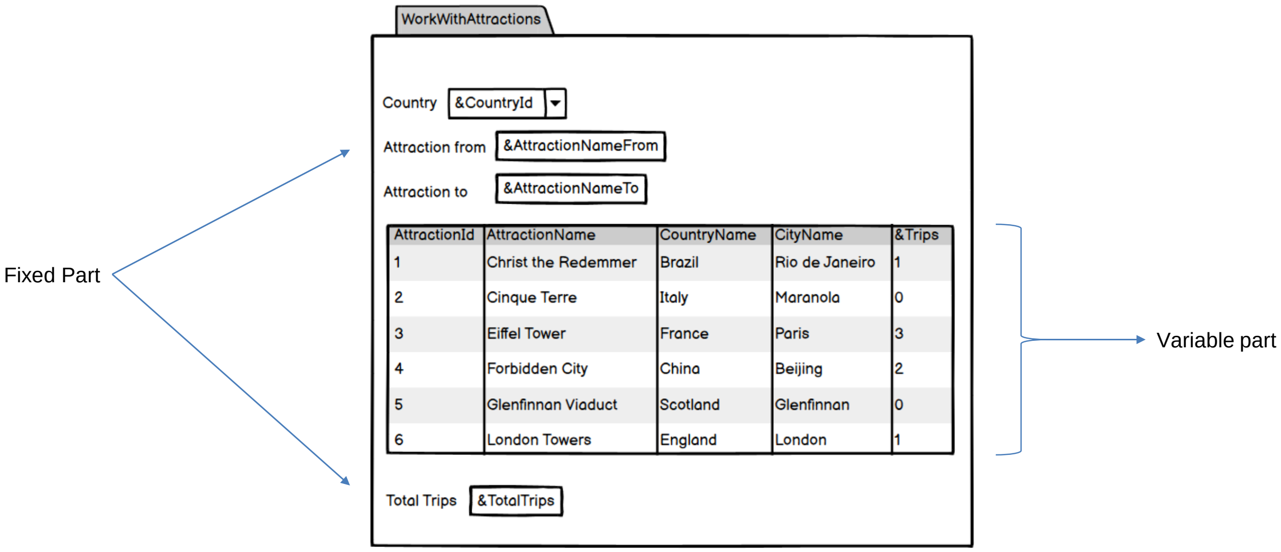
Logic for Loading Data on a Panel Screen

GeneXus™

We will see how the screen of a panel object is loaded, depending on the GUI components included in it.

Parts of a panel object



WorkWithAttractions

Country &CountryId

Attraction from &AttractionNameFrom

Attraction to &AttractionNameTo

AttractionId	AttractionName	CountryName	CityName	&Trips
1	Christ the Redemmer	Brazil	Rio de Janeiro	1
2	Cinque Terre	Italy	Maranola	0
3	Eiffel Tower	France	Paris	3
4	Forbidden City	China	Beijing	2
5	Glenfinnan Viaduct	Scotland	Glenfinnan	0
6	London Towers	England	London	1

Total Trips &TotalTrips

Fixed Part

Variable part

In a panel object we can identify two different parts.

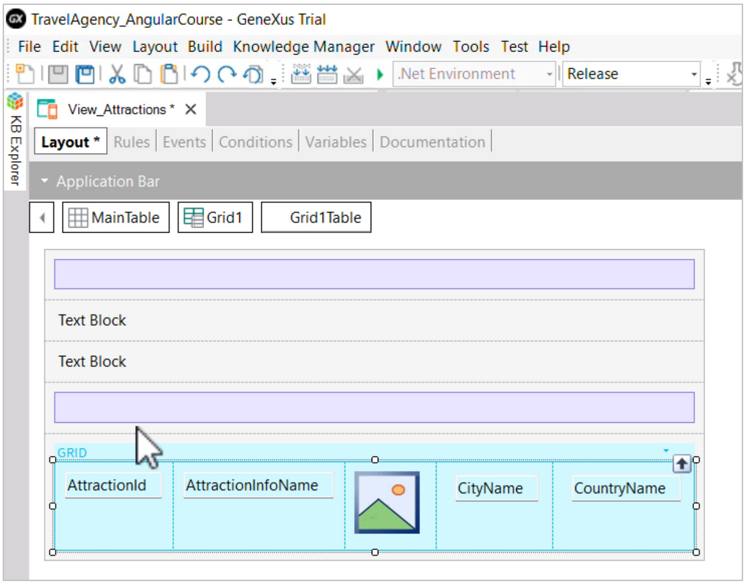
The fixed or flat part is the part that contains everything contained in the form that is not included in any grid. This includes controls such as text blocks, combos, buttons and others that we can drag from the toolbar.

The second part is variable, and is made up of the grid or grids that have been included in the form.

A panel object will always have a fixed part, and there can be a variable part for each of the grids in the panel.

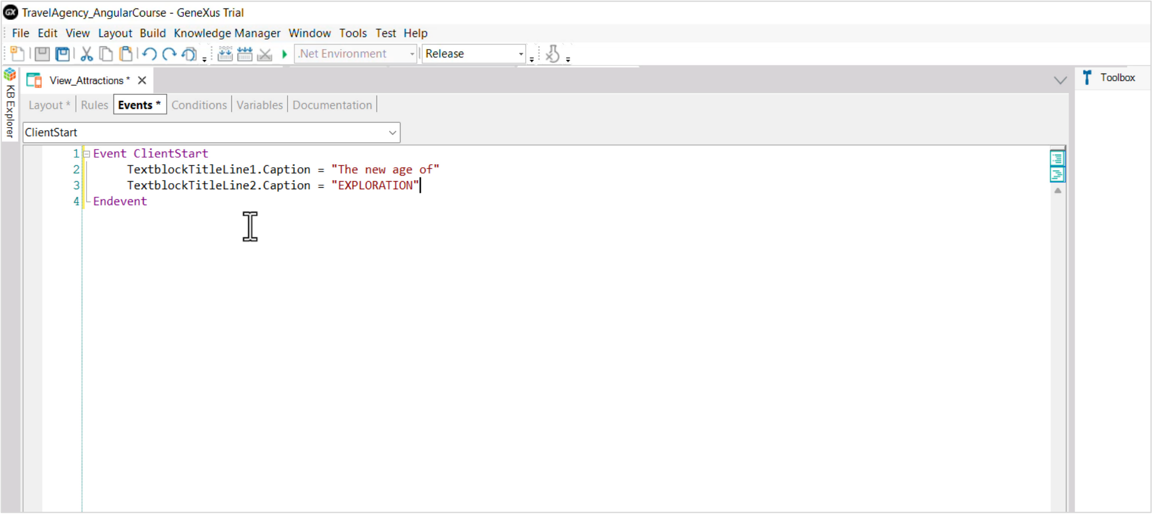
Panel objects differ from web panels in that the fixed part is loaded independently of the variable part.

Demo


TravelAgency_AngularCourse - GeneXus Trial
File Edit View Layout Build Knowledge Manager Window Tools Test Help
.Net Environment
Release
KB Explorer
View_Attractions *
Layout * Rules Events Conditions Variables Documentation
Application Bar
MainTable
Grid1
Grid1Table
Text Block
Text Block
GRID
AttractionId
AttractionInfoName
CityName
CountryName

Before moving on, let's go to GeneXus and add some controls to the fixed part of the View_Attractions panel we had created. Above and outside the grid, we drag a table, then a textblock that we call TextblockTitleLine1, then another textblock that we call TextblockTitleLine2, and below it another table control.

Demo

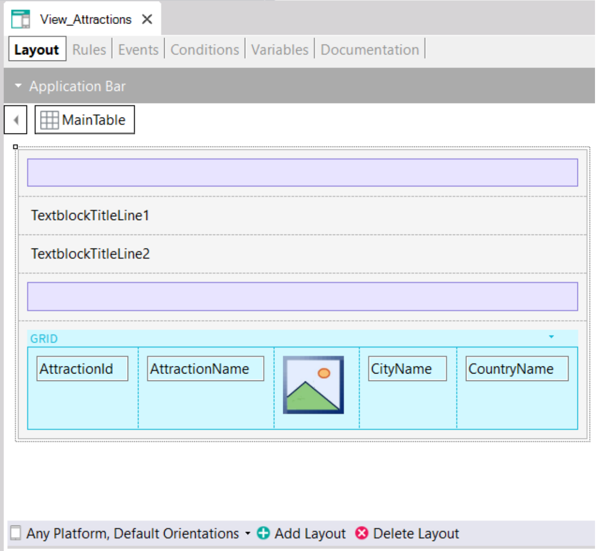


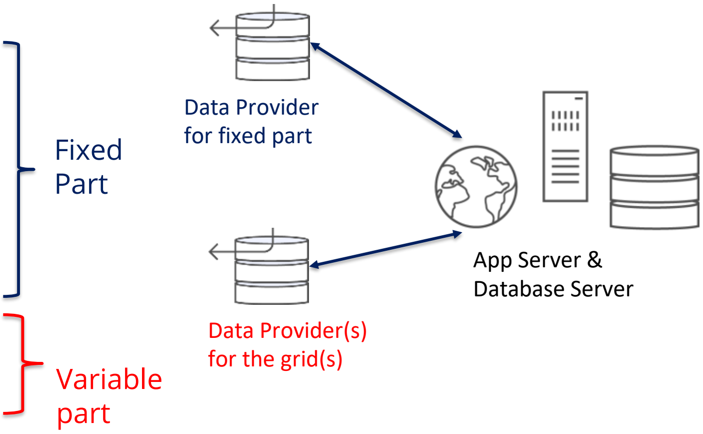
TravelAgency_AngularCourse - GeneXus Trial
File Edit View Layout Build Knowledge Manager Window Tools Test Help
.Net Environment
Release
KB Explorer
View_Attractions *
Toolbox
Layout * | Rules | Events * | Conditions | Variables | Documentation
ClientStart

```
Event ClientStart
      TextblockTitleLine1.Caption = "The new age of"
      TextblockTitleLine2.Caption = "EXPLORATION"
Endevent
```

Now in the ClientStart event we assign to its Caption properties the texts we want to see on screen for the title, "The new age of" for the first textblock, and "EXPLORATION" for the second one. We save and continue with the analysis of the panel data loading.

Parts of a panel object

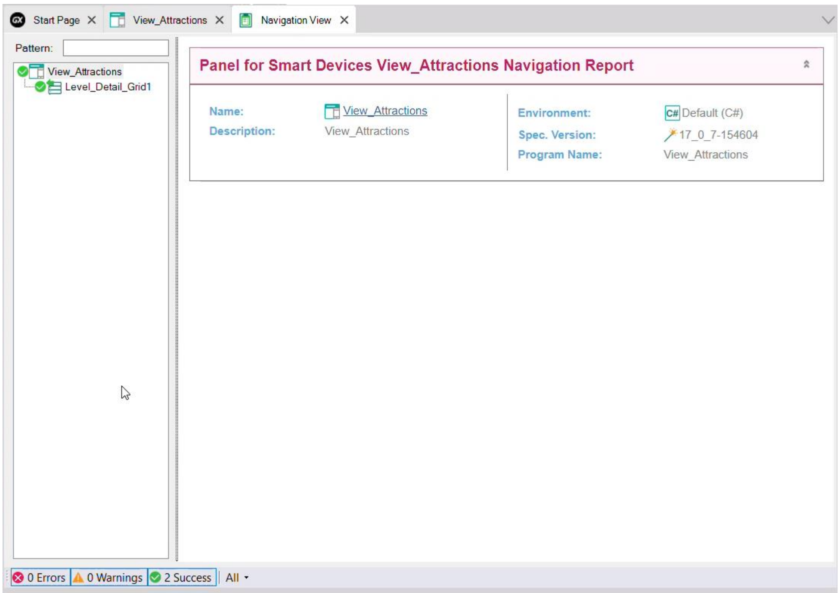



Fixed Part
Variable part
Data Provider for fixed part
Data Provider(s) for the grid(s)
App Server & Database Server

For each part (fixed and grid) GeneXus will automatically generate independent Data Providers that will be published as services on the server and will access the database to obtain the necessary information to load the fixed and variable parts of the panel.

We will not see these data providers in the Knowledge Base because GeneXus will generate and maintain them, but we will be able to see the data accessed in the navigation list.

Navigation list of a panel



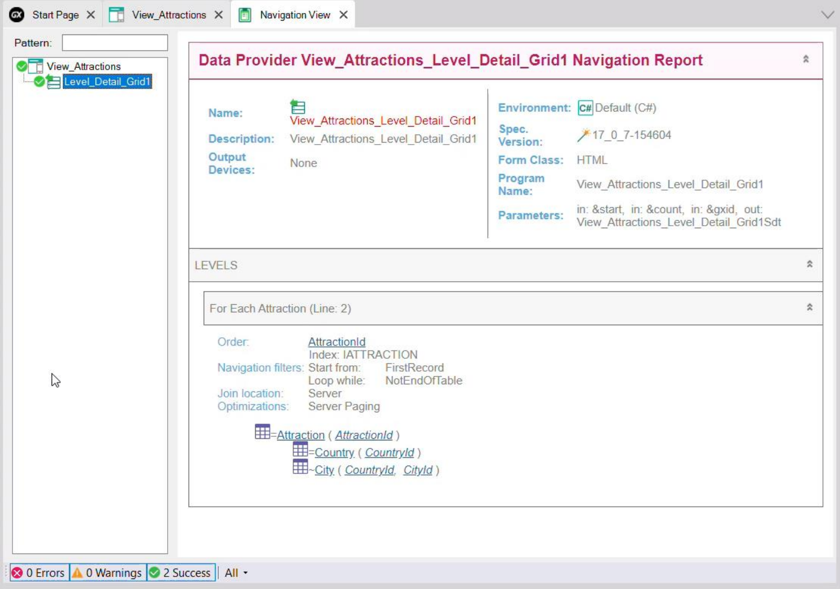
Panel for Smart Devices View_Attractions Navigation Report

Name: View_Attractions
Description: View_Attractions
Environment: Default (C#)
Spec. Version: 17_0_7-154604
Program Name: View_Attractions

In the navigation list of the View_Attractions panel, we can see that below the node of the panel, there is an entry named Level_Detail_Grid1 corresponding to the variable part of the panel, which in this case is composed of Grid1.

If the panel had more than one grid, a detail node would be displayed for each grid as each one would have its own navigation.

Navigation list of a panel



Start Page | View_Attractions | Navigation View

Pattern:

- View_Attractions
  - Level_Detail_Grid1

Data Provider View_Attractions_Level_Detail_Grid1 Navigation Report

Name: View_Attractions_Level_Detail_Grid1
Description: View_Attractions_Level_Detail_Grid1
Output Devices: None

Environment: C# Default (C#)
Spec. Version: 17_0_7-154604
Form Class: HTML
Program Name: View_Attractions_Level_Detail_Grid1
Parameters: in: &start, in: &count, in: &gxid, out: View_Attractions_Level_Detail_Grid1Sdt

LEVELS

For Each Attraction (Line: 2)

```
Order:              AttractionId
                    Index: IATTRACTION
Navigation filters: Start from:    FirstRecord
                    Loop while:    NotEndOfTable
Join location:      Server
Optimizations:      Server Paging

        =Attraction ( AttractionId )
            =Country ( CountryId )
            ~City ( CountryId, CityId )
```

0 Errors | 0 Warnings | 2 Success | All

If we now select the node named Level_Detail_Grid1, we see that it is the list of the Data Provider View_Attractions_Level_Detail_Grid1.

This is the data provider that was automatically created to load Grid1, published as a service on the server and invoked by the panel to access the database and retrieve the attractions.

In the Environment section, we can see that the data provider was generated in C# (.NET) as part of the backend code.

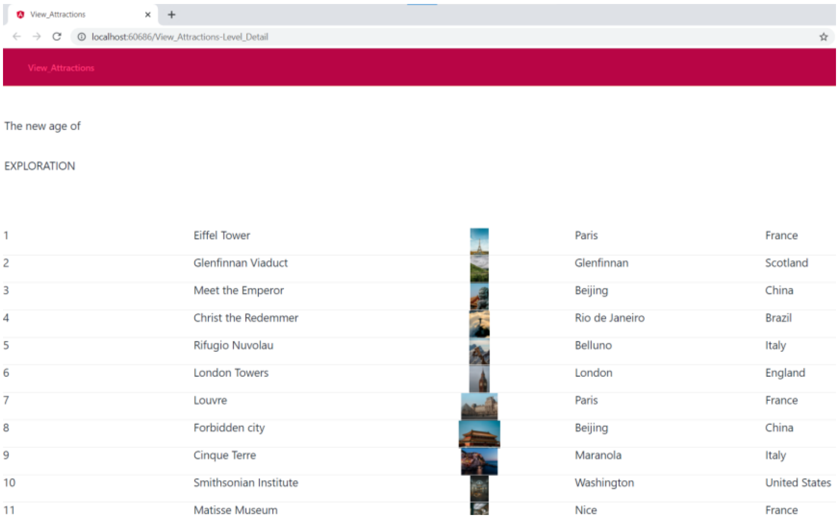
In the Parameters section, we see that the data provider receives some data from the grid and returns a loaded SDT named View_Attractions_Level_Detail_Grid1, which will contain the data to load the grid.

The listed information is similar to what we would see in a web panel with Attraction base table, since the panel grid is running through the Attraction table to show the tourist attractions; for each attraction, it accesses the Country table to retrieve the name of the country and the CountryCity table to retrieve the name of the city of the attraction.

As the fixed part and the variable parts of the panel are loaded separately, we can anticipate that the base tables of each part are completely independent of each other. This means that in a panel object with a single grid there can be 2 base tables, one for the fixed part and another for the grid, as opposed to web panel objects where in this case the base table was a single one.

In another video, we will discuss in detail how the base tables of each part of a panel are determined.

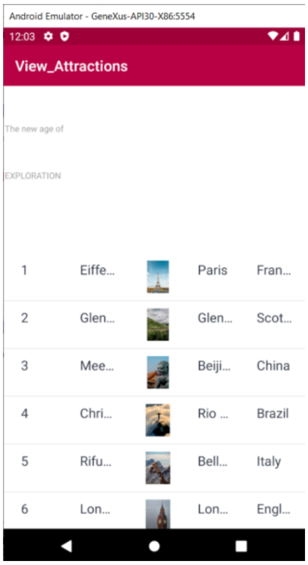
Running the same panel on different platforms



The new age of

EXPLORATION

1	Eiffel Tower		Paris	France
2	Glenfinnan Viaduct		Glenfinnan	Scotland
3	Meet the Emperor		Beijing	China
4	Christ the Redemmer		Rio de Janeiro	Brazil
5	Rifugio Nuvolau		Belluno	Italy
6	London Towers		London	England
7	Louvre		Paris	France
8	Forbidden city		Beijing	China
9	Cinque Terre		Maranola	Italy
10	Smithsonian Institute		Washington	United States
11	Matisse Museum		Nice	France



The new age of

EXPLORATION

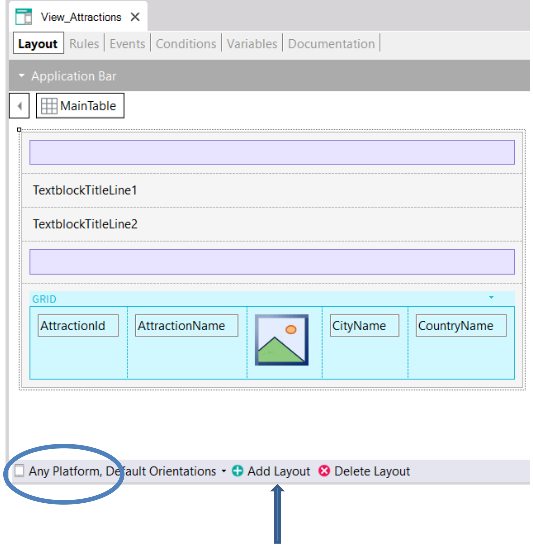
1	Eiffe...		Paris	Fran...
2	Glen...		Glen...	Scot...
3	Mee...		Beiji...	China
4	Chri...		Rio ...	Brazil
5	Rifu...		Bell...	Italy
6	Lon...		Lon...	Engl...

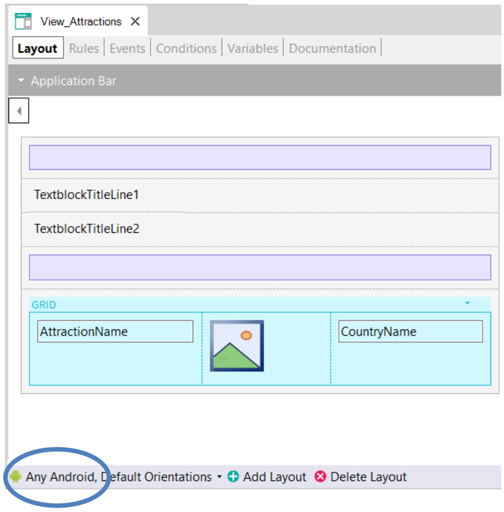
The same panel object can be run without changes both on a web platform – when generated in Angular, for example – or on native mobile platforms when generated in Android or iOS.

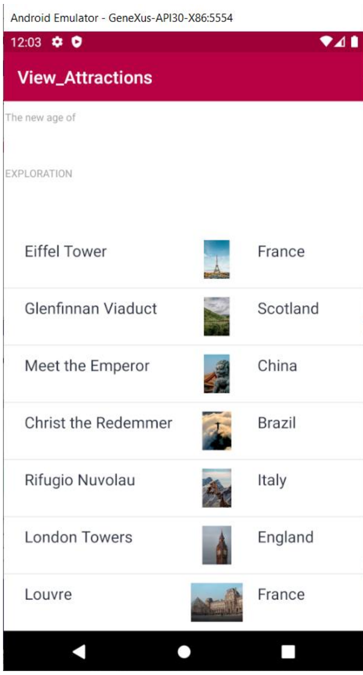
Here we see the execution of the same panel on a web desktop generated with Angular and on an Android cell phone.

Due to the space available, the information does not look the same in each case.

Layouts of a panel







Eiffel Tower	France
Glenfinnan Viaduct	Scotland
Meet the Emperor	China
Christ the Redemmer	Brazil
Rifugio Nuvolau	Italy
London Towers	England
Louvre	France

To ensure that the design of each screen is appropriate, we can create different layouts for each platform. Each layout will be independent of the others, and in each one we will be able to arrange the elements of the graphic interface as we want.

We can also create several layouts for the same platform. For example, if I'm only viewing information or if I'm editing it and I want the appearance or the amount of information displayed to be different in each case.

The form of a panel is a container of one or more layouts and each one may contain a certain number and distribution of different controls, but the logic for loading data on the screen will be the same as above; i.e. each layout may have a fixed part and one or more variable parts if it contains grids.

Therefore, for each of them, the necessary data providers will be created independently to obtain the information from the database when required.

In future videos, we will discuss these topics in more detail.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications