Practical update course

GeneXus™ 16

PART 4

April 2020

TABLE OF CONTENTS

OBJECTIVE

In this practical course you will work with the new Dashboard object to integrate queries in an eye-catching way in your web applications.

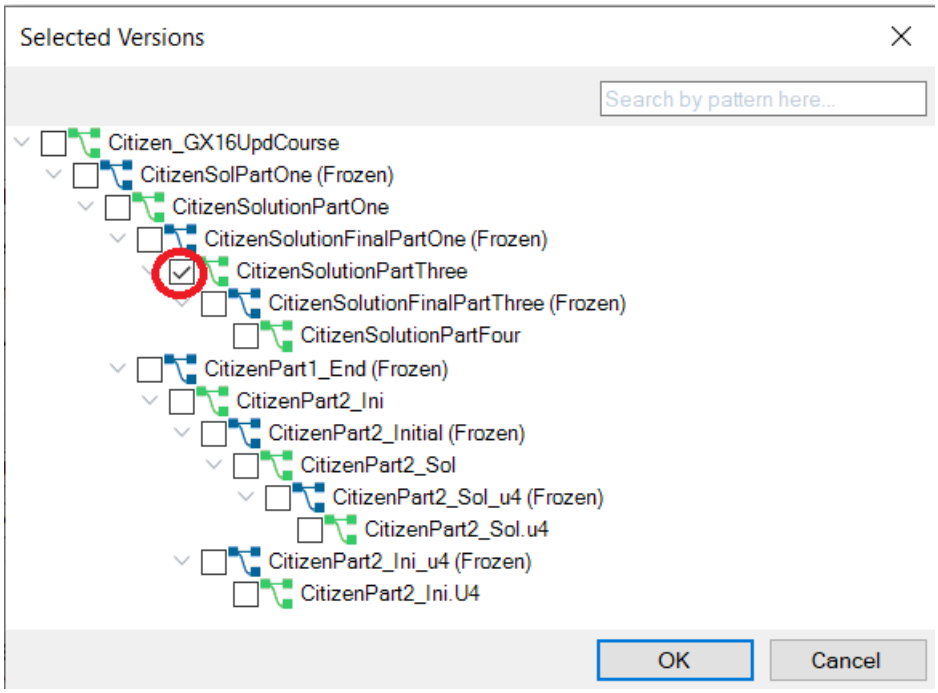
The instructions of these practice exercises have been updated according to upgrade 8 of GeneXus 16. If you do them with a later version, remember that there may be differences between what is shown here and what you see.

You will continue to use the Citizen KB as it was left from the previous practice exercises. If, for any reason, you no longer have this KB, in the following point you will find instructions to obtain a similar one.

Remember that the application you will work on is a simplification of an app for the municipality of a city, which offers a frontend (Web/SD/Conversational) so that citizens, using their user ID, can make claims (for example, for fallen trees, traffic lights that do not work, improperly parked cars, etc.), or carry out formalities (for example, to obtain a driver's license, refinance a debt with the municipality, install advertising elements, etc.), booking a time to be assisted by municipal staff. The frontend also shows the various cultural activities offered by the city. In addition, a web backoffice is available for certain municipal officials to manage the data and view statistics. To this backoffice you will add the dashboard.

STARTING KB

You will start working with the KB that you used in the previous practical course. If you've lost it or can't find it, create another from http://samples.genexusserver.com/v16 (KB called Citizen_GX16Course). Select the version called CitizenSolutionPartThree.



Note: For ODATA services to work, in the "ServiceDS1 (Service)" Data Store you will have to set this property again: Server name = https://services.odata.org/V4/(S(40gwjcqlhjmuyfayfnplov0o))/TripPinServiceRW/

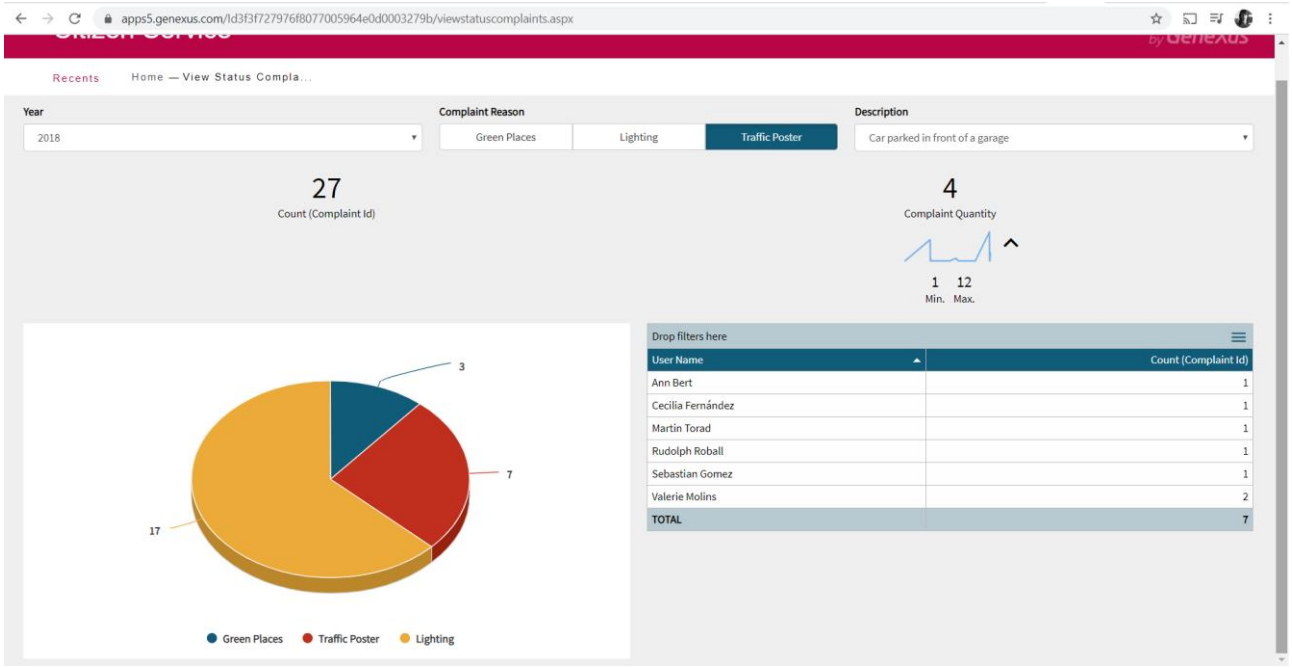
Do a rebuild before you start.

ENTITIES THAT WILL BE USED

Below the Entities_Trns folder you will find the entities that model the app's reality. These practice exercises will be focused on user complaints according to categories of complaints (reasons for complaints, ComplaintReason).

DASHBOARD OBJECT

The screen backoffice should be similar to the one below, for browsing the complaints made by users and filtering them by "Year," "Complaint Reason" and "Complaint Description":



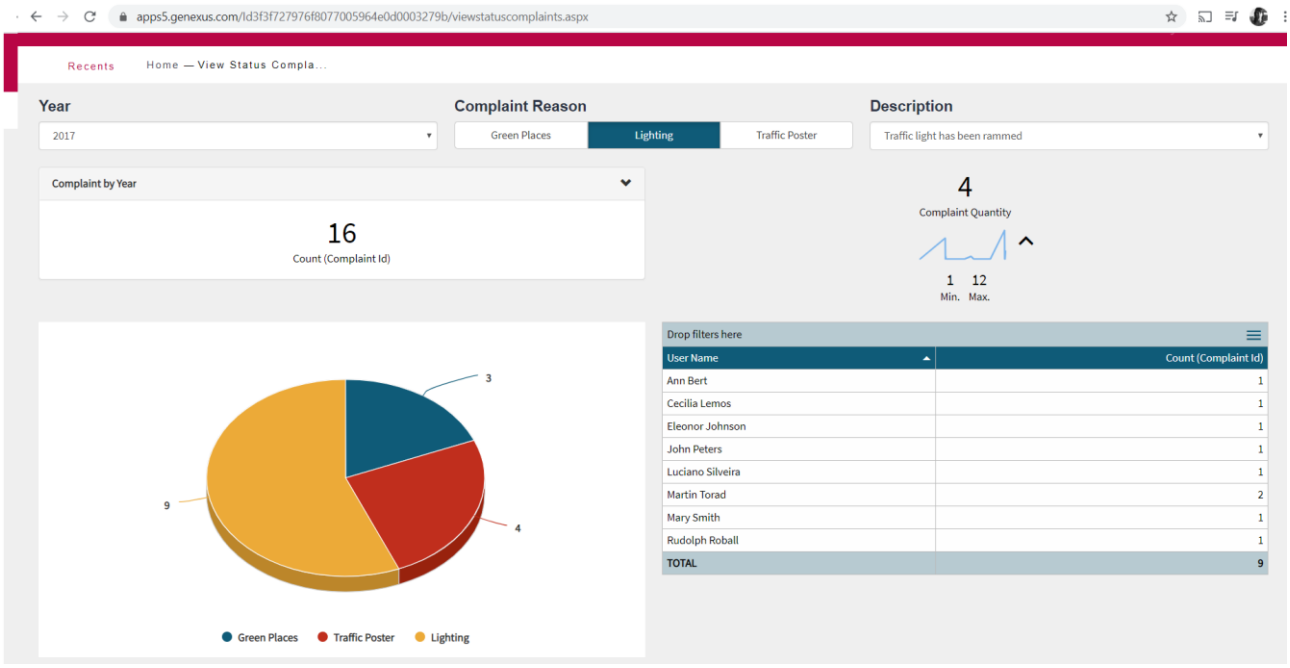
How do you implement it? After doing it, test it at runtime.

SOLUTION

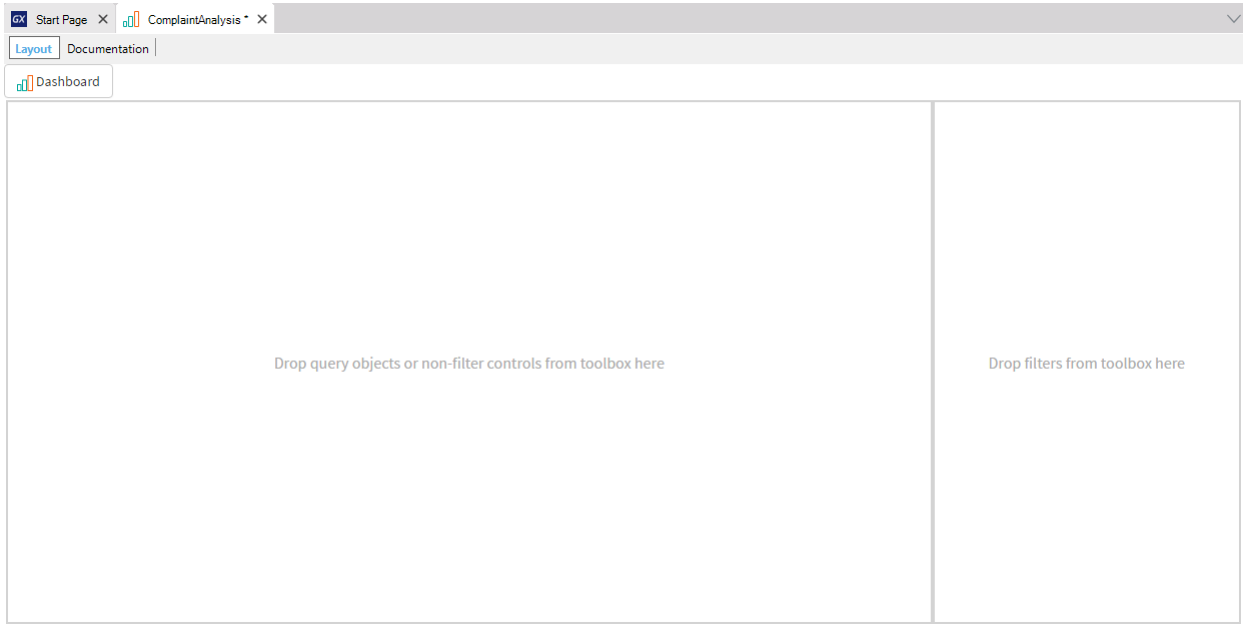
Create an object of Dashboard type (within the Reporting category) called "ComplaintAnalysis."

To use this dashboard you will need the queries that have already been created and are saved in the "Queries" folder.

The objective is to create a dashboard similar to the one shown in the image.



At first, the Dashboard has two areas: “Queries” and “Filters.”



Drag the queries to the Queries area in the locations shown below and set them with these types:

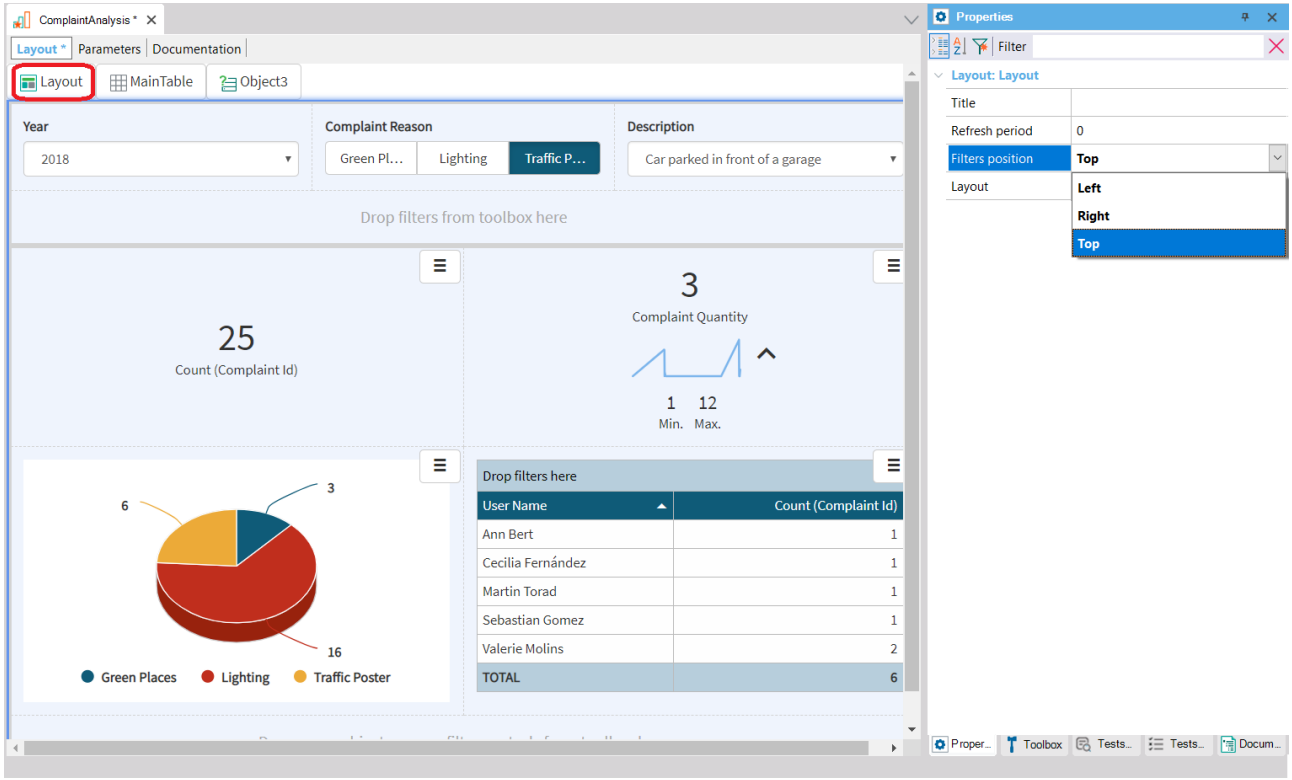
- Upper left corner – **QTYComplaintReasonbyYear** of Card type.
- Upper right corner – **QTYComplaint** of Card type.
- Lower left corner – **QTYComplaintReasonbyYear** of Chart type/Pie.
- Lower right corner – **QTYperUserByYearReasonDesc** of Pivot table type.

When the queries were placed, GeneXus automatically defined the filters they had. Now, change the captions of the filters and the type so that they look as the initial image.

- **“Year”** of static Combo Box Type, with values 2017 and 2018
- **“Complaint Reason”** of dynamic Radio Button Type, where the values are taken from the attribute “ComplaintReasonName”
- **“Description”** of dynamic Combo Box Type, where the values are taken from the attribute “ComplaintDescription” and the following condition is added: “ComplaintReasonName = &ComplaintReasonName;” in this way, the filters “Complaint Reason” and “Description” will be related.

Once the filters are configured, they can be changed because the Dashboard has live editing available.

Tip: To place the filters in the upper part of the Dashboard object, click on the Layout section to enable the properties. In “Filters position,” change the position to Top instead of Right.

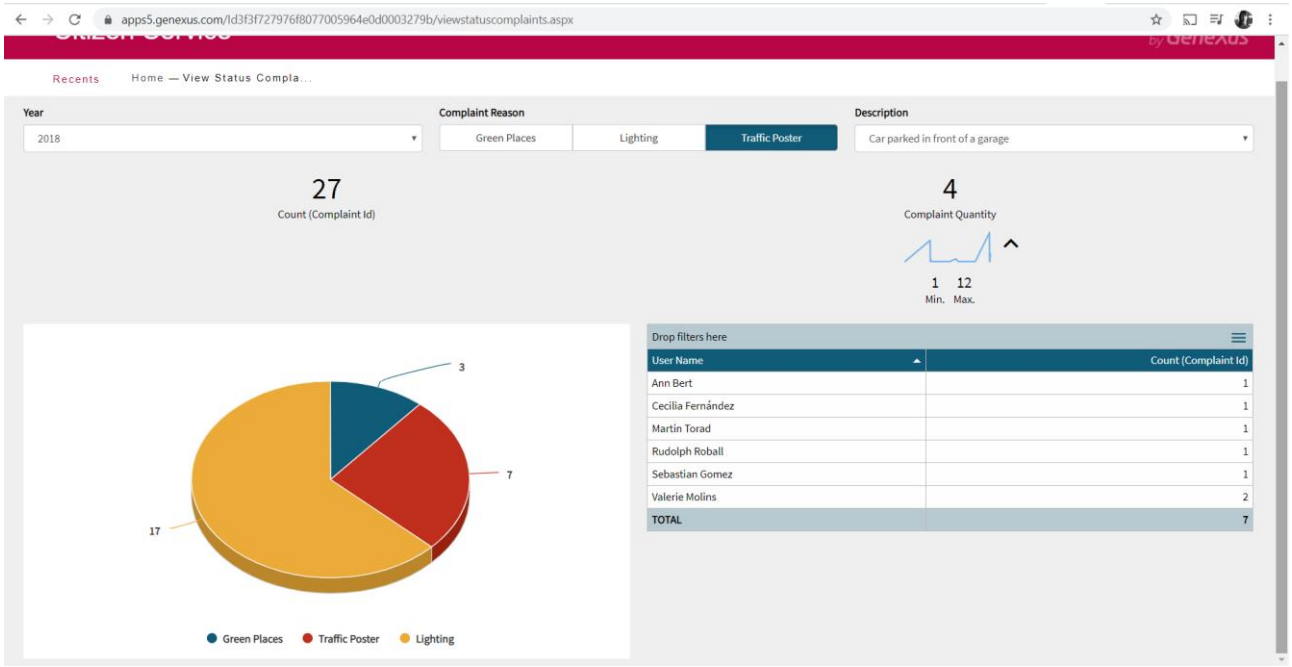


It is time to add the Dashboard to your application. To do so, create a Web panel called "ViewStatusComplaints." In this web panel, add a control of DashboardViewer type. Associate the previously created Dashboard object to the control.

To finish this exercise and be able to incorporate the web panel created to the backoffice, add it to the "ListPrograms" object with the following lines:

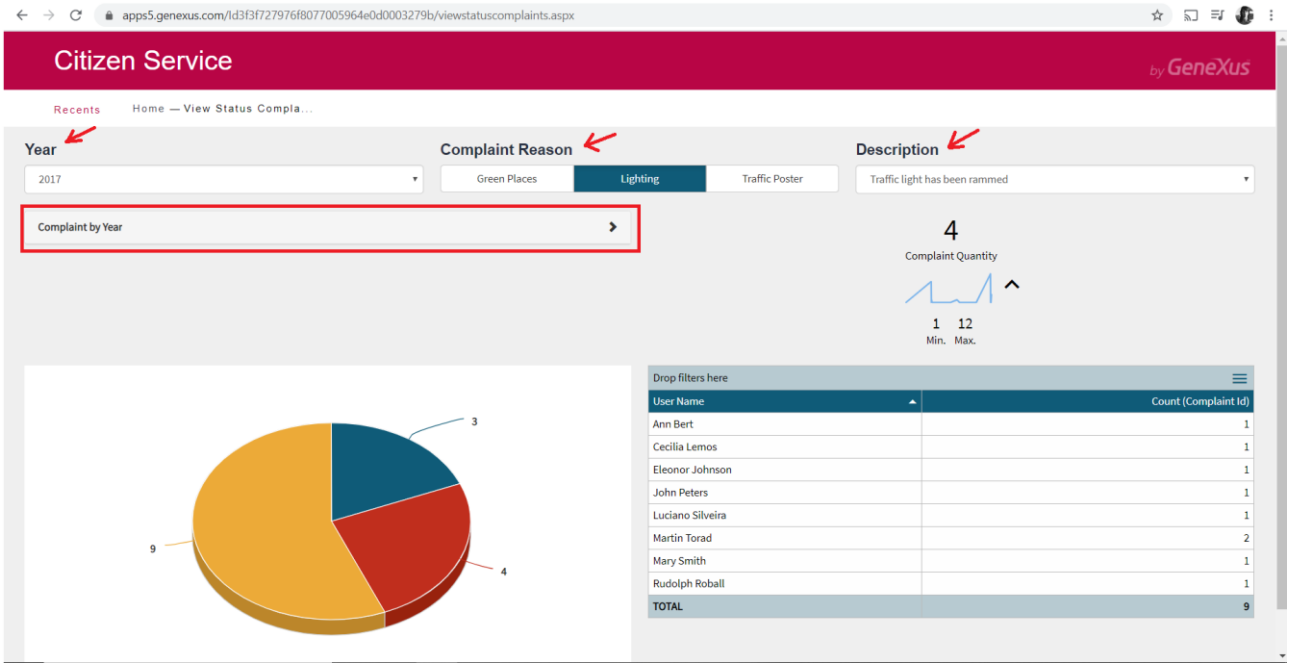
```
&name = !"ViewStatusComplaints"
&description = "View Status Complaints"
&link = ViewStatusComplaints.Link()

Do 'AddProgram'
```

In this way, when opening the Home of your backend (GeneXus/Web Folder) you will find the invocation to your "ViewStatusComplaints" and see the Dashboard running:



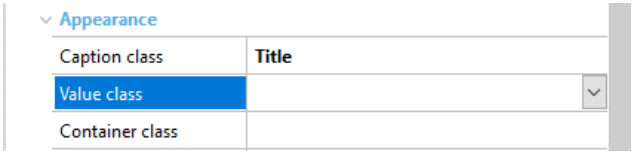
Now, change the filters and see how different queries behave.

In this stage, you will have to change the queries' appearance so that they look as shown below.

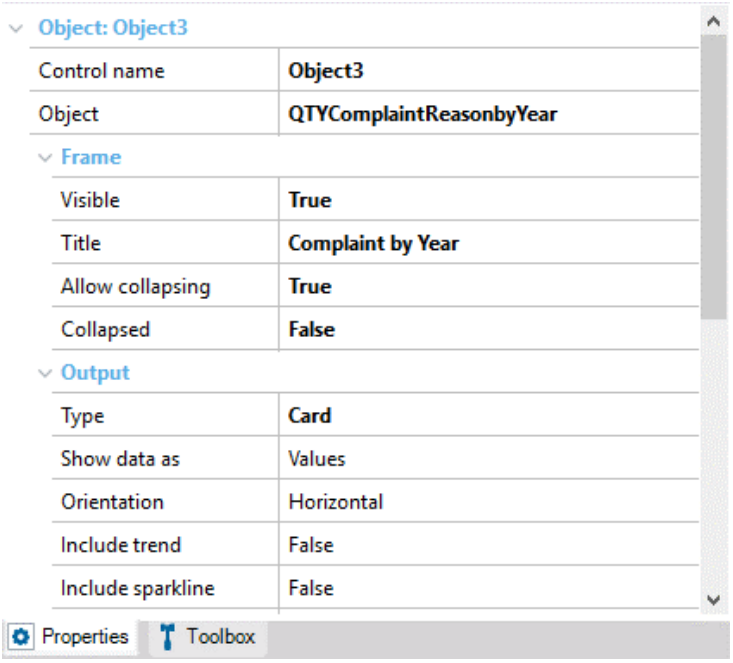


The changes you need to make are to change the size of the filters' title and collapse the query set as a Card.

Tip: To change the class associated with the Title, open the properties in Appearance, Caption class, and select the Title class.



To change the query QTYComplaintReasonbyYear, change from Frame to Visible, enter a title for it and enable it to collapse.



Once you've made all these changes, save the dashboard and run the application again, because these changes are not seen inside GeneXus; they can be seen at runtime.

SOLUTION KB

You can download the KB containing the solution to these practice exercises from GeneXus Server to compare results. This version is called CitizenSolutionPart4.

Note: For ODATA services to work, in the "ServiceDS1 (Service)" Data Store you will have to set this property again: Server name = https://services.odata.org/V4/(S(40gwjcqlhjmuyfayfnplov0o))/TripPinServiceRW/