# 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

Copyright © 2024, Oracle and/or its affiliates.

## 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.

### Assumptions

You have successfully completed Practice 2-1.

Copyright © 2024, Oracle and/or its affiliates.

## 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.

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

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

Copyright © 2024, Oracle and/or its affiliates.

4. 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.

**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.

5. 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.

Copyright © 2024, Oracle and/or its affiliates.

6. 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`.

Copyright © 2024, Oracle and/or its affiliates.

7. 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.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

12. Filter for **orderNumber** or **customerLastname** to match the **orderSearchString**.
    a. Select **orderNumber** for the first attribute.

    b. 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.

    a. Set the second attribute to **$variables.orderSearchString**.

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

Copyright © 2024, Oracle and/or its affiliates.

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**.

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

Copyright © 2024, Oracle and/or its affiliates.

16. Clear the Search field and enter `146`.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

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

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' ]]
```

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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.

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

37. For **Primary Text**, enter:
```
[[$flow.functions.formatCurrency($flow.constants.defaultCurrencyCode
, $current.data.totalAmount ? $current.data.totalAmount : 0)]]
```

Copyright © 2024, Oracle and/or its affiliates.

**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.

38. 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.

Copyright © 2024, Oracle and/or its affiliates.

39. Click the **General** tab.

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

Copyright © 2024, Oracle and/or its affiliates.

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

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

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.

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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**.

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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**.

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

56. Set the following properties:
    - **Action Id**: `order`
    - **Label**: `View Order`

Copyright © 2024, Oracle and/or its affiliates.

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

**Note**: More Redwood icons are available here:
https://static.oracle.com/cdn/fnd/gallery/2410.0.3/images/preview/index.html

Copyright © 2024, Oracle and/or its affiliates.

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].**

Copyright © 2024, Oracle and/or its affiliates.

62. Click **+** to add another **Secondary Action**.
63. For **[2]** click **>** to enter More Properties.

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

64. Click **< Secondary Actions [2]**.

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.

Copyright © 2024, Oracle and/or its affiliates.

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**.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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**.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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.

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

Copyright © 2024, Oracle and/or its affiliates.

80. Select **customerID** and click **Finish**.

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.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

85. Close all open tabs.

This completes this practice.

Copyright © 2024, Oracle and/or its affiliates.

## 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.

Copyright © 2024, Oracle and/or its affiliates.

## 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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

**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.

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

**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.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

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

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

Copyright © 2024, Oracle and/or its affiliates.

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

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)},{"lab
el":"Total
Amount","value":$flow.functions.formatCurrency($flow.constants.default
CurrencyCode, $page.variables.rraOrders.totalAmount ?
$page.variables.rraOrders.totalAmount : 0)}] ]]
```

Copyright © 2024, Oracle and/or its affiliates.

18. The textual information is displayed in the header.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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**.

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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}}`.

Copyright © 2024, Oracle and/or its affiliates.

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.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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**.

Copyright © 2024, Oracle and/or its affiliates.

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.

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.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

```
[[ $variables.rraOrders.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.

Copyright © 2024, Oracle and/or its affiliates.

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

**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.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

39. 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**.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

41. 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.

Live | Design | Code

```
1  <oj-sp-general-overview-page id="oj_gop1" display-options.responsive-padding="on"
2    page-title='[[ "Order "+ $variables.rraOrders.orderNumber ]] '
3    contextual-info='[[ [{"label":"Customer","value":$variables.rraOrders.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.rraOrders.customerIdObject.items[0
9                        readonly="true" label-hint="CustomerName"></oj-input-text>
10                     <oj-input-text value="[[ $variables.rraOrders.customerIdObject.items[0
11                       readonly="true" label-hint="EmailAddress"></oj-input-text>
12                     <oj-sp-input-phone-number label-hint="Input Phone Number"
13                       value.subscriber-number="[[ $variables.rraOrders.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.rraOrders.shipp
19                     <oj-vb-fragment-param name="country" value="[[ $variables.rraOrders.sh
20                     <oj-vb-fragment-param name="state" value="[[ $variables.rraOrders.ship
21                     <oj-vb-fragment-param name="street_1" value="[[ $variables.rraOrders.s
22                     <oj-vb-fragment-param name="street_2" value="[[ $variables.rraOrders.s
23                     <oj-vb-fragment-param name="postalCode"
24                       value="[[ $variables.rraOrders.shippingAddressPostalCode ]]"></oj-vb
25               </oj-vb-fragment>
26         </div>
27   </div>
28   <oj-table scroll-policy="loadMoreOnScroll" slot="main" data="[[$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>
```

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

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

The images now display

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

Copyright © 2024, Oracle and/or its affiliates.

**Add a drawer component to display product data.**

44. 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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

47. 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.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

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.

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

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

Copyright © 2024, Oracle and/or its affiliates.

**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.

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

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

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

Copyright © 2024, Oracle and/or its affiliates.

56. 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        <img
54          :src="[[ $application.path + 'resources/' + $variables.rraOrders.orderLinesCollection.items[0]?.im
55        <p class="oj-p">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        <img
54          :src="[[ $application.path + 'resources/' + $variables.rraOrders.orderLinesCollection.items[0]?.im
55        <p class="oj-p">
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>
```

Copyright © 2024, Oracle and/or its affiliates.

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

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

59. Change the **Value** for the **Bind Text** to:
```
[[ $variables.rraOrders.orderLinesCollection.items[0]?.productName
]]
```

Copyright © 2024, Oracle and/or its affiliates.

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

61. 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**.

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

Copyright © 2024, Oracle and/or its affiliates.

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

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

Copyright © 2024, Oracle and/or its affiliates.

65. 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.

66. 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 **!**

Copyright © 2024, Oracle and/or its affiliates.

67. 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.

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

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

Copyright © 2024, Oracle and/or its affiliates.

70. Return to **Page Designer** to test the drawer.
71. 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.
72. Return to Design mode.
73. 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**.

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

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

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

77. The **Variables** tab opens. Rename **get_rraOrders_orderLinesCollectionVar** to `rraLineItem`.

78. 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.

Copyright © 2024, Oracle and/or its affiliates.

Copyright © 2024, Oracle and/or its affiliates.

79. 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.

Copyright © 2024, Oracle and/or its affiliates.

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

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

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

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

```
[[ $variables.rraLineItem.productName ]]
```

Copyright © 2024, Oracle and/or its affiliates.

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

Copyright © 2024, Oracle and/or its affiliates.

84. 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.

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

Copyright © 2024, Oracle and/or its affiliates.

86. 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.

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

Copyright © 2024, Oracle and/or its affiliates.

88. Bind the **selectedOrderId** to the ***key**.

89. Return to the **Page Designer** tab.
90. 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.

Copyright © 2024, Oracle and/or its affiliates.

91. Return to Design mode.
92. Close all open tabs.

This completes this practice.

Copyright © 2024, Oracle and/or its affiliates.