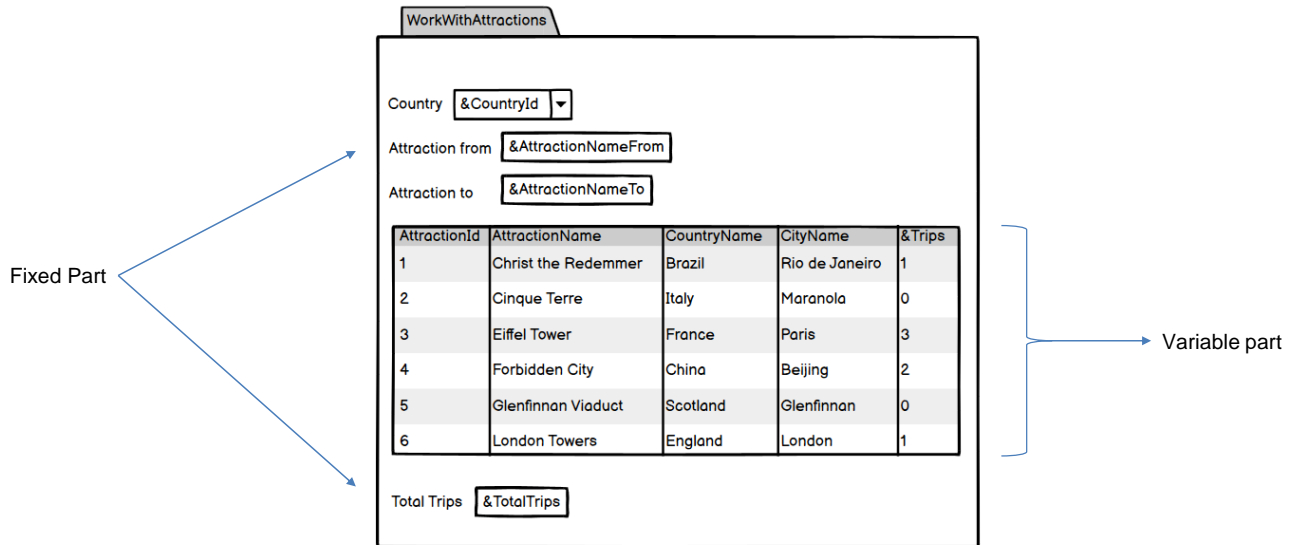


Logic for Loading Data on a Panel Screen

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



In a panel object we can identify two different parts.

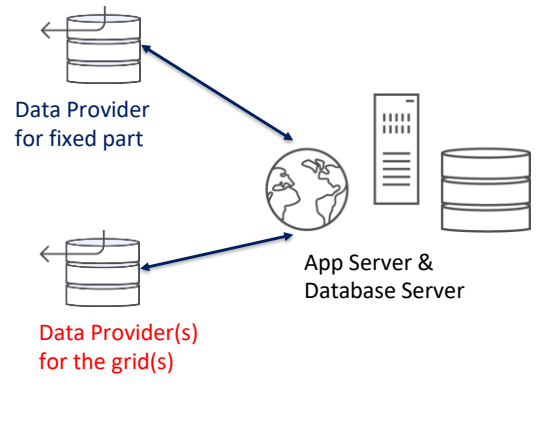
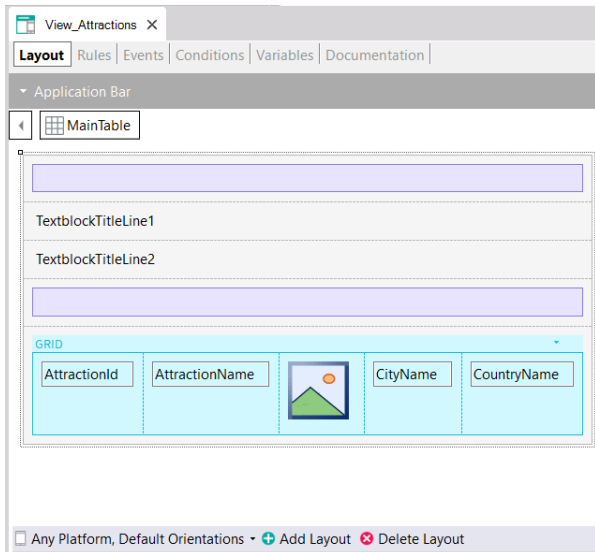
The fixed or flat part is the part that contains everything contained in the form that is not included in any grid. This includes controls such as text blocks, combos, buttons and others that we can drag from the toolbar.

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.

Parts of a panel object

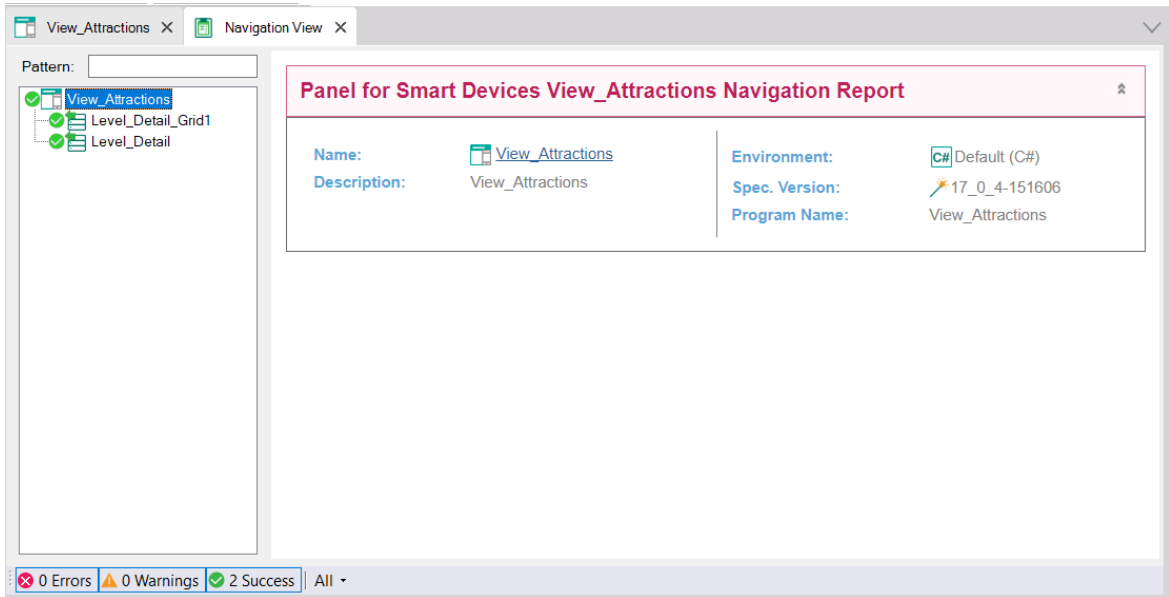


For each part (fixed and grid) GeneXus will automatically generate independent Data Providers that will be published as services on the server and will access the database to obtain the necessary information to load the fixed and variable parts of the panel.

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 by each one in the navigation list.

Let's look at the navigation list of the View_Attractions panel. It has only two text blocks for the title lines and separator tables in its fixed part and a grid with attraction data in the variable part.

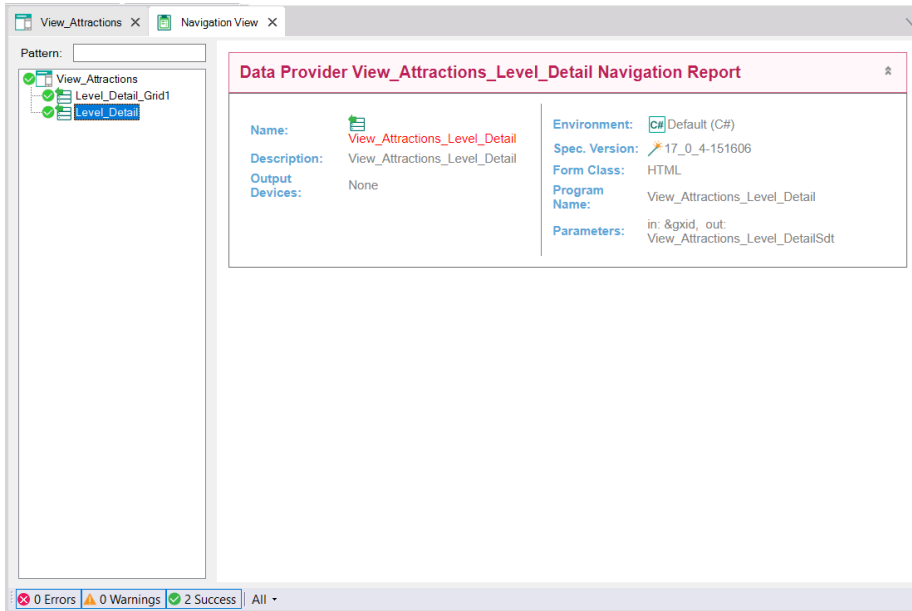
Navigation list of a panel



In the navigation list of the View_Attractions panel, under the node corresponding to the panel there are two entries: one named Level_Detail_Grid1 corresponding to the variable part of the panel – which in this case is made up of the Grid1 – and another one called Level_Detail that will show the loading of the fixed part of the panel.

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

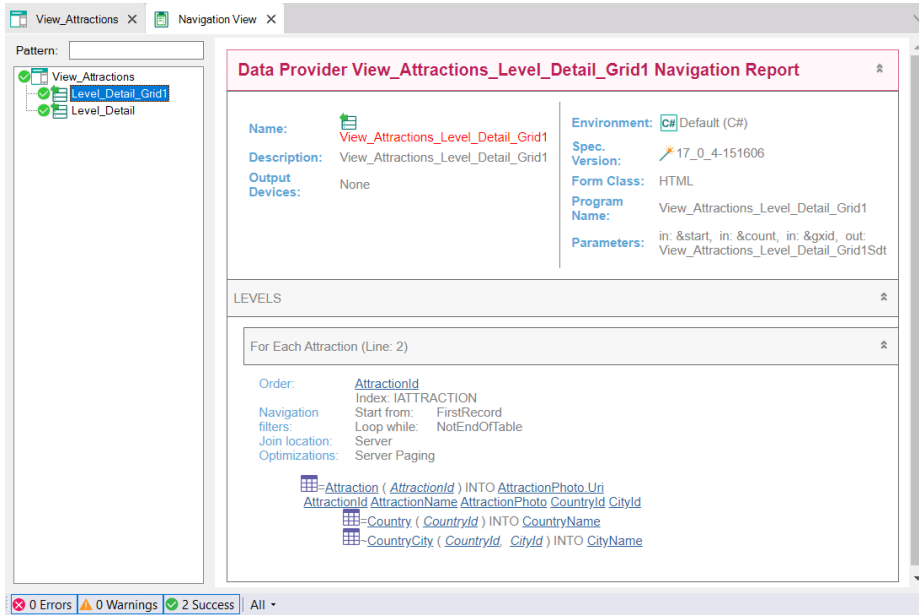


By selecting the Level_Detail node, we see that it is the navigation list of the data provider View_Attractions_Level_Detail.

The list is empty. Since the only fixed part of the panel has two text blocks with fixed data for the title, it is not necessary to go to the database to retrieve information.

If in this fixed part we had added a Dynamic combo box or any other control that requires accessing data from the database to display information, the navigation to access that data would appear here.

Navigation list of a panel



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.

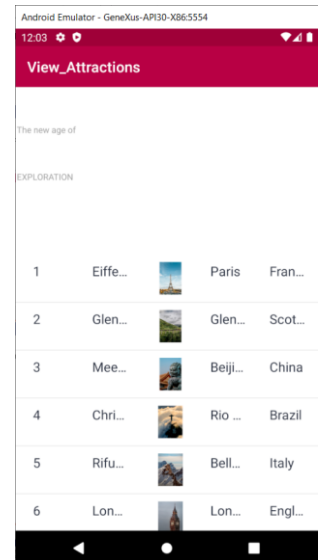
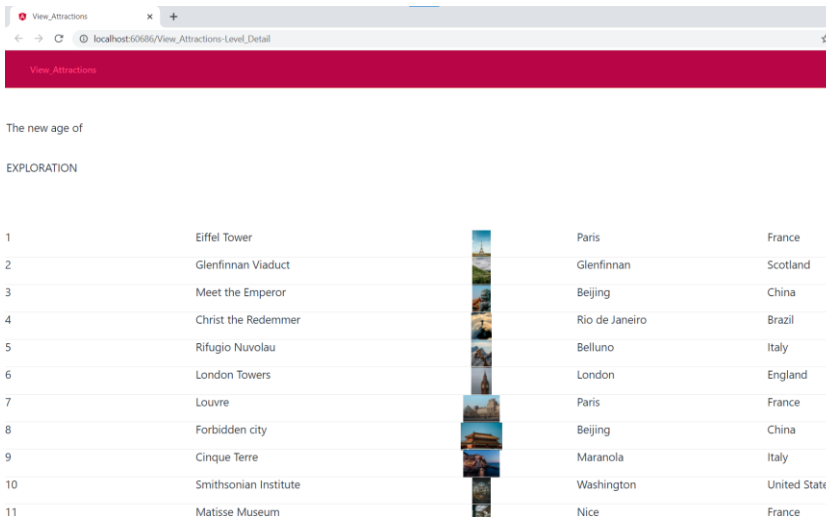
In the Parameters section, we see that the data provider receives some data from the grid and returns a loaded SDT named `View_Attractions_Level_Detail_Grid1`, which will contain the data to load the grid.

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

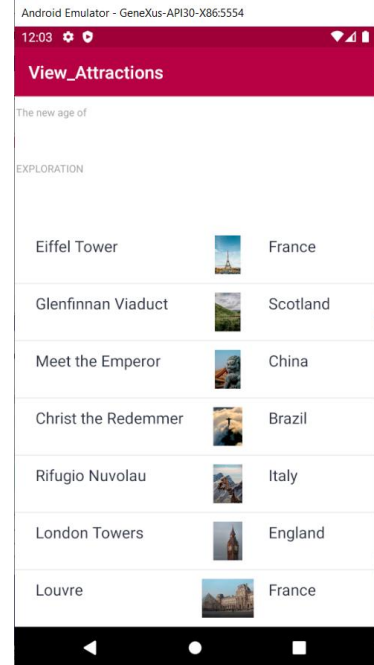
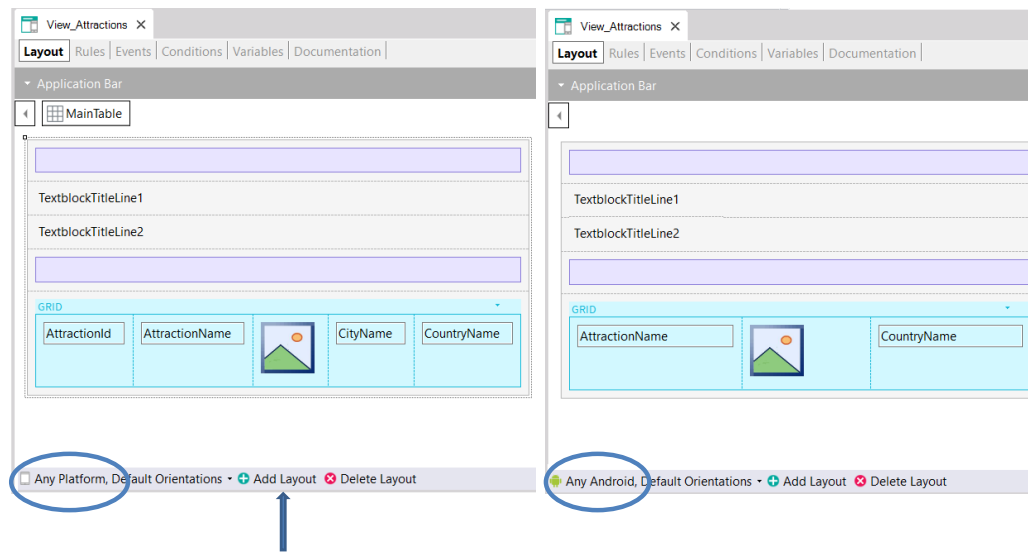


The same panel object can be run without changes both on a web platform – when generated in Angular, for example – or on native mobile platforms when generated in Android or iOS.

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



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