

Practices for Lesson 6:
Create a Dashboard Page

Overview

In these practices, you will:

- Add a dashboard page to display KPIs and a chart using the Dashboard Landing Page Template

Practice 6-1: Create a Dashboard Landing Page

Overview

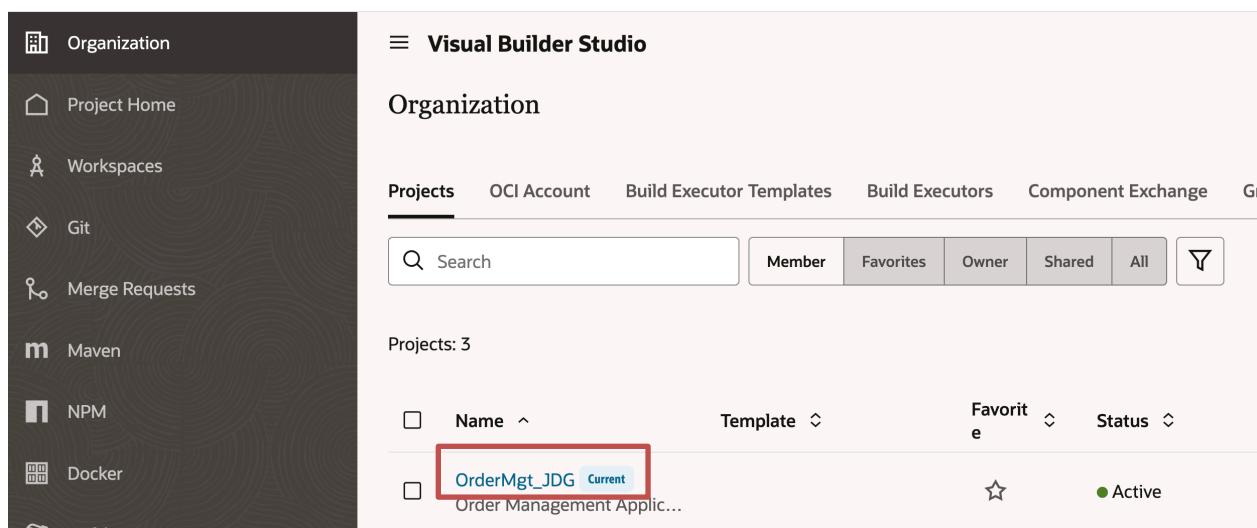
In this practice, you create a dashboard page to display summary information using the Dashboard Landing Page Template. You will display summary values as well as corresponding bar and pie charts.

Assumptions

You have successfully completed Practice 2–1.

Tasks

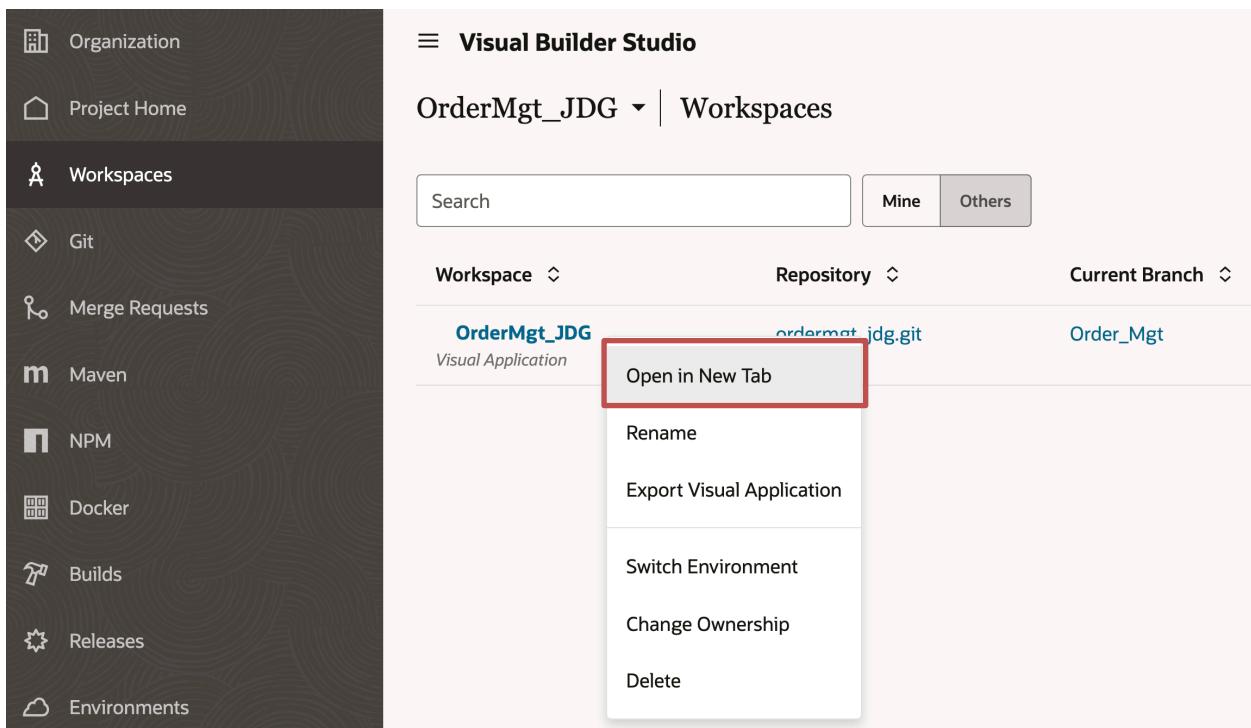
1. Log in to **Visual Builder Studio** and select your project name on the **Projects** tab. Your project name will be **OrderMgt_<your initials>**, where **<your initials>** corresponds to your initials.



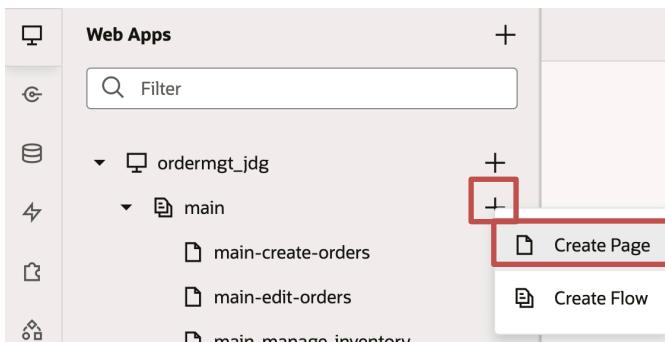
The screenshot shows the Visual Builder Studio interface. On the left is a dark sidebar with icons for Organization, Project Home, Workspaces, Git, Merge Requests, Maven, NPM, Docker, and Builds. The main area is titled "Visual Builder Studio" and "Organization". At the top, there are tabs for Projects (which is selected), OCI Account, Build Executor Templates, Build Executors, Component Exchange, and a partially visible tab. Below the tabs is a search bar with placeholder "Search" and filters for Member, Favorites, Owner, Shared, and All. A "Projects: 3" section follows, with a table showing three projects. The first project, "OrderMgt_JDG" (with a status of "Current" and a tooltip "Order Management Applic..."), is highlighted with a red box. The table has columns for Name, Template, Favorite, and Status. The "Name" column is sorted by Name. The "Template" column is sorted by Template. The "Favorite" column is sorted by Favorite. The "Status" column is sorted by Status. The "OrderMgt_JDG" row also includes a star icon and an "Active" status indicator.

Name	Template	Favorite	Status
OrderMgt_JDG <small>Current</small>	Order Management Applic...	☆	● Active

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



- Close the **Getting Started** tab if it's open.
- Select the **Web Apps** tab in the Navigator. Expand **ordermgt-<your initials>**. Expand **main**. Click the **Create Page +** for the **main** flow to create a new page named **main-dashboard**. Use the **Dashboard Landing Page Template**.



Create Page

Page ID *

main-dashboard

Choose Page Content

Filter



Create an Empty Page

The empty page can be customized from page designer.



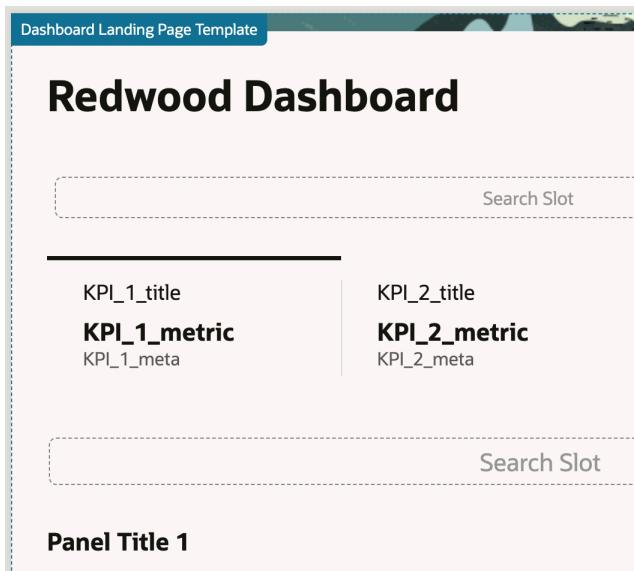
Dashboard Landing Page Template

Summarize business performance and status, including KPIs and visualizations drawn from different sources.

The screenshot shows the Oracle ADF Page Designer interface. The top navigation bar includes tabs for Page Designer, Action Chains (3), Event Listeners (1), Events, Types (1), Variables (5), JavaScript, JSON, and Settings. Below the navigation is a toolbar with search, fit to canvas, auto-fit, zoom, and live/design/code buttons. The main workspace contains a placeholder for the page title and two KPI metrics. The left sidebar lists components under 'Components' (Redwood Patterns, Bottom Drawer Template) and 'Data' (various dashboard and card components). The bottom-left panel shows the page's structure, including a 'Bind If' condition for the 'Scoreboard @kpi' component.

- Select the **Dashboard Landing Page Template** and set its **Page Title** to Redwood Dashboard.

The screenshot shows the Oracle ADF Page Properties dialog for the 'Dashboard Landing Page Template'. The 'General' tab is selected. The 'Page Title' field is highlighted with a red box and contains the value 'Redwood Dashboard'. Other fields visible include 'ID' (oj-sp-dashboard-page-1) and a 'Properties' sidebar.



6. Next, you need to set the text and titles e.g., **KP_1_title**, **KPI_1_metric**, and **KPI_1_meta**, in the scoreboard. This is stored in an array. Click the **Variables** tab to enter the data for the KPIs array. Select the predefined **kpisArray**.

Variables

ADP kpisADP

kpisArray

{ } item[i]

A layout

▶ { } primaryAction actionType

A selectedKpi

7. Delete the current **Default Value** and paste the following code in its place.

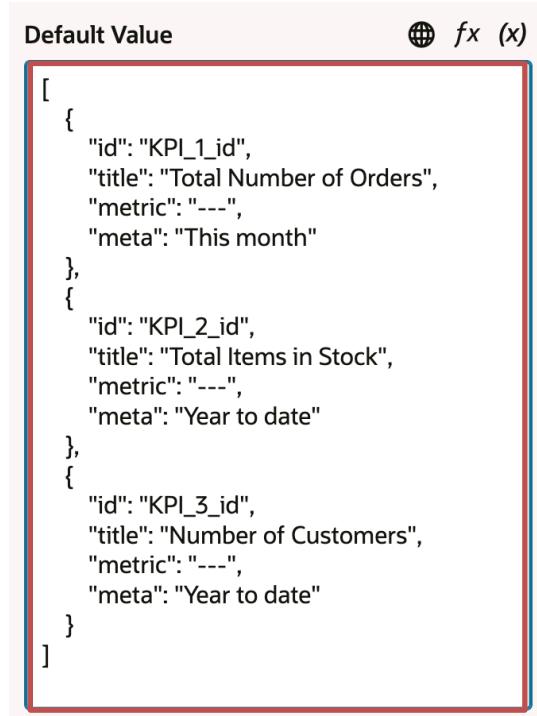
Before:

Default Value

```
[  
  {  
    "id": "KPI_1_id",  
    "title": "KPI_1_title",  
    "metric": "KPI_1_metric",  
    "meta": "KPI_1_meta"  
  },  
  {  
    "id": "KPI_2_id",  
    "title": "KPI_2_title",  
    "metric": "KPI_2_metric",  
    "meta": "KPI_2_meta"  
  }  
]
```

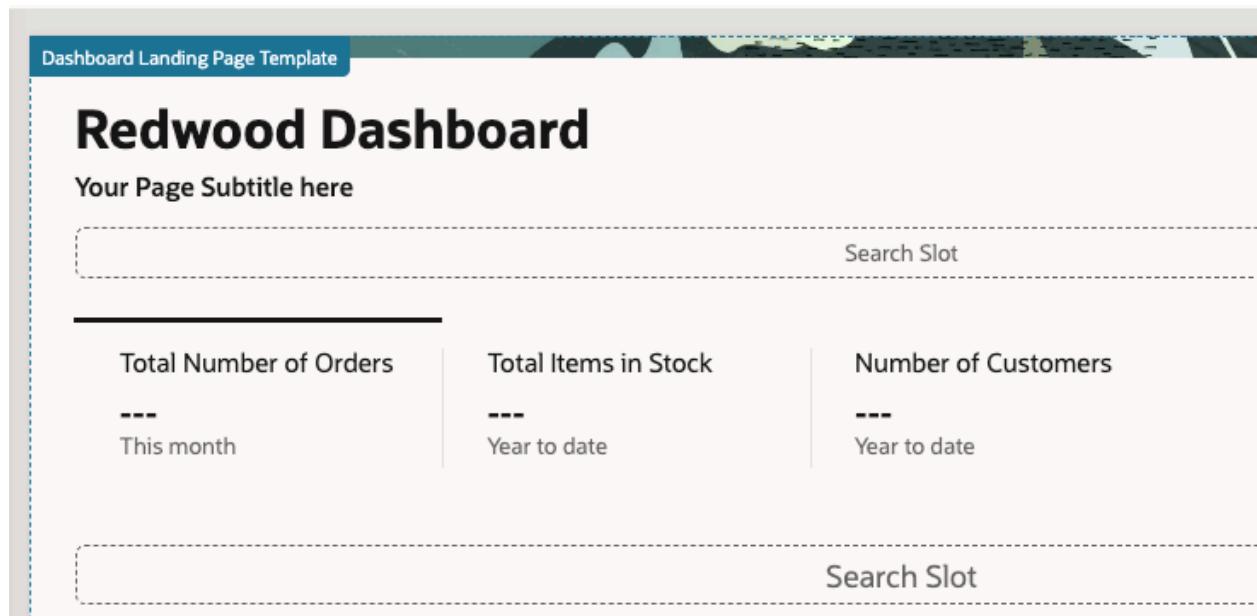
```
[  
  {  
    "id": "KPI_1_id",  
    "title": "Total Number of Orders",  
    "metric": "---",  
    "meta": "This month"  
  },  
  {  
    "id": "KPI_2_id",  
    "title": "Total Items in Stock",  
    "metric": "---",  
    "meta": "Year to date"  
  },  
  {  
    "id": "KPI_3_id",  
    "title": "Number of Customers",  
    "metric": "---",  
    "meta": "Year to date"  
  }  
]
```

After:



```
Default Value   fx (x)  
[  
  {  
    "id": "KPI_1_id",  
    "title": "Total Number of Orders",  
    "metric": "---",  
    "meta": "This month"  
  },  
  {  
    "id": "KPI_2_id",  
    "title": "Total Items in Stock",  
    "metric": "---",  
    "meta": "Year to date"  
  },  
  {  
    "id": "KPI_3_id",  
    "title": "Number of Customers",  
    "metric": "---",  
    "meta": "Year to date"  
  }]
```

8. Click the **Page Designer** tab to return to the page. Notice the scoreboard has been updated to reflect your changes. Now, you will set the values for each of the KPIs. KPI values are loaded using an action chain.

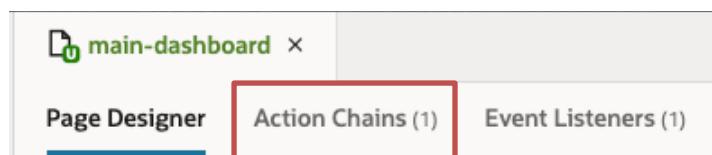


The screenshot shows the 'Dashboard Landing Page Template' interface. At the top, there's a header bar with the title 'Redwood Dashboard'. Below the header, there's a subtitle placeholder 'Your Page Subtitle here'. Underneath, there are three data cards representing KPIs:

- Total Number of Orders:** Displays '---' and 'This month'.
- Total Items in Stock:** Displays '---' and 'Year to date'.
- Number of Customers:** Displays '---' and 'Year to date'.

Each card has a dashed rectangular outline below it, labeled 'Search Slot'.

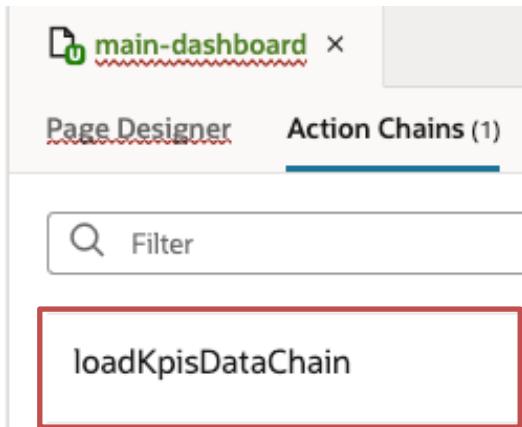
9. Click the **Action Chains** tab to build the action chains that will load the data.



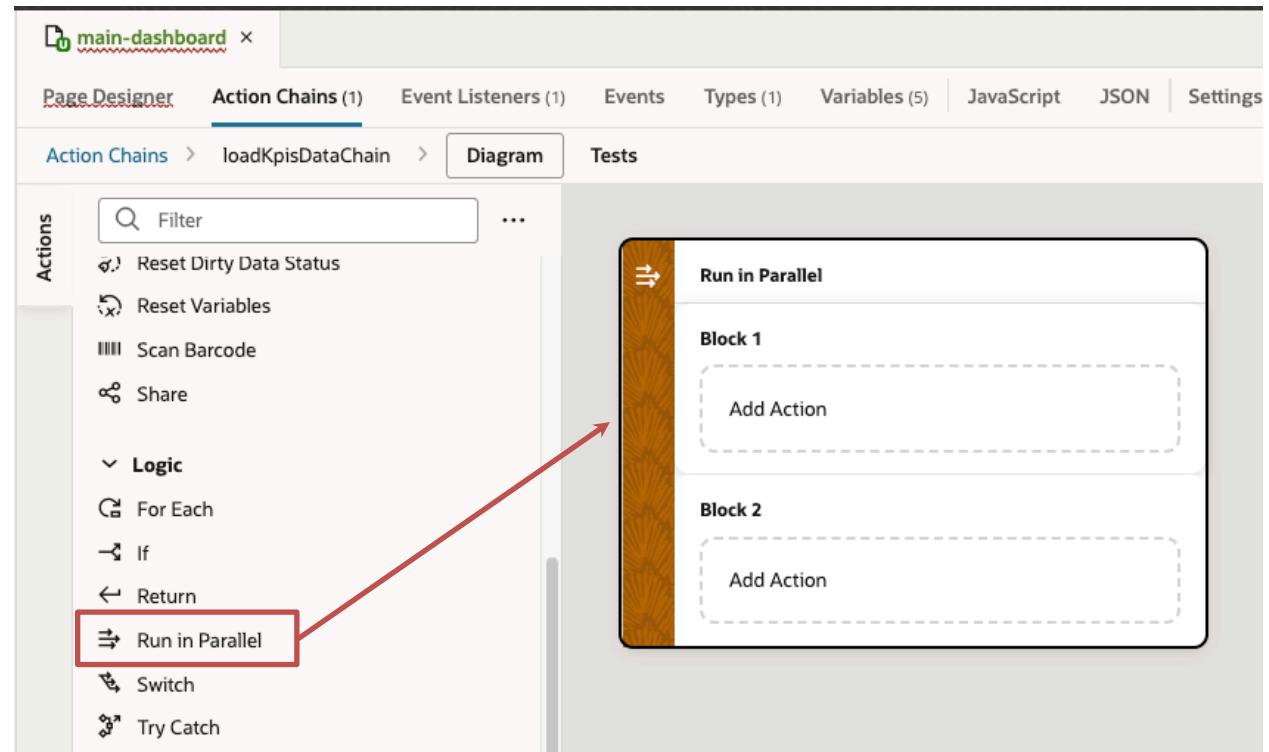
The screenshot shows the 'Action Chains' tab selected in the navigation bar of the 'main-dashboard' page. The tabs are labeled 'Page Designer', 'Action Chains (1)', and 'Event Listeners (1)'. The 'Action Chains (1)' tab is highlighted with a red border.

Notice there is one action chain created for you: **loadKpisDataChain**. This action chain calls the individual action chains to load the data values for each part of the scoreboard component.

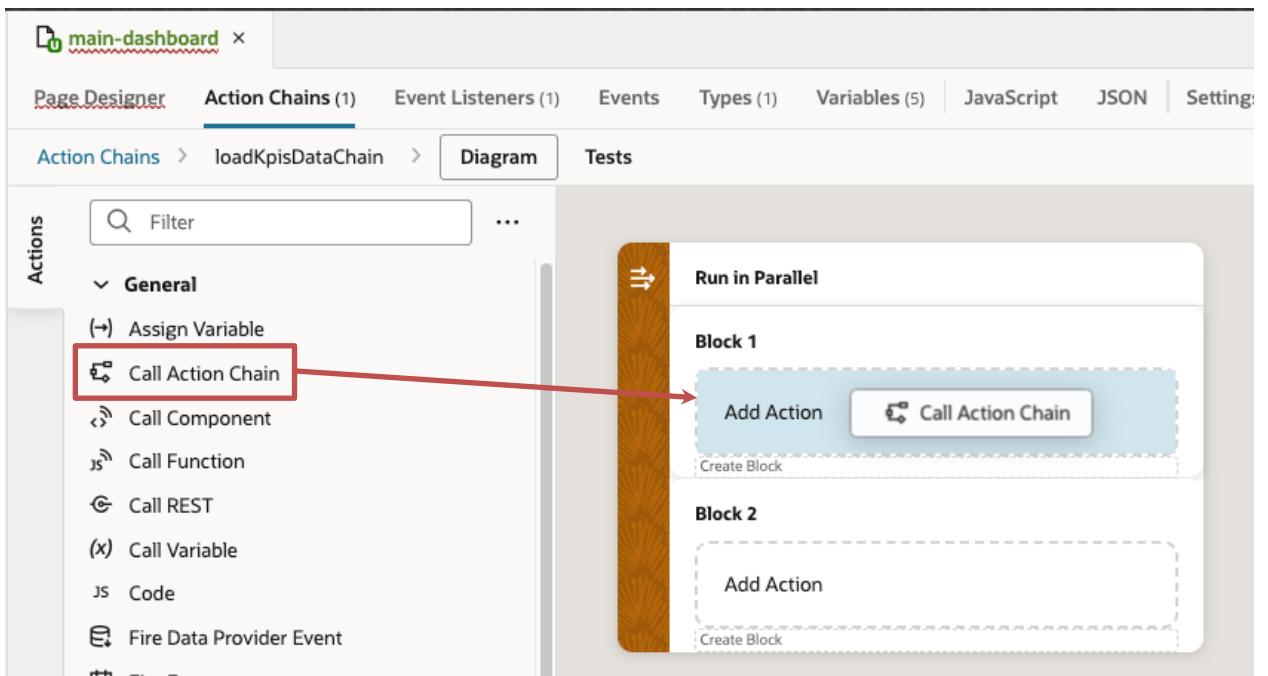
10. Click the **loadKpisDataChain** to open it in the Action Chain editor.



11. Drag a **Run in Parallel** action to the action chain canvas.



12. Drag a **Call Action Chain** action to the Block 1 Add Action area.



13. With the Call Action Chain action selected, click its **Create** link in the Properties panel to create the new action chain.

Call Action Chain

ID	...
Description	
Action Chain *	<input type="button" value="Create"/> <small>Must have required property 'chain'</small>

14. Set its ID to `load_KPI_1_data` and click **Create**.

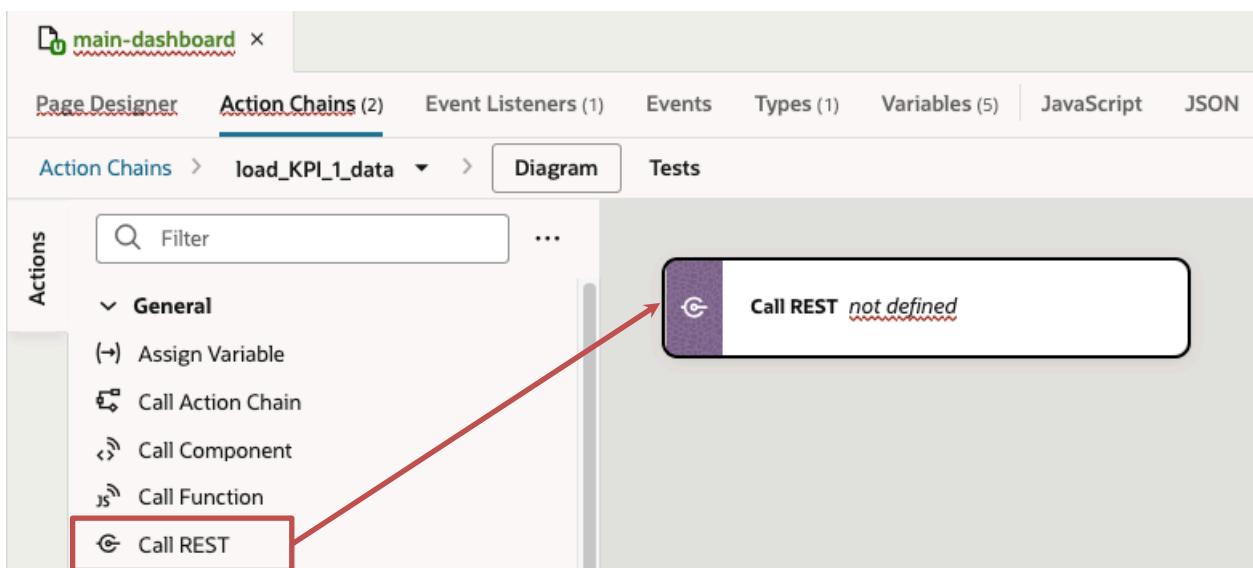
Create Action Chain

<input checked="" type="radio"/> JavaScript	<input type="radio"/> JSON
Scope *	Page
ID *	<input type="text" value="load_KPI_1_data"/>

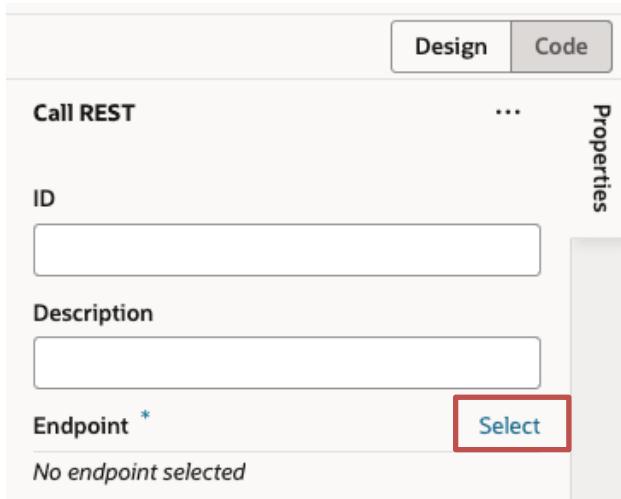
15. In the interest of time and not doing too much redundant work, you will configure just this action chain. Click the **Go to Action Chain** link for the new action chain. The action chain opens in its editor.



16. Drag a **Call REST** action to the canvas.



17. With the Call REST action selected, click the **Select** link for its endpoint in the Properties pane.



18. Expand **Business Objects > rraOrders** and select the first **GET rraOrders** and click **Select**.

The screenshot shows a 'Select Endpoint' interface. At the top is a search bar labeled 'Filter'. Below it is a tree view of business objects under 'Business Objects': 'rraCustomers', 'rraOrderLines', and 'rraOrders'. Under 'rraOrders', there are two entries: 'GET /rraOrders' and 'GET /rraOrders'. The first 'GET /rraOrders' entry is highlighted with a red border. Below these entries is a table with three rows: 'Service' (Business Object), 'ID, Action Hint' (getall_rraOrders), and 'Description' (Get all). To the right of the table, there are three curly brace symbols: '}', '}', and '}'.

19. By default, only 25 records are returned. Override this by setting the **limit** to 1000. This will ensure all the records are returned and you receive a more accurate count. **Note:** The **limit** controls how many records are retrieved in a REST call.

20. Click the **limit** Input Parameter.

Endpoint * Select

GET getall_rraOrders

Header Parameters	Assign
A Metadata-Context	Not Mapped
A REST-Framework-Version	Not Mapped

Input Parameters	Assign
A expand	Not Mapped
A fields	Not Mapped
A finder	Not Mapped
# limit	Not Mapped
A links	Not Mapped
# offset	Not Mapped
☒ onlyData	Not Mapped
A orderBy	Not Mapped
A q	Not Mapped
☒ totalResults	Not Mapped

21. Select **limit** in the list of UriParams.

Target

▼ Parameters

▼ { } uriParams

- A expand
- A fields
- A finder
- # limit**
- A links

22. Enter 1000 and click **Save**.

```
# uriParams.limit
```

1 1000

Assign Input Parameters

Sources

- Local
- Input Parameters +
- Page (main-dashboard) +
 - Variables
 - gridLayout
 - { } kpisADP
 - [] kpisArray
 - { } primaryAction
 - selectedKpi
 - Functions +
 - System
- Flow (main) +
 - Functions +
 - System

uriParams.limit

1 1000

Target

- Parameters
 - { } → { } uriParams
 - A expand
 - A fields
 - Δ finder
 - # limit
 - A links
 - # offset
 - onlyData
 - A orderBy
 - A q
 - totalResults

```
response = Call REST businessObjects/getall_rraOrders
URI Parameters
# limit: 1000
```

23. Add an **Assign Variable** action after the Call REST action.

Actions

General

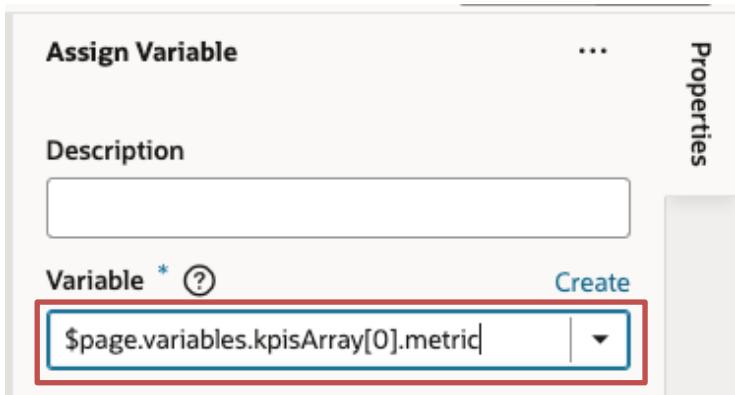
- (-) Assign Variable
- Call Action Chain
- Call Component
- Call Function
- Call REST

response = Call REST businessObjects/getall_rraOrders
URI Parameters
limit: 1000

(-) Assign Variable

24. With the **Assign Variable** action selected, enter

\$page.variables.kpisArray[0].metric for its **Variable**.

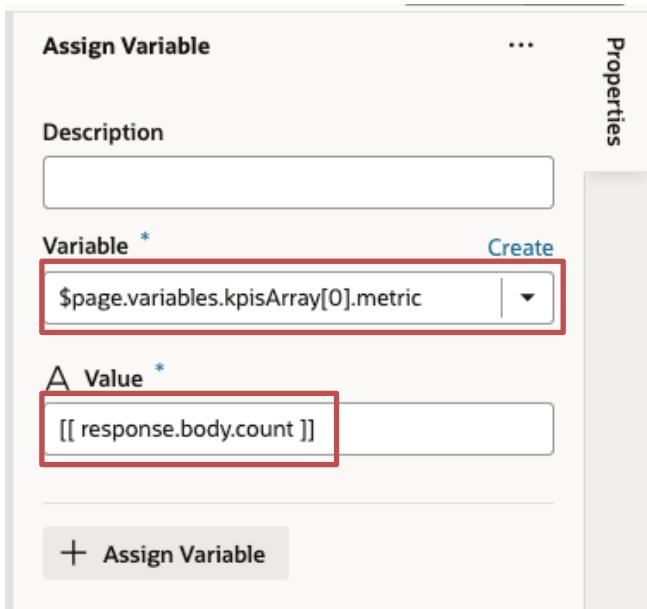


25. For the Value, click the (x) to select a variable. Expand Local > response > body and select count.

This screenshot shows the 'Value' selection dialog. At the top left is the title 'Value'. To the right are icons for 'Variable' (with a red box), 'Create', and '(x)'. Below the title is another 'Variable' input field containing '\$page.variables.kpisArray[0].metric'. A red box highlights this field. At the bottom left is another 'Value' input field. The main area is titled 'Variables' and contains a search bar labeled 'Filter'. Below the search bar is a tree view:

- Local
 - { } response
 - { } body
 - # count
 - hasMore

A red box highlights the '# count' option under the 'body' node.



26. Confirm that this worked. Click the **Page Designer** tab. After a moment, you should see **500** displayed for the **Total Number of Orders**.

Note: You are setting the value for the **metric** key in the object in the **kpisArray**. You can provide this value in other ways depending on your needs and data. We're using a simple count of the records returned for this example so you can focus more on how to configure the **Dashboard Landing Page Template** rather than performing business-related calculations, but the idea is the same.

The screenshot shows the Redwood Dashboard page. At the top is the title 'Redwood Dashboard' and a subtitle 'Your Page Subtitle here'. Below this is a 'Scoreboard' section. Inside the scoreboard, there are two cards: one for 'Total Number of Orders' (value 500, subtext 'This month') and one for 'Total Items in Stock' (value ---, subtext 'Year to date').

You could repeat similar steps for the count of Products and Customers, but in the interest of time you don't have to do that. If you choose to do it, make sure to set the correct array element in the **kpisArray**:

Products would be: `$page.variables.kpisArray[1].metric`

Customers would be: `$page.variables.kpisArray[2].metric`

Add Charts to the Dashboard Grid

You will add three charts in the following steps: two charts to KPI_1 and one chart to KPI_2. A **switcher** component will display the correct chart(s) based on which KPI is selected.

27. Drag a **Bar Chart** to the Dashboard Panel under the Dashboard Grid for KPI_1_id.

main-dashboard x

Page Designer Action Chains (3) Event Listeners (1) Events

Components

Bar Container

Tab Bar

Tabs with Content

Toolbar

Charts

Bar Chart

Gauges

Progress Bar

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

Search Exchange

Filter

<> div

Dashboard Landing Page Template

Bind If

|Scoreboard@kpi

Switcher

Defer@KPI_1_id

Dashboard Grid

Dashboard Panel

Bar Chart

Defer@KPI_2_id

Dashboard Grid

Dashboard Panel

28. On the Bar Chart's **Quick Start** tab, click **Add Data**.

The screenshot shows a navigation bar with tabs: General, Data, Events, All, and Quick Start. The Quick Start tab is highlighted with a red box. Below it, there is a section titled 'Add Data' with a database icon and the text: 'Map the Chart to a data source, so we can populate it with data.'

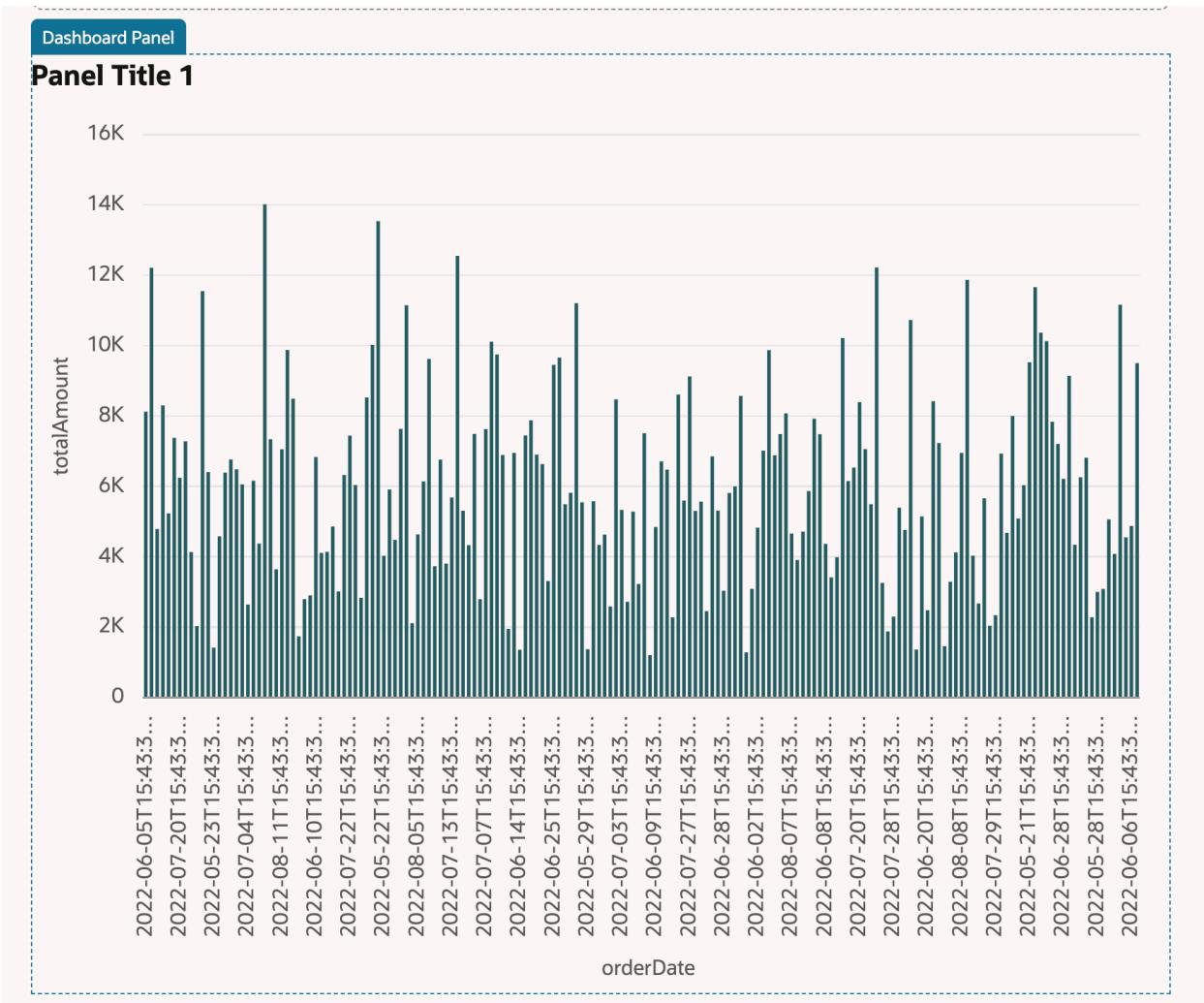
29. Click **rraOrders** and click **Next**.

The screenshot shows a list of Business Objects: rraCustomers, rraOrderLines, rraOrders, rraProducts, and rraSuppliers. The rraOrders item is highlighted with a red box.

30. Select **totalAmount** for the Values and **orderDate** for the **Categories** and click **Next** and **Finish**.

The screenshot shows the 'Bar Chart Component Fields' configuration. Under 'Values (Y Axis)', the expression '# totalAmount X' is selected. Under 'Categories (X Axis)', the expression 'A orderDate X' is selected.

The chart displays. Now, you can modify some of its properties to improve how it looks.



31. Click **Dashboard Panel** to enter a Panel Title.

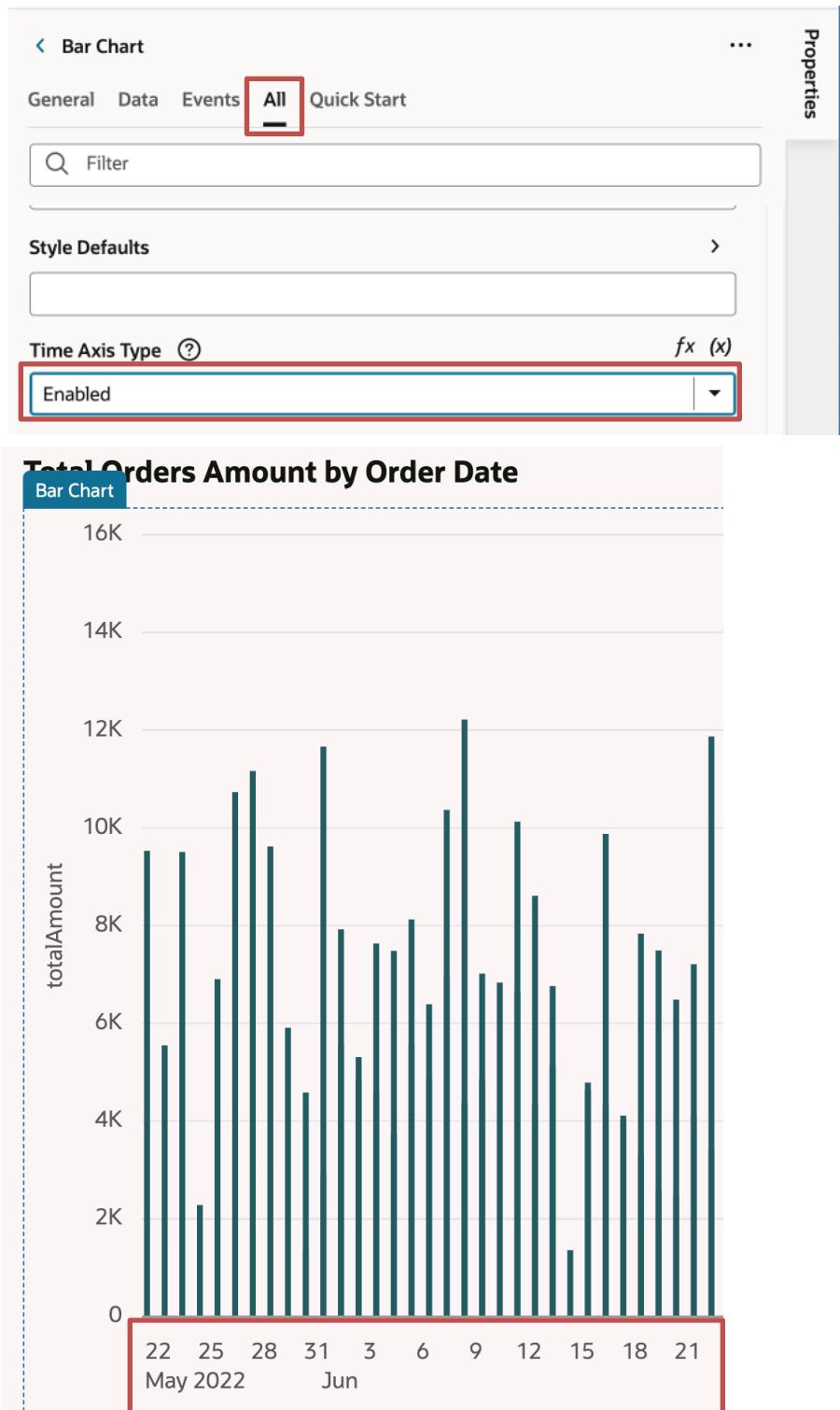
- ▼ <> div
- ▼ 📈 |Dashboard Landing Page Template
- ▼ → Bind If
- ▣|Scoreboard@kpi
- ▼ □ Switcher
- ▼ ⏲|Defer@KPI_1_id
- ▼ □|Dashboard Grid
- ▼ 📈|Dashboard Panel
- ?(?)

32. Enter Total Orders Amount by Order Date for its **Panel Title**.

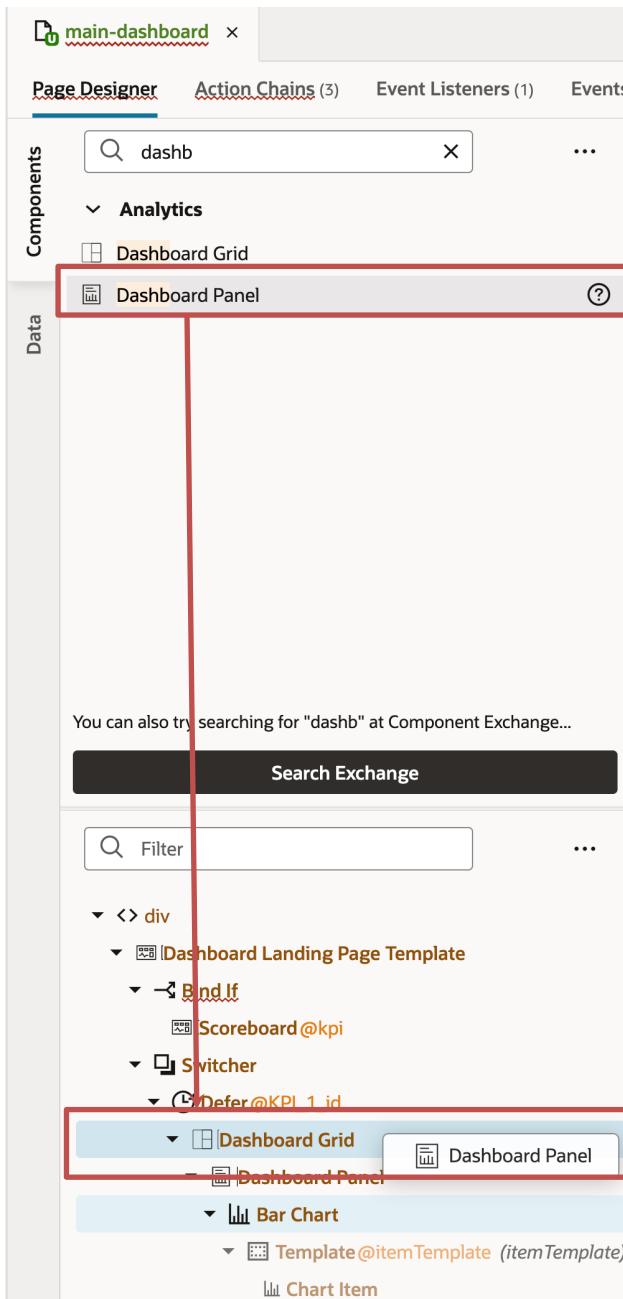
The screenshot shows the 'Dashboard Panel' configuration screen. At the top, there are tabs for 'General', 'Events', and 'All'. The 'General' tab is selected. On the right, there's a 'Properties' sidebar with a '...' button. Below the tabs, there's an 'ID' input field and a 'Panel Title' input field containing the text 'Total Orders Amount by Order Date'. A warning message '⚠ String not externalized for translation.' is displayed below the title input. The main area shows a bar chart with the title 'Total Orders Amount by Order Date'. The Y-axis is labeled 'totalAmount' and ranges from 0 to 16K. The X-axis is labeled 'orderDate' and shows dates from 2022-06-05T15:43:3... to 2022-09-06 00:15:43:3.... The chart displays daily fluctuations in total orders.

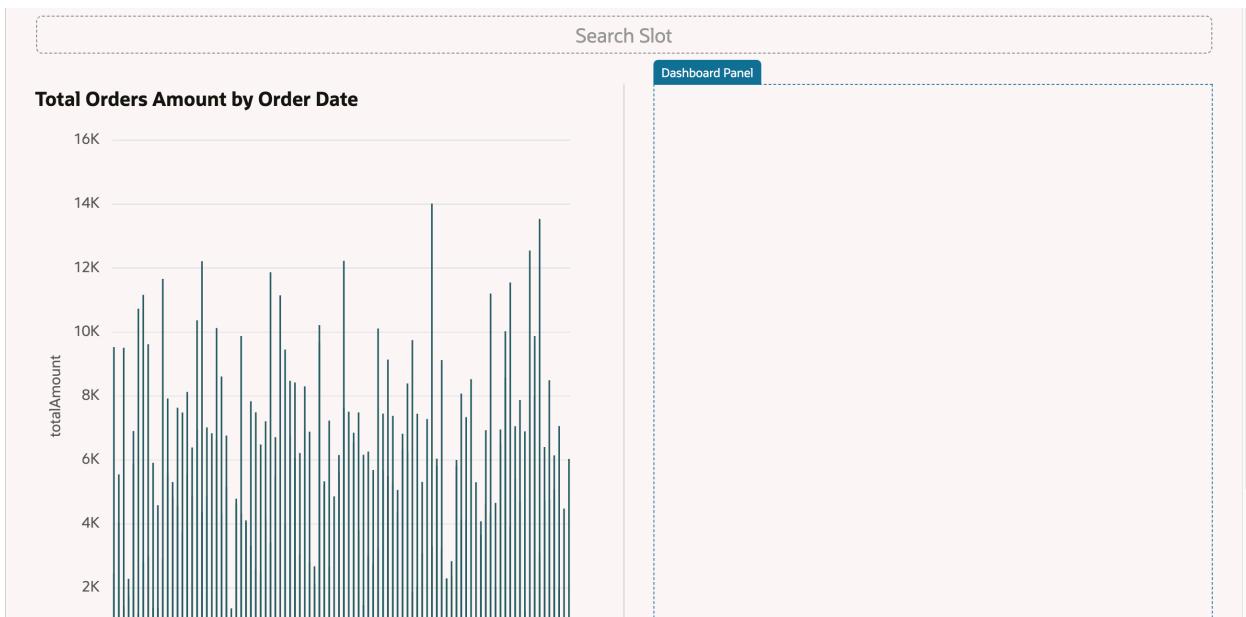
33. Click the **Bar Chart** and click its **All** tab. Set its **Time Axis Type** to **Enabled**. This will make the date more readable.

The screenshot shows the configuration of a 'Bar Chart' item. It is nested under 'Dashboard Grid' and 'Dashboard Panel'. The 'Bar Chart' item has a red border around it. Below it is a 'Template @itemTemplate (itemTemplate)' section, which also has a red border. The 'Chart Item' section is shown at the bottom.



34. Add another Dashboard Panel to the first Dashboard Grid. Select **Default** for the slot when prompted.





Tip: If you don't see the new panel on the right, hide the Navigator by clicking the Web Apps icon.

The figure shows the Oracle Analytics Cloud Page Designer interface. At the top, there is a title bar with a "main-dashboard" page name and a "Page Designer" tab. Below the title bar is a search bar with the text "dash". The left side features a sidebar with icons for "Components" (highlighted with a red box), "Data", and "Page Designer" (which is currently selected). The main area displays a hierarchical component tree. A "Bind If" condition is applied to a "Scoreboard@kpi" component. A "Switcher" component contains a "Defer@KPI_1_id" condition, which points to a "Dashboard Grid" component. This grid contains a "Dashboard Panel" component, which in turn contains a "Bar Chart" component. The "Template@itemTemplate (itemTemplate)" part of the tree is collapsed. At the bottom of the tree, another "Dashboard Panel" component is highlighted with a red box.

35. Drop a Pie Chart onto the new Dashboard Panel.

The screenshot shows the Oracle ADF Page Designer interface. The top navigation bar includes tabs for "Page Designer", "Action Chains (3)", "Event Listeners (1)", and "Events". Below the navigation is a search bar with the query "pie". The main area is divided into two panels: "Components" on the left and "Structure" on the right.

Components Panel: A search result for "pie" is displayed under the "Charts" category. The "Pie Chart" component is highlighted with a red border.

Structure Panel: The "Structure" panel shows the hierarchical structure of the dashboard page. A red arrow points from the "Pie Chart" component in the Components panel down to its corresponding node in the Structure panel. The structure is as follows:

- <> div
 - Dashboard Landing Page Template
 - Bind If
 - |Scoreboard @kpi
 - Switcher
 - Defer @KPI_1_id
 - Dashboard Grid
 - Dashboard Panel
 - Bar Chart
 - Template @itemTemplate (itemTemplate)
 - Chart Item
 - Defer @KPI_2_id
 - Dashboard Grid
 - Dashboard Panel

36. Click the **Add Data** option under **Quick Start** for the Pie Chart.

The screenshot shows a navigation bar with tabs: General, Data, Events, All, and Quick Start (which is highlighted with a red box). Below the tabs is a section titled 'Add Data' with the sub-instruction 'Map the Chart to a data source, so we can populate it with data.' A red box highlights this section. To the right is a 'Properties' panel.

37. Select **rraOrders**, and click **Next**.

This screenshot shows the 'Choose the source of your data' step. Under the 'Business Objects' category, there are five options: rraCustomers, rraOrderLines, rraOrders, rraProducts, and rraSuppliers. The 'rraOrders' option is selected and highlighted with a red box.

38. Select **totalAmount** for **Slice Values** and **orderStatus** for **Slice Colors**. Slice Values are the summary amounts and are automatically calculated by the component and the Slice Colors are the direct categories. In this case, Order Status values are DRAFT or PROCESSING. Click **Next** and **Finish**.

This screenshot shows the 'Pie Chart Component Fields' configuration. It includes two sections: 'Slice Values *' containing '# totalAmount X' and 'Slice Colors *' containing 'A orderStatus X'. Both sections are highlighted with red boxes.

39. The pie chart displays. You can modify some of the properties to improve how it looks. Click **Pie Chart** and, on the **General** tab, change its **Style** to **3D On**.

The screenshot shows the Oracle Analytics Cloud interface. At the top, there is a navigation bar with icons for Home, Recent, Dashboards, Reports, Predictions, and Help. Below the navigation bar is a search bar with placeholder text "Search dashboards, reports, predictions, and more...". The main area is a hierarchical tree view:

- Defer@KPI_1_id
 - Dashboard Grid
 - Dashboard Panel
 - Bar Chart
 - Template@itemTemplate (itemTemplate)
 - Chart Item
 - Dashboard Panel
 - Pie Chart
 - Template@itemTemplate (itemTemplate)

A red box highlights the "Pie Chart" item in the tree view. Below the tree view is a "Style" section with a "Style" button and a "Properties" panel on the right.

In the "Properties" panel, the "Style" tab is selected. It shows the "3D" button set to "On" (which is highlighted with a red box). There is also a "Pie Inner Radius" input field set to "0".

At the bottom, there is a preview area titled "Pie Chart" showing a 3D pie chart with two segments: one orange segment labeled "33.7%" and one teal segment labeled "66.3%".

40. Lastly, you'll add a **Bar Chart** to the **KPI_2_id** area. Drag a **Bar Chart** onto the **Dashboard Panel** for **KPI_2_id**.

Page Designer Action Chains (4) Event Listeners (1) Events (0)

Components

Search bar: bar

...
...

Bar Chart

Gauges

Progress Bar

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

Search Exchange

Filter

Scoreboard@kpi

Switcher

Defer@KPI_1_id

Dashboard Grid

Dashboard Panel

Bar Chart

Template@itemTemplate (itemTemplate)

Chart Item

Dashboard Panel

Pie Chart

Template@itemTemplate (itemTemplate)

Chart Item

Defer@KPI_2_id

Dashboard Grid

Dashboard Panel

Bar Chart

Structure

- It worked, but you don't see anything. Why? Because the switcher component is displaying the Dashboard for KPI_1_id by default. Switch to **Live** mode and click the **Total Items in Stock** KPI at the top the page. The screen changes and now KPI_2 displays.

The screenshot shows a dashboard with three KPI cards:

- Total Number of Orders:** 500 (This month)
- Total Items in Stock:** 3 (Year to date) - This card is highlighted with a red border.
- Number of Customers:** 3 (Year to date)

- The bar chart you just added is now visible. Return to **Design** mode to complete configuring the new pie chart.
- Select the **Bar Chart**, click its **Quick Start** tab, and click **Add Data**.

The screenshot shows the 'Add Data' configuration interface for a Bar Chart:

- Defer@KPI_2_id
- Dashboard Grid
- Dashboard Panel
- Bar Chart** (highlighted with a red border)

- Select **rraProducts** and click **Next**.

The screenshot shows the 'Choose the source of your data' step in the Bar Chart configuration:

Choose the source of your data

Business Objects

rraCustomers rraOrderLines rraOrders **rraProducts** (highlighted with a red border) rraSuppliers

- Select **numOfItems** for the **Values (Y Axis)** and **categoryName** for **Categories (X Axis)** and click **Next** and **Finish**.

The screenshot shows the 'Bar Chart Component Fields' configuration step:

Bar Chart Component Fields

Values (Y Axis) *
numOfItems X

Categories (X Axis) *
A categoryName X

46. Test whether this works by entering **Live** mode and click each KPI at the top in turn. The display at the bottom changes to show the correct charts. What happens if you click Number of Customers? Nothing is displayed because you didn't add the needed **Defer**, **KPI_3_id**, and related components.

The Defer component delays applying bindings to its children until it is activated. It “defers” adding the children to the DOM until the parent activates the subtree. This can be used with content as in the case of the switcher here. It can also be used with large pages when you don’t want to bind and process children when they’re not displayed on the page, and you want to wait to activate the child subtree until that portion of the page is visible. The Structure pane supports this by rendering the font and icons of the deferred components in lighter text if they’re not currently visible on the screen.

47. Close all tabs.

This completes this practice.

