind If
    - Form Layout
      - Input Text [[${variables...customerName}]]
      - Input Text [[${variables...emailAddress}]]
      - Input Phone Number
  - shippingaddress
    - Table@main
      - Cell Template @multiFieldCellTemplate2
        - Hyperlink (Bind Text)
      - Cell Template @multiFieldCellTemplate3
        - Image
      - Cell Template @multiFieldCellTemplate4

```

Table

General Data Events All Quick Start

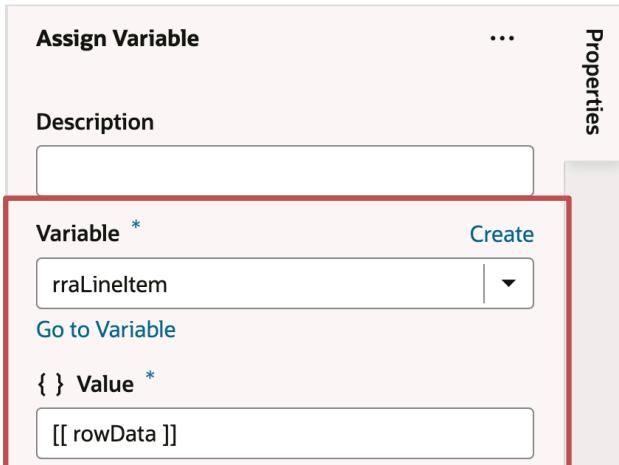
+ Event Listener ▾

- On 'First Selected Row'
- On 'Select Multiple Rows'
- On 'ojBeforeRowEdit'
- On 'ojBeforeRowEditEnd'
- + Create

41. Add an **Assign Variable** action to the canvas. Choose **rraLineItem** for **Variable** and ***rowData** for **Value**. The rowData is the data from the row where the hyperlink was clicked. VBS will automatically copy data from the selected rowData into the rraLineItem variable's fields. You'll use this to populate the drawer's fields.

The screenshot shows the configuration of an **Assign Variable** action in Oracle ADF Faces Designer. The action is named "Assign Variable".

- Description:** An empty text input field.
- Variable ***: A dropdown menu showing the following options:
 - [] contextualInfo
 - { } contextualInfo[0]
 - [] isDrawerOpen
 - SOP orderLinesCollectionListSDP
 - { } rraLineItem **(highlighted with a red box)**
 - { } rraOrders
 - A selectedOrderId
- Properties** panel: Shows a validation error message: "Must have required property 'variable'".
- Variables** panel:
 - Variable ***: A dropdown menu showing "rraLineItem".
 - Value ***: A dropdown menu showing "* rowData" **(highlighted with a red box)**.
 - Variables** section:
 - Filter input field.
 - Local section: "Create" button.
 - Input Parameters section: "Create" button.
 - * rowData **(highlighted with a red box)**
 - * rowKey
- Action** panel:
 - Assign Variable** icon.
 - Assign Variable** label.
 - Code** input field containing the VBS code: `{ } \$page.variables_rraLineItem = rowData` **(highlighted with a red box)**.



42. Return to the **Page Designer** to assign the new data elements to the existing fields. Select **Image** under **General Drawer Popup**.



43. Click its **Data** tab and replace the existing code with:

```
[[ $application.path + 'resources/' +
$variables_rraLineItem.imageUrl ]]
```



44. Select the **Bind Text** and click its **Data** tab. Replace the existing code with:

```
[[ $variables_rraLineItem.productName ]]
```



Bind Text

...

Properties

Value

[[\$variables.rralLineItem.productName]]

Convert to: [Highlight Text](#)

45. Test this in **Live** mode. Click a **Product Number** link. The drawer opens and displays the correct product image and name.

Order F000000146
Your Page Subtitle here

Customer Norma Robinson Date 8/13/2022

CustomerName Norma Robinson EmailAddress nrobinson@i.com

Shipping Address
909 Chocolate Ave
18op959
90747
US

ProductName ◇

ProductName	Action
Silicon Smart Active Smartwatch, Pink	R
Silicon Smart Active Smartwatch, Pink	R
AV Noise Cancelling Headphones	R
Green Gate LaserJet Printer	R
Oro Gold Wearable, White	R

Product detail
1 product

Oro Gold Wearable, White

The screenshot shows the Oracle ADF Page Designer interface. The main window is titled "Order F000000146" and contains a subtitle "Your Page Subtitle here". It displays customer information: Customer Norma Robinson and Date 8/13/2022. Below this, there is a table with columns "ProductName" and "Price". The table lists several products: Silicon Smart Active Smartwatch, Pink; Silicon Smart Active Smartwatch, Pink; AV Noise Cancelling Headphones; Green Gate LaserJet Printer; and Oro Gold Wearable, White. A modal window titled "Product detail" is overlaid on the main window, showing an image of "AV Noise Cancelling Headphones" and its product name.

46. Return to Design mode.

Before you test this all together, you must add one thing to the **main-manage-orders** page. Do you remember what that is? You didn't pass the selected order number to the main-view-orders page. At the time you created the navigation to the main-view-orders page, it didn't have any input parameters defined. You'll resolve that now.

47. Open the **main-manage-orders** page.

The screenshot shows the Oracle ADF Page Designer interface with the "main-manage-orders" page selected in the left sidebar. The page title is "Manage Orders" and it includes a subtitle "Page Subtitle". There is a search bar with placeholder text "Try a customer last name or order number". Below the search bar, there is a processing message "Judy Evans PROCESSING". The right side of the screen shows the page content area.

48. Select the **List Item Template** and click its **Events** tab and click the link for the event to access the existing action chain. The action chain opens.

The screenshot shows the Oracle ADF interface for configuring a List Item Template. The top part shows the template structure with a 'Smart Search Page Template' containing a 'Collection Container' which includes a 'List View' and a 'Row Template @itemTemplate (itemTemplate)'. The 'List Item Template' is selected and highlighted with a red border. Below this, the 'Properties' pane is open, showing the 'Events' tab selected. An action chain named 'ListItemTemplateSpSecondaryActionChain' is defined, with the 'Event' set to 'spSecondaryAction' and the 'Action Chain' also set to 'ListItemTemplateSpSecondaryActionChain'. The input parameters for this action chain are listed: 'actionId' ({{ \$event.detail.actionId }}), 'current' ({{ \$current }}), 'index' ({{ \$current.index }}), and 'key' ({{ \$current.key }}).

49. Select **toMainViewOrders**. Notice the **selectedOrderId** variable appears in the Property pane with no value.

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

The screenshot shows the action chain code within the Oracle ADF interface. It consists of a single case statement within a switch block. The case is triggered by the value 'order'. Inside the case block, the variable 'toMainViewOrders' is assigned the value 'Navigate To Page main-view-orders'. The code concludes with a 'break' statement to exit the switch block.

Page * Create
main-view-orders
Go to Page
Parameters
A selectedOrderId *

50. Bind the **selectedOrderId** to the ***key**.

Parameters
A selectedOrderId * fx (x)

Variables
Filter
Local Create
Input Parameters Create
A actionId
* current
index
* key
Page Create
Variables
[] allFilters Edit
[] appliedFilters Edit

Parameters
A selectedOrderId * fx
[[key]]

51. Return to the **Page Designer** tab.

52. Now test that the Manage Orders page and View Orders page work together. Close the main-view-orders page. Enter Live mode for the main-manage-orders page and select the ... button to view an order. The main-view-orders page opens and displays the details for that order. Enter Live mode. Then, select a row in the table and click its Product Number link and the drawer opens, displaying the correct product information.

Order F000000314

Your Page Subtitle here

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

CustomerName Kathryn Rogers	EmailAddress krogers@internalmail.com	Input Phone Number +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

53. Return to Design mode.
54. Close all open tabs.

This completes this practice.

Practice 3-3: Create the View Customers Page

Overview

In this practice, you create the page to view customer details. You display summary information for the customer as well as a list of orders for that customer.

The screenshot shows a customer profile for "Rogers". On the left, there is a sidebar with the customer's name and contact information:

- FirstName: Kathryn
- LastName: Rogers
- Phone: 6505,060,019
- EmailAddress: krogers@internalmail.com
- ShippingAddressCity: Vision City
- ShippingPostalCode: 95,727

On the right, there is a list of orders:

Status	Order ID	Date	Total Amount
DRAFT	F000000313	8/7/2022	\$5,151.00
DRAFT	F000000314	8/11/2022	\$1,664.00
DRAFT	F000000315	8/21/2022	\$4,481.00
PROCESSING	F000000300	6/2/2022	\$5,311.00
PROCESSING	F000000301	6/5/2022	\$3,786.00
PROCESSING	F000000303	6/16/2022	\$4,335.00
PROCESSING	F000000304	6/19/2022	\$7,494.00

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.

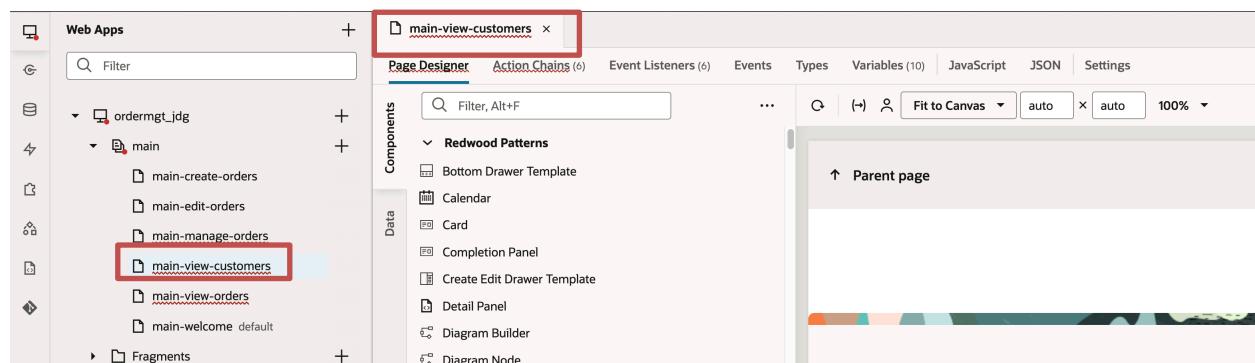
The screenshot shows the Visual Builder Studio interface. On the left is a sidebar with icons for Organization, Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, Builds, Releases, and Environments. The 'Workspaces' icon is highlighted. The main area is titled 'Visual Builder Studio' and 'Organization'. Below it is a navigation bar with tabs: Projects (which is active), OCI Account, Build Executor Templates, Build Executors, Component Exchange, and a gear icon. A search bar and filter buttons for Member, Favorites, Owner, Shared, and All are also present. The main content area shows 'Projects: 3'. A table lists three projects: 'OrderMgt_JDG' (Current, Order Management Applic...), 'ordermgt_jdg.git' (ordermgt_jdg.git), and 'Order_Mgt' (Order_Mgt). The 'OrderMgt_JDG' row is highlighted with a red box around its name.

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

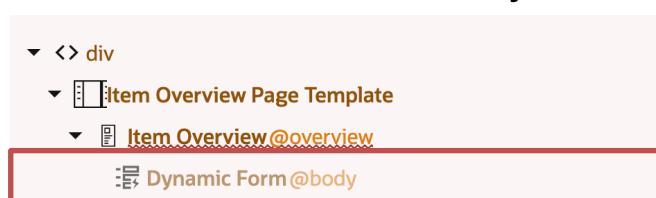
The screenshot shows the Visual Builder Studio interface. The sidebar on the left has the 'Workspaces' icon highlighted. The main area is titled 'Visual Builder Studio' and 'OrderMgt_JDG ▾ | Workspaces'. A search bar and filter buttons for Mine and Others are present. Below is a table with columns for Workspace, Repository, and Current Branch. It lists 'OrderMgt_JDG' (Visual Application), 'ordermgt_jdg.git' (ordermgt_jdg.git), and 'Order_Mgt' (Order_Mgt). A context menu is open over the 'OrderMgt_JDG' row, with the 'Open in New Tab' option 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.

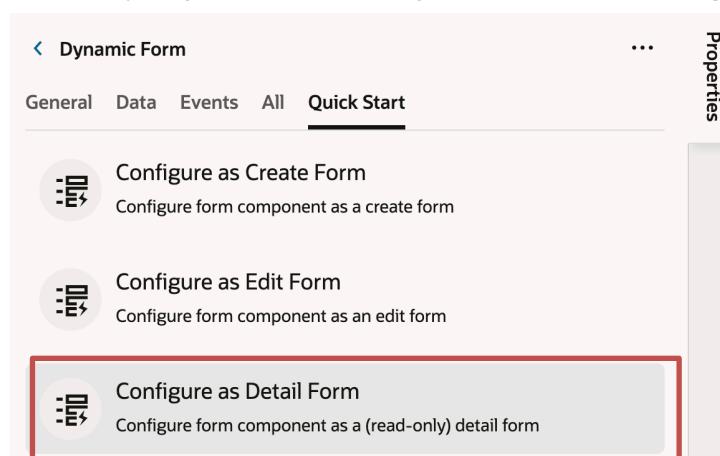
4. Select the **Web Apps** tab in the Navigator. Expand **ordermgt_<your initials>**. Expand **main**. Select the **main-view-customers** page. It opens in Page Designer.



5. Look at the **Structure** pane. Notice this page template is preconfigured to use a dynamic form for the overview area. Click the **Dynamic Form@body** element.



6. In its Property pane, click the **Quick Start** tab. Quick Starts are, as the name suggests, a quick way to get started building the form. Click **Configure as Detail Form**.



7. Select **rraCustomers** and click **Next**.



8. Enter a **Label** of `view_customer` and click **Select fields to display**.

The screenshot shows a configuration form. It includes fields for 'Label *' (containing 'view_customer'), 'ID *' (containing 'viewCustomer'), and 'Description' (an empty text area). Below these fields is a section titled 'Use Simple Layout' containing a button labeled 'Select fields to display >'. A red box highlights the 'Select fields to display' button.

9. Deselect all the fields except **firstName**, **lastName**, **phone**, **emailAddress**, **shippingAddressCity**, and **shippingPostalCode**.

The screenshot shows a user interface for selecting fields. On the left, a vertical list of fields is displayed with checkboxes. On the right, a panel titled "Select fields to display" lists the chosen fields.

Fields List:

- A lastUpdatedDate
- A lastUpdatedBy
- # objectVersionNumber
- # objectVersionNumber1
- # objectVersionNumber2
- # phone
- # primaryBillingAddressId
- # primaryShippingAddressId
- A shippingAddressCity
- A shippingAddressCountry
- A shippingAddressLine1
- A shippingAddressLine2
- A shippingAddressState

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

A red box highlights the following selected fields:

- A firstName
- A lastName
- # phone
- A emailAddress
- A shippingAddressCity
- # shippingPostalCode

A blue bar labeled "Field" is visible at the bottom of the selected list.

10. As needed, drag the fields into the following order, using the  icon to drag fields into position. Note that a blue line indicates where the field will be positioned.

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

Field

- ... A lastName
- ... A firstName
- ... A lastName
- ... # phone
- ... A emailAddress
- ... A shippingAddressCity
- ... # shippingPostalCode

11. When done, it should look like this:

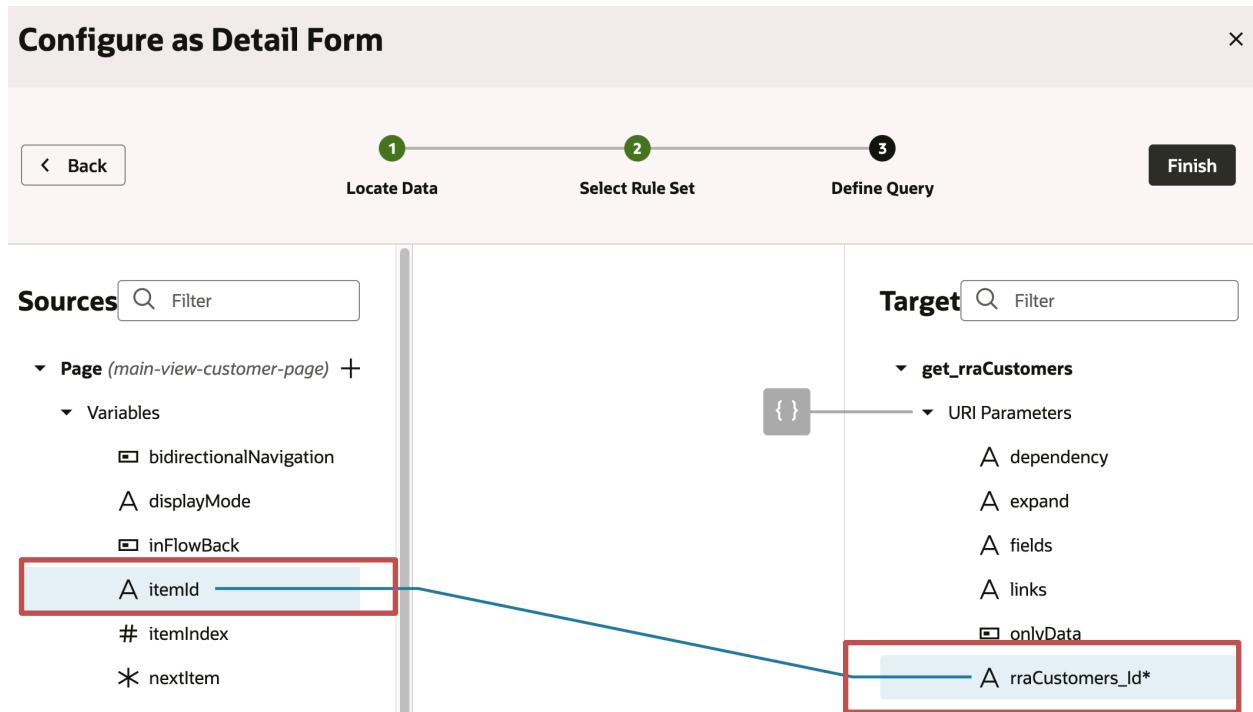
Select fields to display [Use template instead ↗](#)

Field

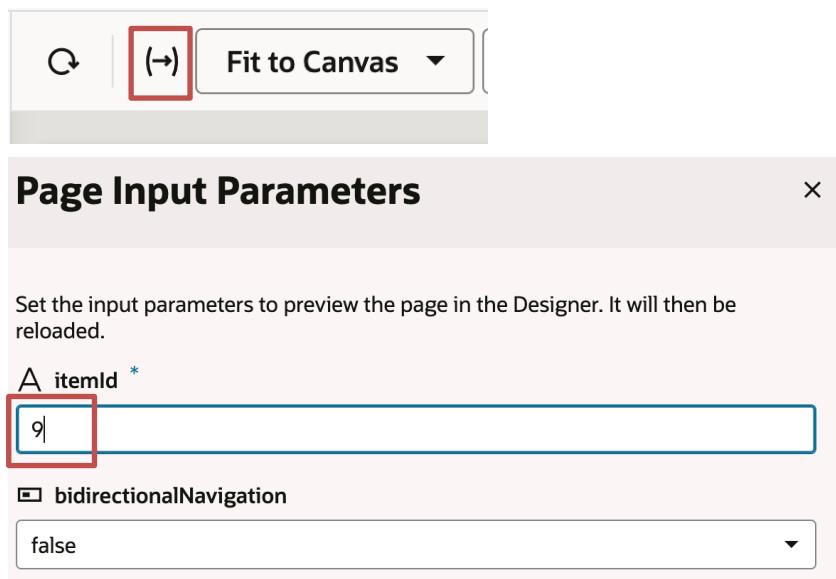
- ... A firstName
- ... A lastName
- ... # phone
- ... A emailAddress
- ... A shippingAddressCity
- ... # shippingPostalCode

12. Click **Next**.

13. Drag **itemId** onto **get_rraCustomers > rraCustomers_Id**. Click **Finish**.



14. You are returned to Page Designer. Click the **Page Input Parameters** button to confirm that there is a value for **itemId**. If there is not, enter 9.



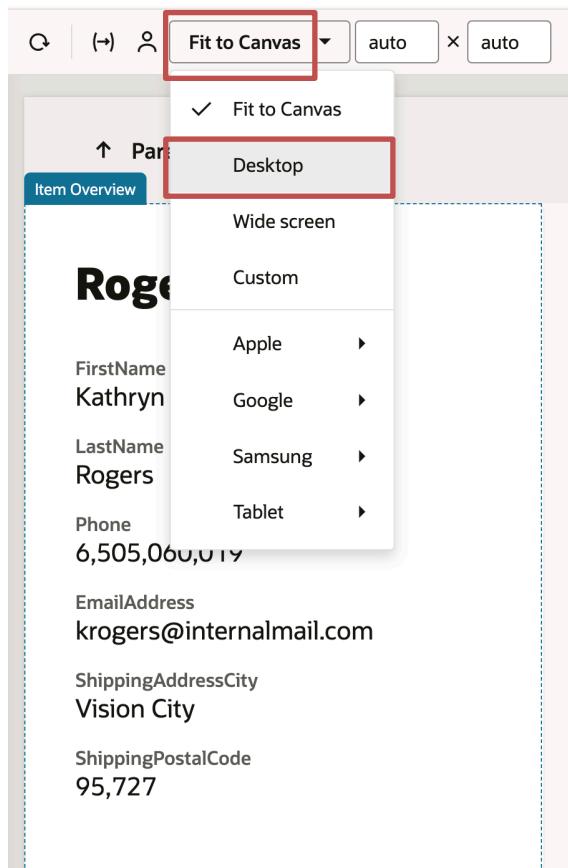
15. Select **Item Overview** in the **Structure** pane and paste this code into its **All > Title** property.

```
[[ $variables.rraCustomers.lastName ]]
```

The screenshot shows the Oracle ADF interface. In the center, there is a modal dialog titled "Item Overview". Inside the dialog, under the "Title" section, the value "[[\$variables.rraCustomers.lastName]]" is displayed in a text input field, which is highlighted with a red border. The "Title" field has an asterisk (*) indicating it is required.

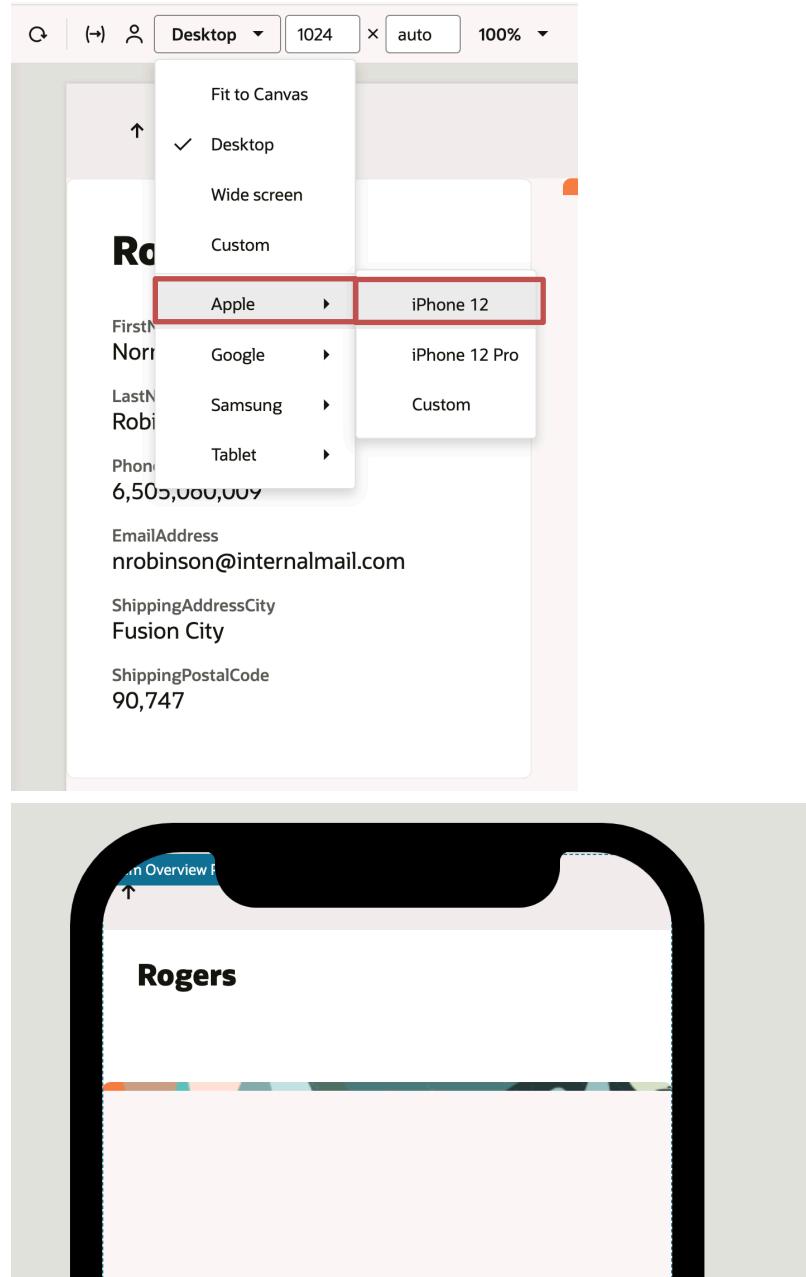
16. If you don't see any data displayed, close the Navigator pane. If you still don't see data, change the screen size to **Desktop**.

The screenshot shows the Oracle ADF Page Designer. On the left, the "Components" palette is open, showing categories like Redwood Patterns, Data, and Page Components. The "Data" category is selected, showing various data components. In the center, the "Page Designer" tab is active, showing a page template with a title "Rogers" and several data fields: FirstName (Kathryn), LastName (Rogers), Phone (6505,060,019), EmailAddress (krogers@internalmail.com), ShippingAddressCity (Vision City), and ShippingPostalCode (95,727). The "Item Overview" component is selected in the Structure pane on the left.

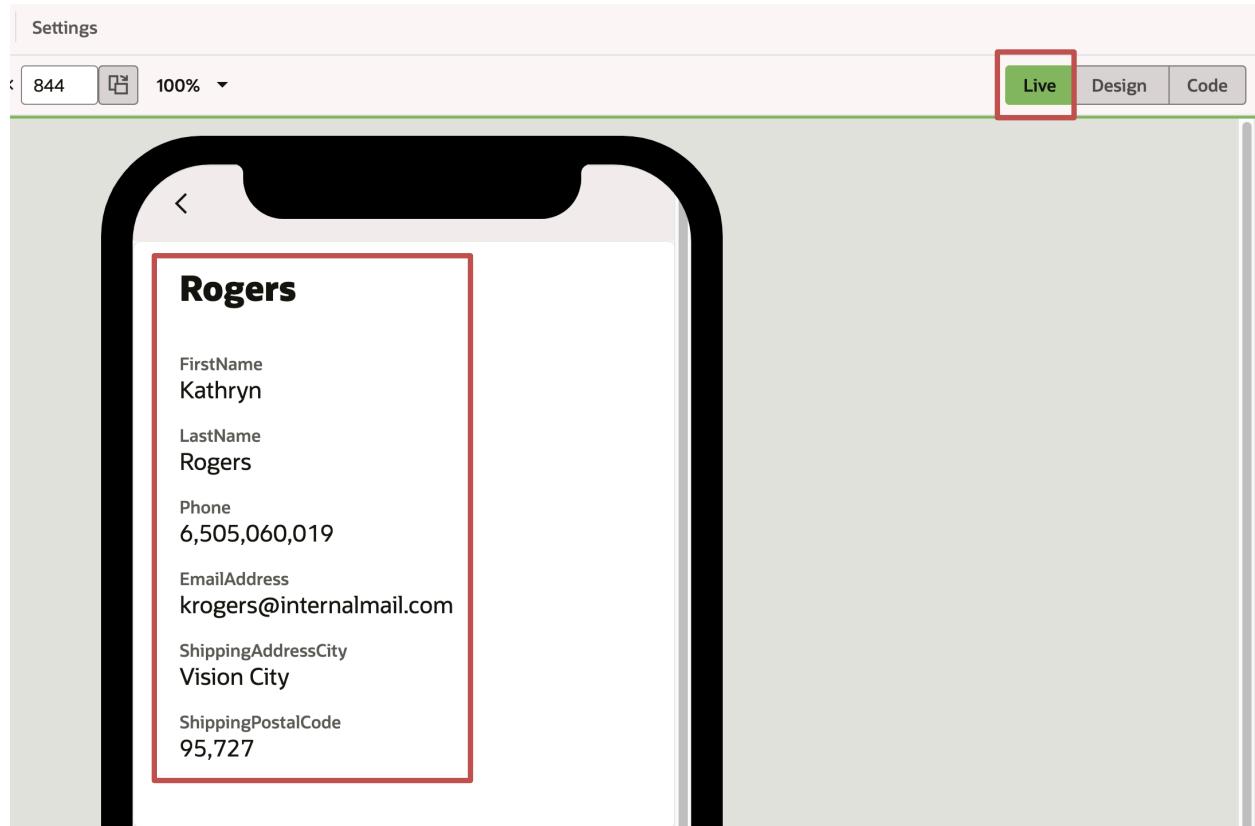


The screenshot shows the Oracle APEX 'Item Overview Page Template' page. At the top, there's a toolbar with icons for search, refresh, user profile, and a dropdown menu. The dropdown menu is open, showing 'Desktop' (selected), '1024', 'auto', and '100%'. Below the toolbar, the page content displays the record details. The 'FirstName' and 'LastName' fields are bolded. The 'Phone' field contains the value '6,505,060,019'. The 'Email' field contains 'krogers@internalmail.com'. The 'ShippingAddressCity' field contains 'Vision City'. The 'ShippingPostalCode' field contains '95,727'.

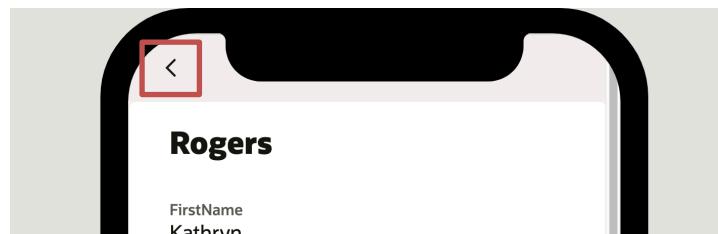
17. The page is sensitive to screen size. Change the screen size to **Apple iPhone 12**.



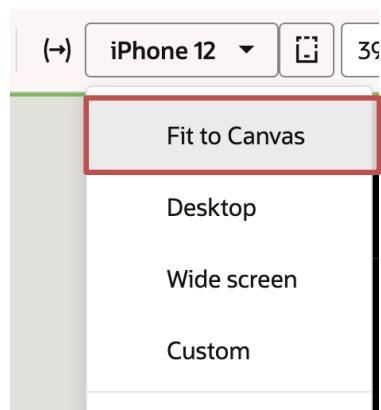
18. Enter **Live** mode and click the **customer's name**.



19. Click the < to collapse the contents again.



20. Change the screen size back to **Fit to Canvas**.



21. Return to Design mode when you are done testing.

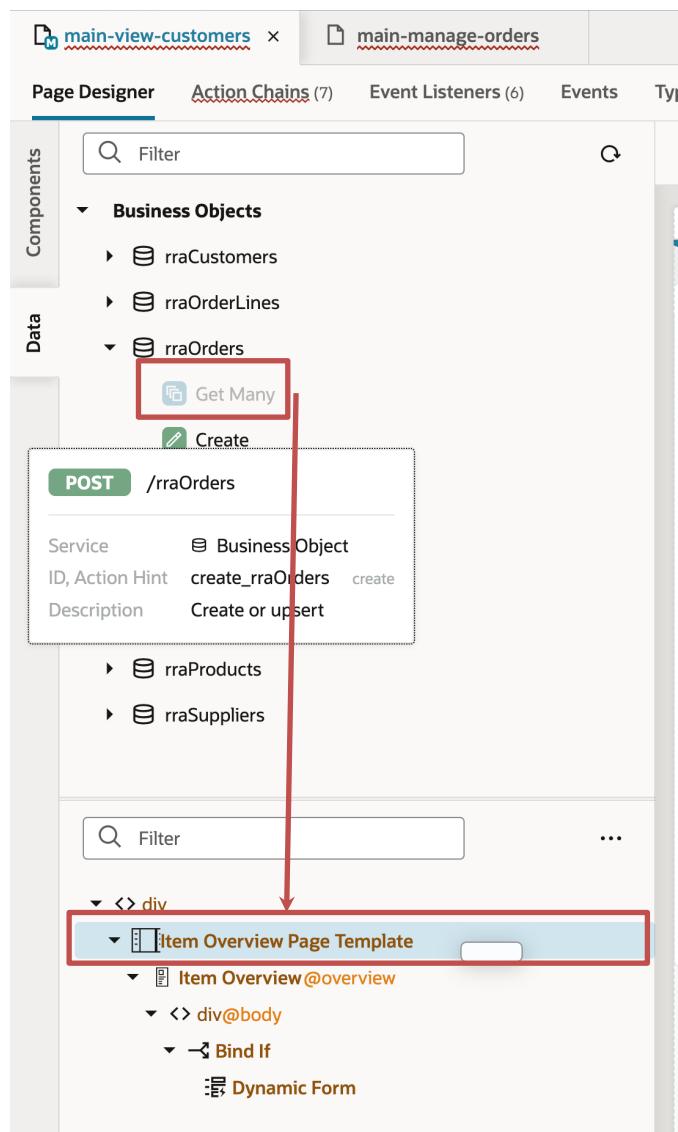
Note: You will notice both the phone number and postal code are not correctly formatted.

Since this a dynamic form, you would have to create templates for their formatting.

While this is not difficult, it is outside the scope of this course, as it has nothing to do with the configuration you are doing with Redwood. Dynamic components are covered in the Build Visual Applications Using Visual Builder Studio and Develop Fusion Apps Using Visual Builder Studio courses.

Configure the main area of the Item Overview page with order data for this customer.

22. Click the **Data** tab and expand **Business Objects > rraOrders** and drag **Get Many** onto the **Item Overview Page Template** element. Select **main slot** when prompted. Then choose to **Render as List**, not Dynamic, when prompted.



The screenshot shows the Oracle ADF Faces Designer interface. On the left, there's a tree view of the page template structure:

- <> div
- Item Overview Page Template
 - Item Overview@overview
 - <> div@body
 - Bind If
 - Dynamic Form

To the right, a modal dialog titled "Item Overview Page Template | Slot" is open, showing a list of slots:

- item overview slot
- main slot

The "main slot" is highlighted with a red border.

Below the tree view, another modal dialog titled "Item Overview Page Template" is open, showing the "Render as" options for the "Item Overview@overview" component:

- Table
- Dynamic
- List
- Table

The "List" option is highlighted with a red border.

23. Click **Next** to accept the **List Item Template**. Select the following fields in order and click **Next**.

- orderNumber
- orderDate
- orderStatus
- totalAmount – set it to Currency

Item Template Fields

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

The diagram illustrates the structure of Item Template Fields. At the top, there is a horizontal bar with five slots: **Leading Slot**, **Overline slot**, **Default slot**, **Metadata slot**, **Trailing Slot**, and **Action Slot**. Below this bar, the fields are organized into groups:

- Leading slot:** Contains the value "A orderNumber X" and a "Text" type selector.
- Overline slot:** Contains the value "A orderDate X" and a "Text" type selector.
- Default slot:** Contains the value "A orderStatus X" and a "Text" type selector.
- Secondary slot:** Contains the value "# totalAmount X" and a "Currency" type selector. This field is highlighted with a red border.
- Tertiary slot:** Not explicitly shown in the visual representation.

24. You need to filter the list of orders to return only those for this customer. Click the **filterCriterion** and set up the filter in the Builder section. 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

25. Choose **customerId** for the first attribute, select **contains (\$co)**, and then select **\$variables.itemId** for the second attribute. Click **Done** and **Finish**.

Include items Match All Match Any

IF

customerId

contains (\$co)

\$variables.itemId

{ } filterCriterion

Builder

Code

Include items Match All Match Any

IF

customerId contains \$variables.itemId

The list of orders for the customer displays.

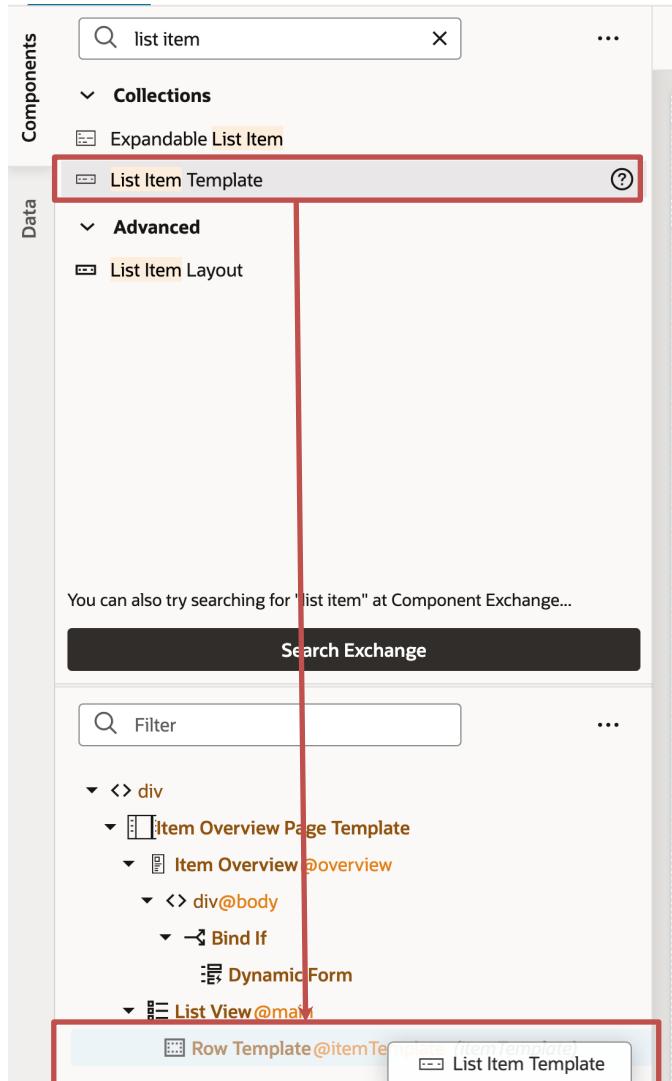
A screenshot of a web application interface. On the left, there is a sidebar with customer information: First Name (Norma), Last Name (Robinson), Phone (6505060009), Email (nrobinson@internalmail.com), Shipping City (Fusion City), and Shipping Postal Code (90747). To the right, a list view shows five orders. Each order entry includes a timestamp, an ID, a status, and a quantity. The orders are:

Order ID	Timestamp	Status	Quantity
F000000146	2022-08-13T15:43:33+00:00	DRAFT	6895
F000000147	2022-08-17T15:43:33+00:00	DRAFT	8006
F000000148	2022-08-20T15:43:33+00:00	DRAFT	1953
F000000299	2022-05-29T15:43:33+00:00	PROCESSING	1019

26. Try to format this section a little better. Delete the List Item Layout.

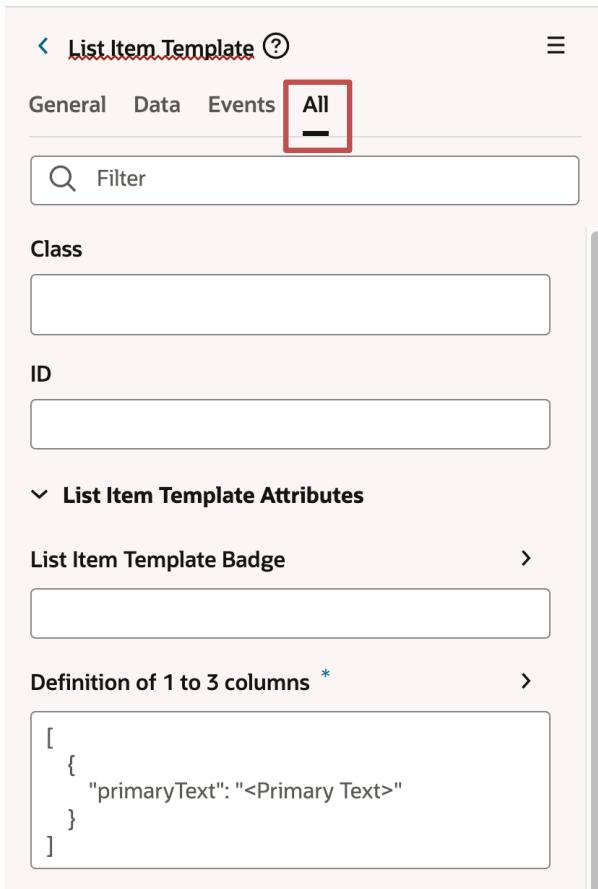
A screenshot of the Oracle Dynamic Forms editor. The left pane shows a tree structure of form components. Under "List View @main", the "Row Template @itemTe" node has its "List Item Layout" child node selected and highlighted with a red box. To the right, a toolbar includes a "Delete" button, which is also highlighted with a red box. Below the toolbar, there are three options: "div@leading (Bind Text)", "div@overline (Bind Text)", and "div (Bind Text)".

27. Drag a **List Item Template** onto the **Row Template**.



28. Click the **All** tab in the new **List Item Template's** Property pane.





29. Paste the following code into the **Definition of 1 to 3 columns** property. This code is in the code text file.

```
[[ [{"primaryText":$current.data.orderNumber,"secondaryText":$flow.functions.formatDate($current.data.orderDate),"overlineText":$current.data.orderStatus === 'DRAFT' ? 'New Order' : 'Order'}, {"overlineText":"Total Amount","primaryText":$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode, $current.data.totalAmount ? $current.data.totalAmount : 0)}]]
```

List Item Template

General Data Events All

Filter

Class

ID

List Item Template Attributes

List Item Template Badge

Definition of 1 to 3 columns *

```
[[{"primaryText":$current.data.orderNumber,"secondaryText":$flow.functions.formatDate($current.data.orderDate),"overlineText":$current.data.orderStatus === 'DRAFT' ? 'New Order' : 'Order'}, {"overlineText":"Total Amount","primaryText":$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode, $current.data.totalAmount ? $current.data.totalAmount : 0)}]]
```

The orders section is formatted, and column headers are added to display based on order status.

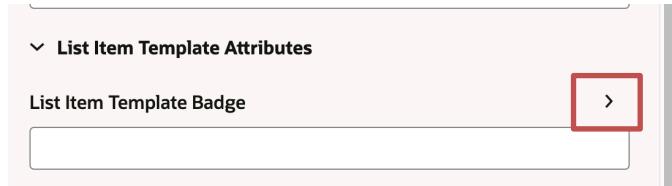
[↑ Parent page](#)

Rogers

FirstName Kathryn	New Order F000000313 8/7/2022	Total Amount \$5,151.00
LastName Rogers	New Order F000000314 8/11/2022	Total Amount \$1,664.00
EmailAddress krogers@internalmail.com	New Order F000000315 8/21/2022	Total Amount \$4,481.00
ShippingAddressCity Vision City	Order F000000300 6/2/2022	Total Amount \$5,311.00
ShippingPostalCode 95,727	Order F000000301 6/5/2022	Total Amount \$3,786.00
	Order F000000303 6/16/2022	Total Amount \$4,335.00
	Order F000000304 6/19/2022	Total Amount \$7,494.00

Note: You may wish to inspect the code to see how this was done. Note the use of the JavaScript ternary expression to set up the column header text and the use of JavaScript functions to format the date and order total.

30. Click the **Show sub-properties >** icon for the **List Item Template Badge**.



31. Enter `[[$current.data.orderStatus]]` for the **Badge Text** property and `[[$current.data.orderStatus === 'DRAFT' ? 'warning' : 'info']]` for the **Badge Status**. Select **Leading** for the **Badge Position**.

32. The page is complete. Test it in Live mode. Open the main-manage-orders page and click the ... button to View Customer. The page should navigate to the main-view-customers page and display the correct customer and order information. **Hint:** Remember to hide the Web Apps tab or change the screen size to **Desktop** to make sure the summary information appears.

Set up the Orders List View to navigate to the main-view-orders page when an order is selected.

33. On the **main-view-customers** page, make sure the Page Designer is the active tab.

34. Select the **List View** for the Orders. Add an **On 'First Selected Item'** event listener.

The screenshot shows two parts of the Oracle ADF interface. The top part is a tree view of the 'Item Overview Page Template' structure:

- <> div
- Item Overview Page Template
 - Item Overview@overview
 - <> div@body
 - Bind If
 - Dynamic Form
 - List View@main
 - Row Template@itemTemplate (itemTemplate)
 - List Item Template

35. In the action chain, add a **Navigate to Page** action to navigate to the **main-view-orders** page.

The screenshot shows the 'Action Chain' configuration screen:

- Page *: main-view-orders (highlighted with a red box)
- Create button
- Go to Page button

36. Bind the ***rowKey** to the **selectedOrderId**.

The screenshot shows the Oracle ADF Page Designer interface. At the top, there's a 'Parameters' panel with a single parameter: 'selectedOrderId *'. Below it is a 'Variables' panel. The 'Variables' panel has a 'Filter' input field. It contains several sections: 'Local' (with a 'Create' button), 'Input Parameters' (with a 'Create' button), 'rowData' (marked with a red asterisk), 'rowKey' (which is highlighted with a red rectangle), 'Page' (with a 'Variables' section containing 'bidirectionalNavigation', 'displayMode', 'inFlowBack', and 'itemId', each with an 'Edit' button), and a 'Page' section with a 'Create' button.

Below the variables panel is a code editor window with the following content:

```
toMainViewOrders = Navigate To Page main-view-orders
Parameters
A selectedOrderId: rowKey
```

37. Return to the Page Designer and test that this works in **Live** mode by selecting a row for an order. The page should navigate to the main-view-orders page and display the details of the selected order.
38. Close all open tabs.

This completes this practice.