Appian
Academy

Appian Step-by-Step #10

Exercise to Accompany
Reports: Build Basic Charts and Grids

The Appian Step-by-Step series consists of 12 exercises that accompany the courses in the Appian Developer learning path. Exercises build upon each other. Complete exercises in order and keep the app and all objects until you are done with the project.

Exercise 10: Reports

In this exercise, you will create a report for Supervisors and Registrars. This report will contain:

- A column chart of vehicles by make and category
- A pie chart of vehicles by make
- A drillable bar chart by make

You will create each chart as a separate interface. After, you will combine them into a single master interface. This master interface will be used to create a report object that you will later add to your application site.

Create a Column Chart

Let's create the column chart of vehicles by make and category first.

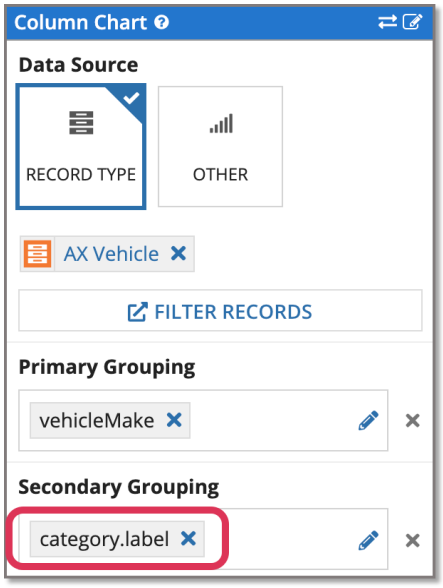
1. In your application, click **New**, and then **Interface**. Name the interface **AX_VehicleColumnChart**, and add a description, "An interface with the column chart of vehicles by make and category." Save it in **AX Interfaces**, and click **Create**.

2. In the **Components** palette, scroll down to **Charts**. Drag and drop a **Column Chart** component onto the canvas. Click the title **Column Chart**, and rename it to **Vehicles by Make and Category**.

3. In the **Component Configuration** pane, select **Record Type** as the **Data Source**. Start typing **AX Vehicle** into **Search record types**, and then select it from the list of auto-suggestions.

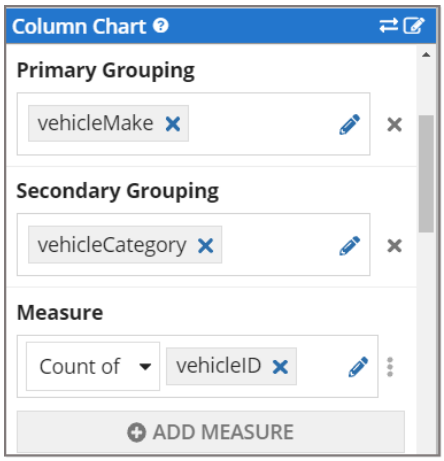
4. Once you select AX Vehicle , you will notice that the primary grouping as well as measure properties get auto-populated. The primary grouping should be by **vehicleMake**. If your chart has a different field, remove that value, and select **vehicleMake**.

5. Next, you will configure the **Secondary Grouping** property. Click **Add Grouping**, and click the arrow for the dropdown menu in the **Secondary Grouping** field. Scroll all the way down to the bottom of the dropdown menu. Hover over **category**, and then select **label**.

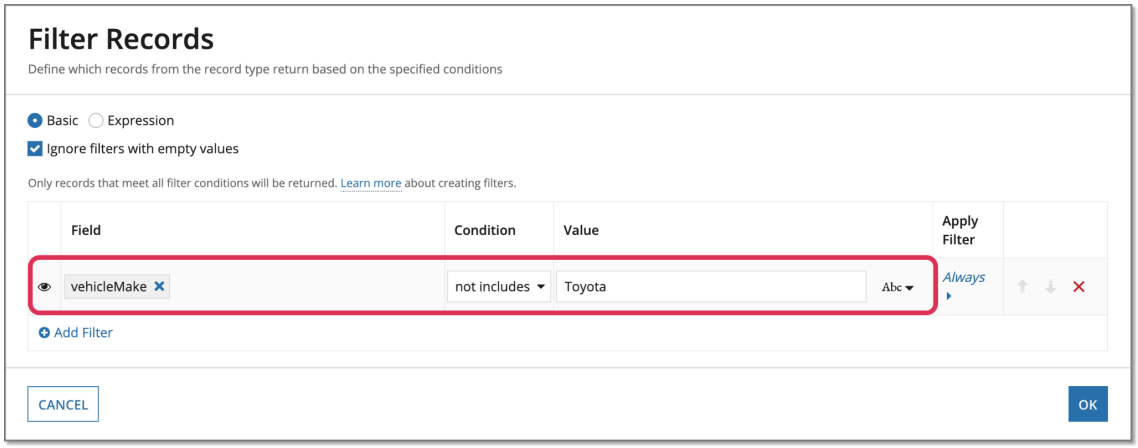


6. Scroll down to **Stacking**, and select **Normal** from the dropdown menu. Then, select the **Show data labels** check box.

7. Now, scroll up to **Measure**. This field got auto-populated correctly with the count of vehicle IDs. Keep the auto-populated value.



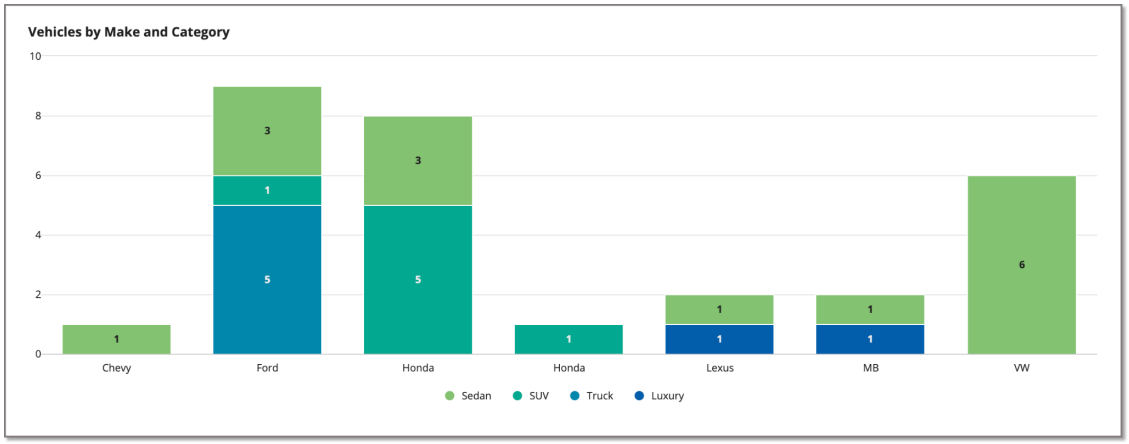
8. Next, add a record filter. Record filters are helpful when you want to show some, but not all record data. For example, if you want to hide Toyotas in the chart, click **Filter Records** at the top of the **Component Configuration** pane, and select **vehicleMake** as the **Field**. Select **not includes** in the **Condition** column. Type **Toyota** into the **Value** field.



Click **OK**. You will notice that Toyotas are now hidden.

9. Scroll down to **Color Scheme**, and select **Rainforest**. Note that you can also adjust the height of this chart. This selection is available below the **Color Scheme** menu.

10. Click **Save Changes**. Refer to the image below to see the final column chart:



Create a Pie Chart

Next, let's create a pie chart of vehicles by category.

1. In your application, create another interface. Name it **AX_VehiclePieChart**, and add a brief description: "An interface with the pie chart of vehicles by make." Save it in **AX Interfaces**. Click **Create**.

2. In the **Components** palette, scroll down to **Charts**. Drag and drop a **Pie Chart** component onto the canvas. Click the title **Pie Chart**, and rename it to **Vehicles by Make**.

3. In the **Component Configuration** pane, select **Record Type** as the **Data Source**. Start typing **AX Vehicle** into **Search record types**, and then select this record from the list of auto-suggestions.

4. Once you select AX Vehicle, you will notice that the primary grouping and measure properties auto-populate. The primary grouping should be by **vehicleMake**. If your chart has a different field, remove that value, and select **vehicleMake**.

5. Scroll down to **Series**, and select the checkboxes for **Show data labels** and **Show as percentage**.

6. Next, sort your chart from largest to smallest percentage. Scroll to the **Sort** property, and click **Add Sort**. Select **vehicleId_count_measure1** from the **Sort By** dropdown menu. Select the **Descending** radio button as the **Order**.

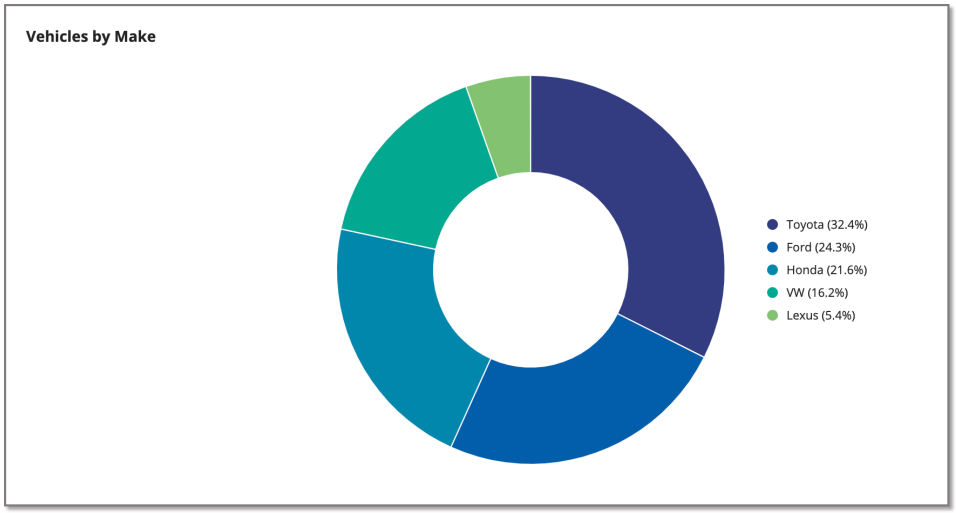
7. Scroll to the **Data Limit** property field. This property determines how many unique groupings are shown on the chart. As a best practice, pie charts are best used when the number of unique groupings is less than 5. Type **5** into **Data Limit** to limit this pie chart to the top five makes in the Acme Automobile fleet.

8. Scroll down to **Color Scheme**, and select **Rainforest**.

9. Scroll to **Style**, and note that you can toggle between the Donut and Pie styles for this chart.

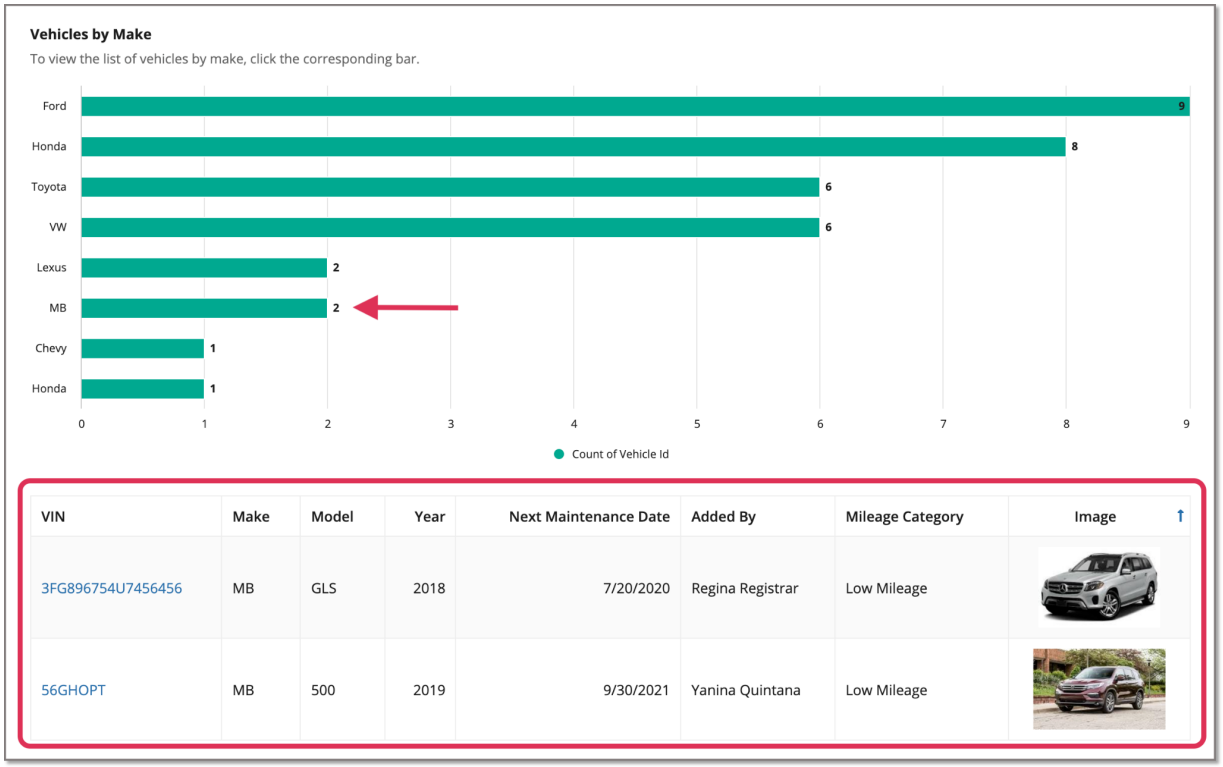
10. Scroll to **Series Labels**, and select **In Legend**.

11. Click **Save Changes**. Refer to the image below to see the final pie chart:



Create a Drillable Bar Chart

In this step, you'll create a drillable bar chart of vehicles by make. If you click any of the bars in this drillable chart, the grid with the vehicles of the selected make will display right below the bar chart:



Follow the steps below to create the drillable chart.

1. In your application, click **New**, and select **Interface**. Name the interface **AX_VehicleDrillableChart**, and add a description, "An interface with the drillable bar chart of vehicles by make." Save this interface in **AX Interfaces**. Click **Create**.

2. In the **Components** palette, scroll down to **Charts**. Drag and drop a **Bar Chart** component onto the canvas. Click the title **Bar Chart**, and rename it to **Vehicles by Make**.

3. In the **Component Configuration** pane, select **Record Type** as the **Data Source**. Start typing **AX Vehicle** into Search record types, and then select it from the list of auto-suggestions.

4. Once you select AX Vehicle, the primary grouping as well as measure properties get auto-populated. Your primary grouping should be **vehicleMake**. If your chart has a different grouping, remove that value, and select **vehicleMake**.

5. Scroll down to **Instructions**, and add an instruction, "To view the list of vehicles by make, click the corresponding bar." This will ensure that users know that the bar chart is clickable.

6. Select the **Show legend** and **Show data labels** checkboxes. Update the **Color Scheme** to **Rainforest**.

7. Scroll up to the **Sort** property, and click **Add Sort**. Select **vehicleId_count_measure1** from the **Sort By** dropdown menu. Select the **Descending** radio button as the **Order**.

8. Click **Save Changes**.

Add a Read-Only Grid

1. Now it's time to build the grid. In the **Components** palette, drag and drop a **Read-Only Grid** underneath the bar chart.

2. In the **Component Configuration**, select **Record Type,** and then choose **AX Vehicle** as your target record.

3. Remove the default label **Read-only Grid**. Also, scroll down to **Records Actions**, and click **X** next to **Add Vehicle** to remove it.

4. Deselect the **Show refresh button** and **Show search box** checkboxes.

5. Next, let's add a record filter to this grid to ensure that vehicles display by make. This can be done after you add the **make** rule input. Click the **plus sign** next to **Rule Inputs**, and type **make** into the **Name** field, and **Text** into the **Type** field. Click **Create**.

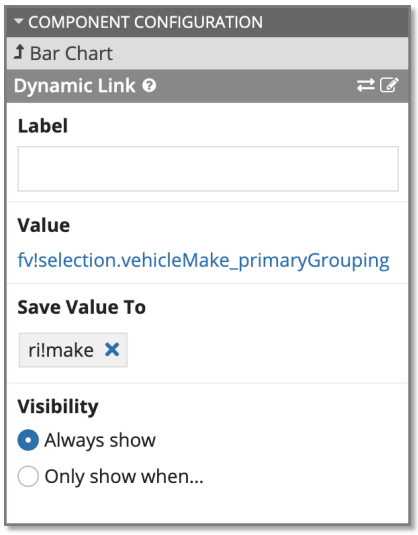
6. Click **Filter Records**. In the new dialog box, click **Add Filter**. Select **vehicleMake** in the **Field** column, and select **equal to** in the **Condition** column. In the **Value** column, click the **arrow** next to Abc, and select **Input/Variable** from the dropdown menu. In the **Value** dropdown, select **ri!make - Text**, and click **OK**.

7. Finally, let's configure the dynamic link that will link the bar chart to the grid. Select the bar chart in the canvas, and scroll to **Link** in the **Component Configuration** pane. Click **Insert Link**, and then click **Dynamic Link**.

8. Click **fv!selection**. Edit this expression to:

```
fv!selection.vehicleMake_primaryGrouping.
```

In the **Save Value To** field, select **ri!make**:



vehicleMake_primaryGrouping is the alias for the bar chart's Primary Grouping property. To look it up, go to **Primary Grouping** and click the **Expression Editor** icon next to it.

9. Click **Save Changes**. Your drillable chart is now fully configured. To test the chart, click any bar in the bar chart and see the list of vehicles in the grid dynamically update to the selected make.

10. To hide the grid when no make is selected, select the grid and scroll to the **Layout** menu. Expand it, and in **Visibility** select the **Only show when** radio button. Click **Edit Condition**, and enter the following expression: **not(isnull(ri!make))**. This expression will show the grid only if a make is selected.

11. Click **OK**, then **Save Changes**.

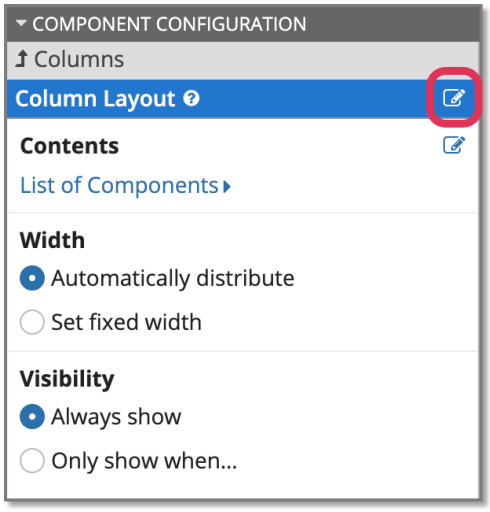
Create a Report Interface

Now, let's combine all three charts into a single interface.

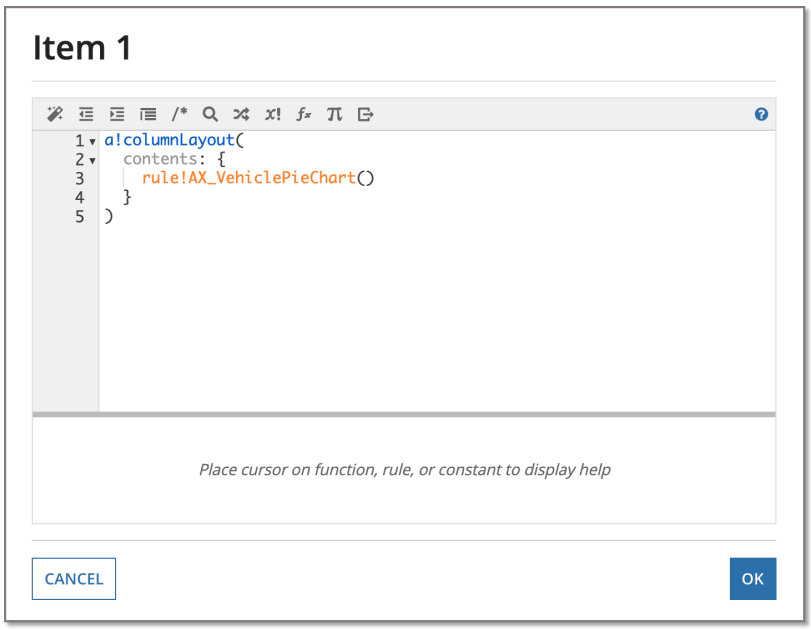
1. In your application, create a new interface. Name it **AX_VehicleReportInterface**. Add a brief description, "An interface that shows the charts of vehicles by category and make." Save it in **AX Interfaces**, then click **Create**.

2. In the **Components** palette, drag and drop a **Section** layout onto the canvas. Next, grab the **Columns** layout, and drop it inside the **Section** layout. Delete one column, and update the section label to **Fleet Data**.

3. Click the **first column** to select it, and then in the **Component Configuration** pane click the **Edit as Expression** icon located next to **Column Layout**:



4. In the **Expression Editor,** click within the curly brackets, and type **rule!AX_VehiclePieChart**. Click **OK**.



5. Click the **second column**, and then repeat the steps above, but this time type the following expression: **rule!AX_VehicleColumnChart**. Click **OK**.

6. Grab another **Section** layout from the **Component** palette, and drop it underneath the first section. Remove the default label for this section.

7. Next, let's add a local variable to this interface. In this exercise, we need a local variable to help us with holding the temporary make selection that we configured in the drillable bar chart.

   Local variables are added to an interface only in the Expression Mode, and they go at the very beginning of the interface expression, encapsulating the whole expression within parentheses. We will use the **a!localVariables** function to declare a local variable **local!selection**.

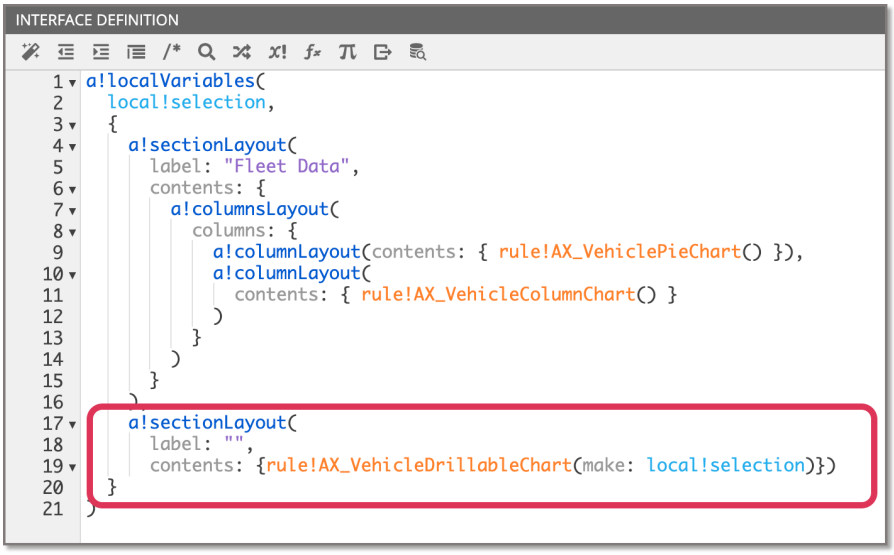
8. Click to access the **Expression Mode**:

9. Click before the curly bracket for the expression, and press **Enter**.

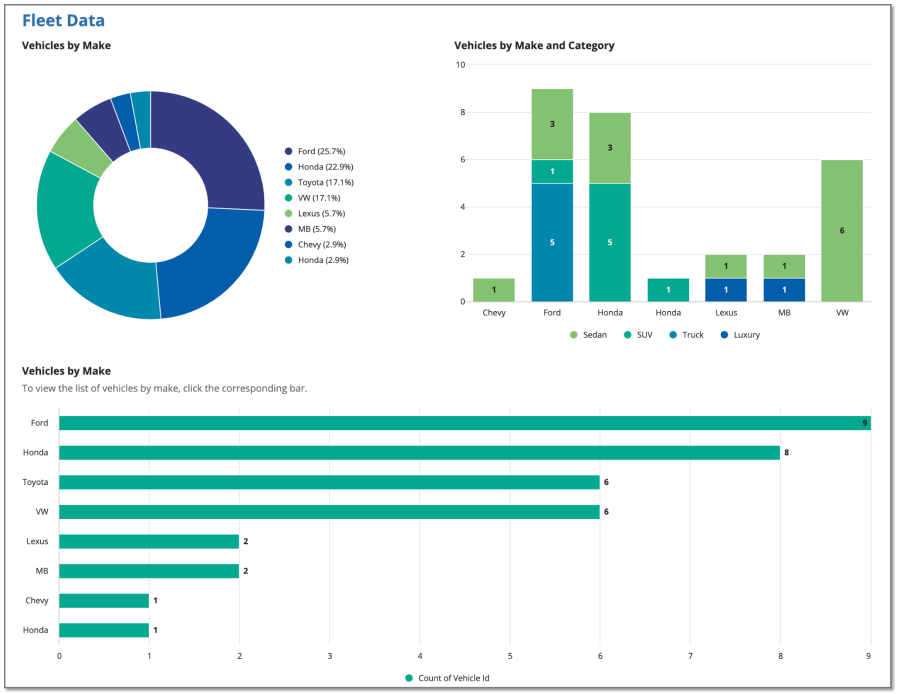
   - In line 1, start typing **a!localVariables**, and then select this option when it gets auto-suggested.
   - Since all local variables must be defined first in the code, create a new line right after a!localVariables, and type **local!selection**.
   - Add a **comma** after it, and delete the **closing parenthesis**.
   - Scroll to the end of the entire interface expression, and add the **parenthesis**.

10. Because we are already in the Expression Mode, let's add the drillable bar chart to the interface using an expression. Locate the last section that we added to this interface in the Design Mode, and click inside the curly brackets for **contents**. Type the following expression:

```
rule!AX_VehicleDrillableChart(make: local!selection)
```



11. Return to **Design Mode**, and click **Save Changes**. Your interface is now complete and ready to be used for the report object:



Create a Report Object

Now that you've combined the charts into a single interface, create a Report object. Follow the steps below to finalize your report.

1. In your application, click **New**, then **Report**. Name it **AX Fleet Report**, and add a brief description: "Report for Supervisors and Registrars that contains all relevant vehicle analytics." Select **AX_VehicleReportInterface** in the **Interface** field, and click **Create**.
2. Configure security for the report. Add the **AX All Users** as **Viewers** and **AX Administrators** as **Administrators**. Click **Save**.

Set this object aside for now. In the next exercise you'll add it as a page to your site.