Horizontal Grids, Flex Grids, and Flex Control

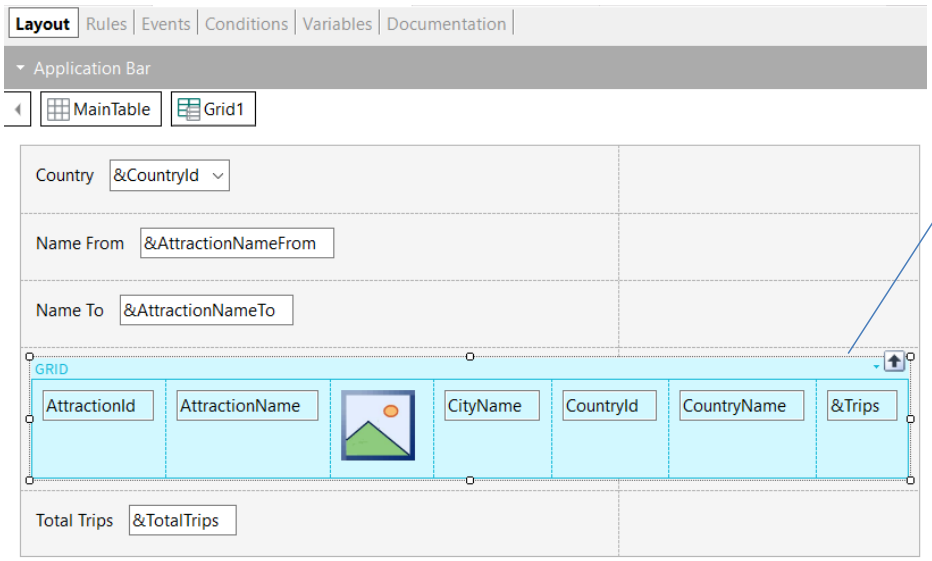
GeneXus™

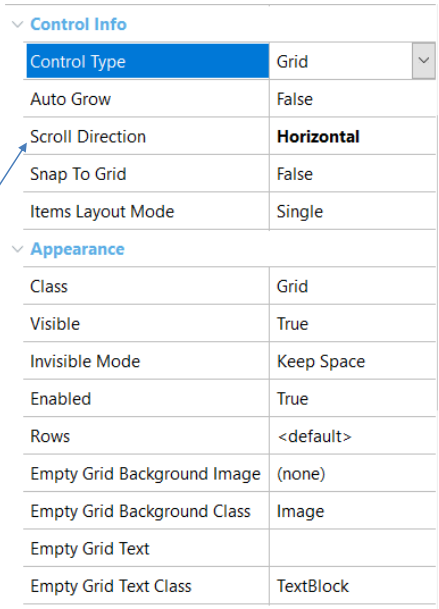
When we saw the finished travel agency application at the beginning of this course, we could scroll the attractions horizontally as in a carousel.
In this video, we will see how to configure a grid to adopt this behavior, as well as other types of grids and flexible controls available to develop an application.

Smart Grid

Let's start with the standard grid, which can be set to scroll horizontally.

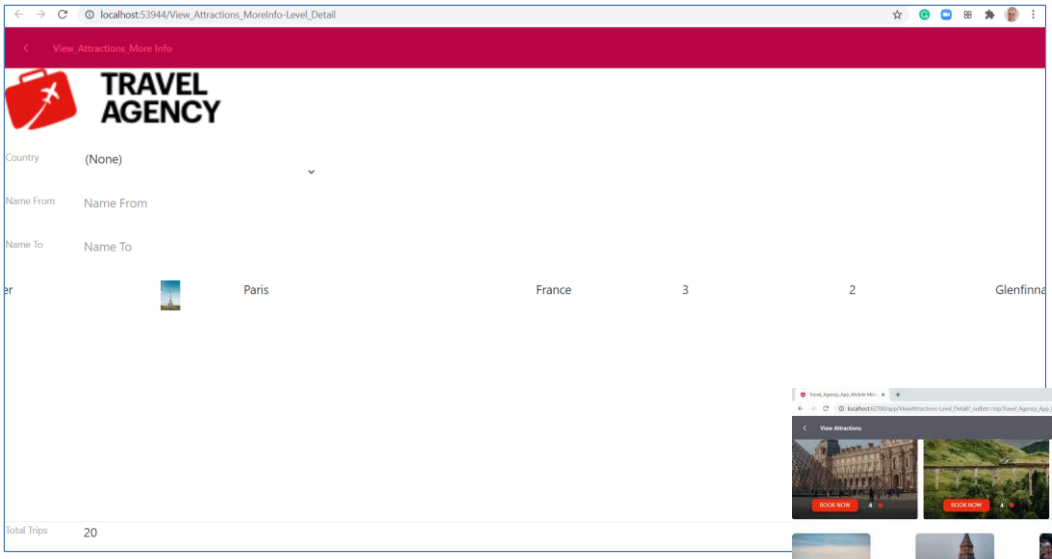
Setting horizontal scroll in a grid

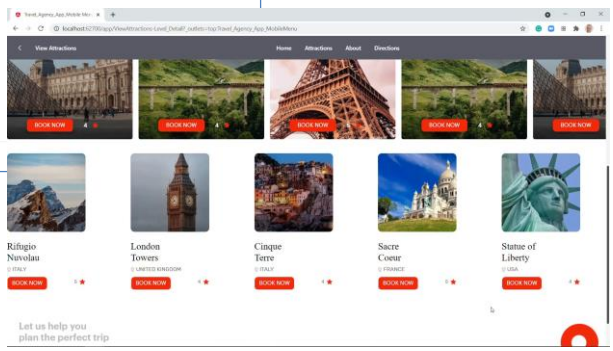




In upgrade 11 of GeneXus 16, the grid control became a Smart Grid, so there are properties that allow us to change how the information will be displayed, in a more flexible way.

If all we want is to scroll the records horizontally, we can set the Scroll Direction property of the grid to Horizontal and if we execute...

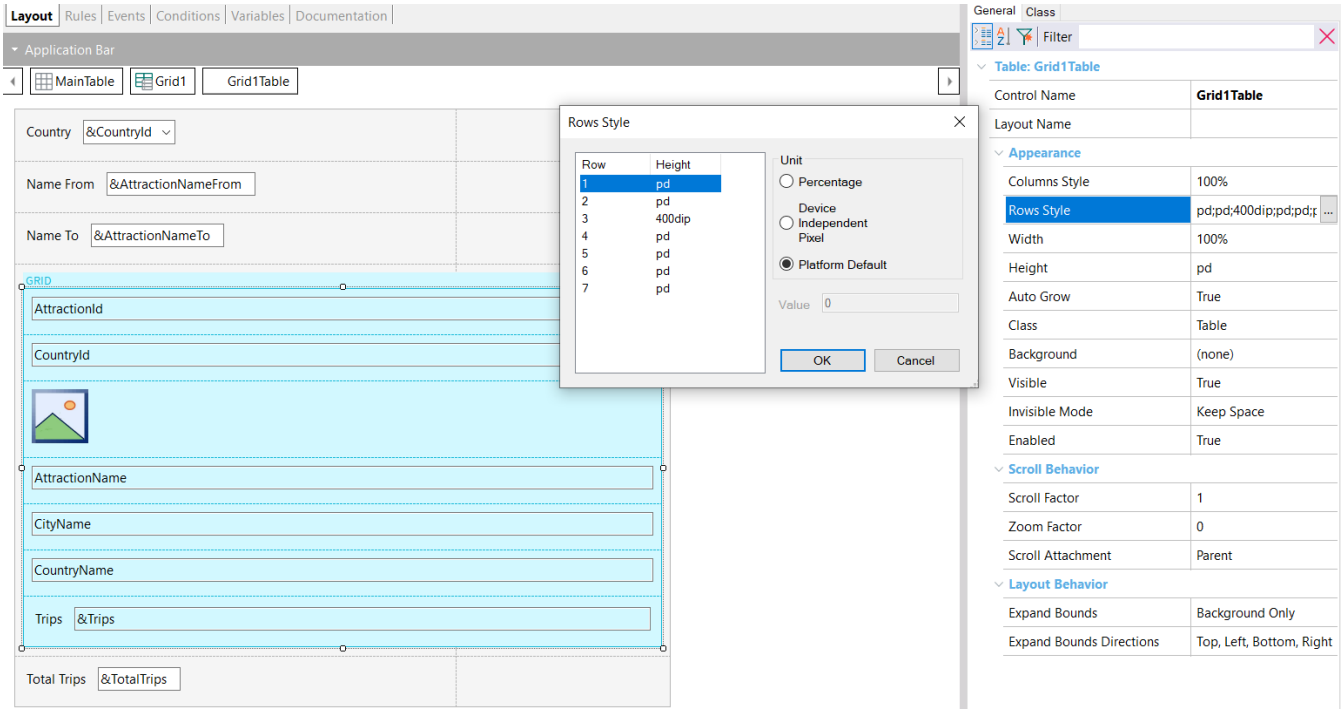




Now the grid takes up a single row and by clicking on it we can move to the right and then back to the left.

If we go back to the finished application, to the section of the screen where you scroll through all the available attractions, in addition to the horizontal scrolling, several attractions can be viewed at the same time, so we need to find out how to configure this.

Moreover, for each attraction, the attraction data is arranged differently from what we achieve in our execution.



That is, if we compare the design of our grid columns with the finished application, we not only have to change how many items go in each row or column, but we have to change the grid design as well.

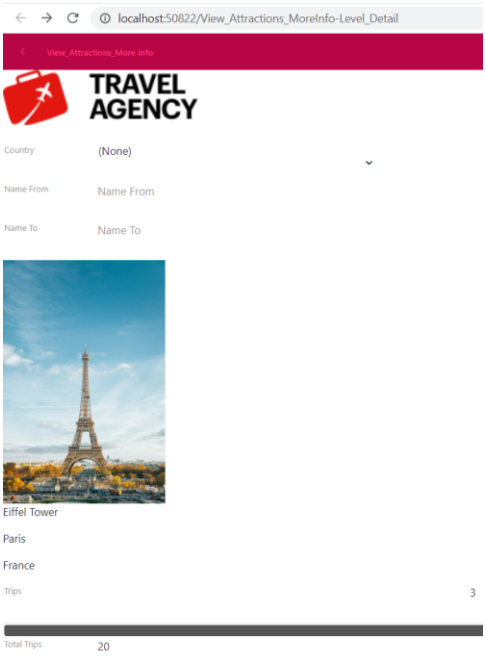
To do this, we go to the CountryId attribute and move it below AttractionId. Then we do the same with the rest of the attributes of the columns, placing them all downwards. Next, we select the attributes AttractionId and CountryId, we set visible to False and the Invisible mode property to Collapse space.

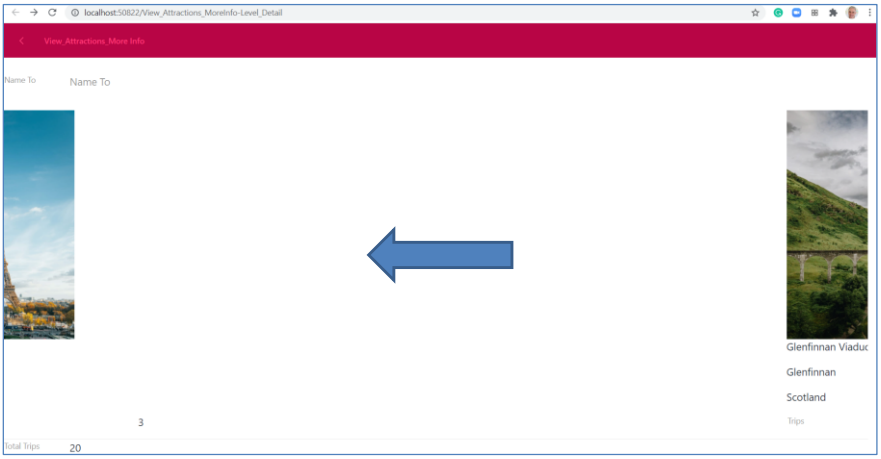
Let's adjust the properties of the table inside the grid. To do so, we select the AttractionId attribute and then select the Grid1Table table. We adjust the Rows Style property and give 400 dips to the third row which is the one in the picture.

As we want to scroll horizontally, we select the grid and set the Scroll Direction property to Horizontal.

We run it.

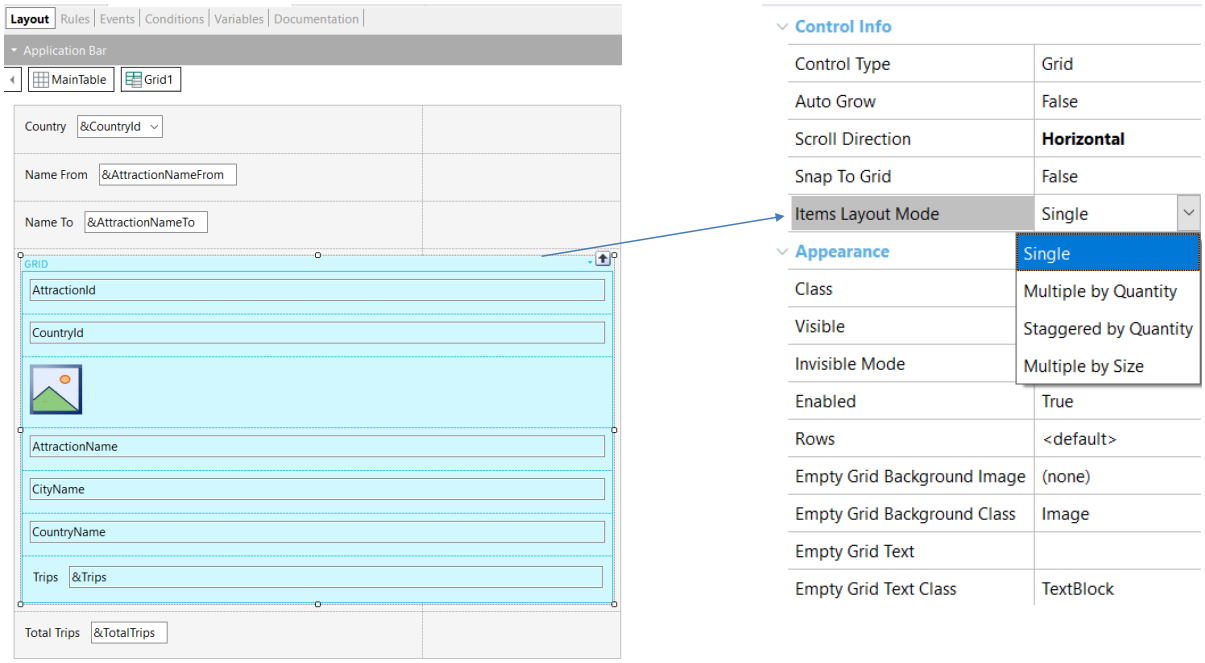
New design and horizontal scrolling





Now the data is displayed more similarly to what we wanted, but we still see only one record at a time on the screen.

Property: Items Layout Mode

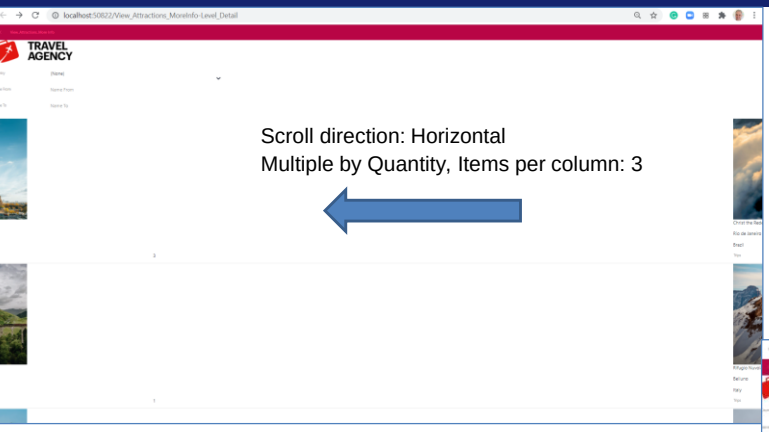


In addition to the Scroll Direction property of the grid, we have the Items Layout Mode property to change the way in which the grid data is displayed.

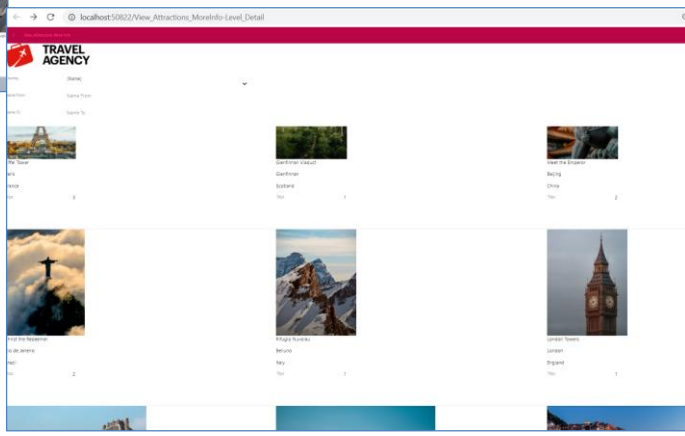
Several values are available:

Single – This is the default value and causes the grid to display only one item per row.
Multiple by Quantity – This value indicates that a fixed number of elements will be displayed in each row/column of the grid. It will be determined by the properties Items Per Row if the scroll direction is vertical, or Items Per Column if the scroll direction is horizontal.
Staggered by Quantity – The grid is displayed in a staggered layout, where elements can have different sizes. The number of items per row or column is determined by the Items Per Row or Items Per Column properties, depending on the direction of the grid scrolling.
Multiple by Size - The number of items displayed in each row/column will be determined by the size of each item. To achieve this, we use the values of the properties Maximum Width, Minimum Width if the scroll direction is vertical, and Maximum Height and Minimum Height if the scroll direction is horizontal.

We are going to assign the value Multiple by Quantity, and since we have horizontal scroll direction, the Items Per Column property is displayed. We set it to 3 and execute.



Scroll direction: Horizontal
Multiple by Quantity, Items per column: 3



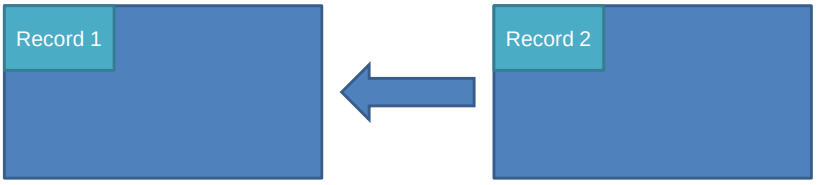
Scroll direction: Vertical
Multiple by Quantity, Items per rows: 3

We see a single column on the page and in each column 3 attractions that are visible if we scroll down. When paging to the right, the following 3 attractions appear downwards.

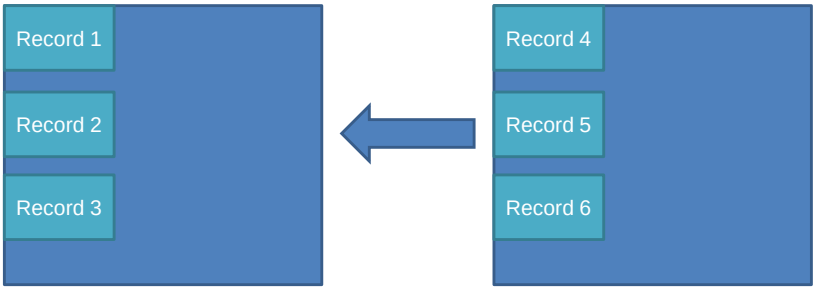
This is not what we want; let's change the scrolling to vertical and leave the properties with the same values.

Now we see 3 attractions in each row and to see the next ones we must scroll down.

Property: Items Layout Mode, Scroll direction = horizontal



Single



Multiple by Quantity

Multiple by Size

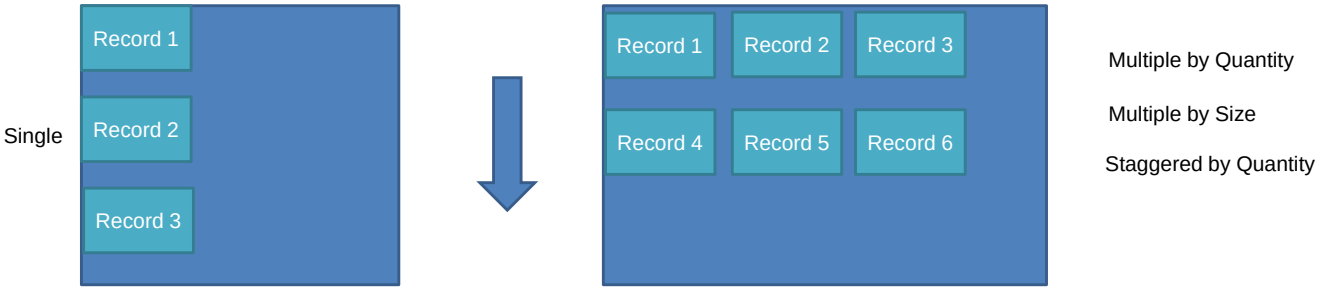
Staggered by Quantity

In summary, in the standard grid (smart grid), if scrolling is horizontal as in our example, we can define how many records we want to see per column; i.e. within the same column downwards.

But we will always view one column at a time.

The number of records per column displayed downwards can be defined as a fixed value, or depending on the height (minimum and maximum) that we want to give to each item. If we define to do it with a fixed value we can see it staggered or not, depending on whether we choose Staggered by Quantity, or Multiple by Quantity, respectively.

Property: Items Layout Mode, Scroll direction = vertical



If scrolling is vertical, we can configure how many records we want to see in each row.

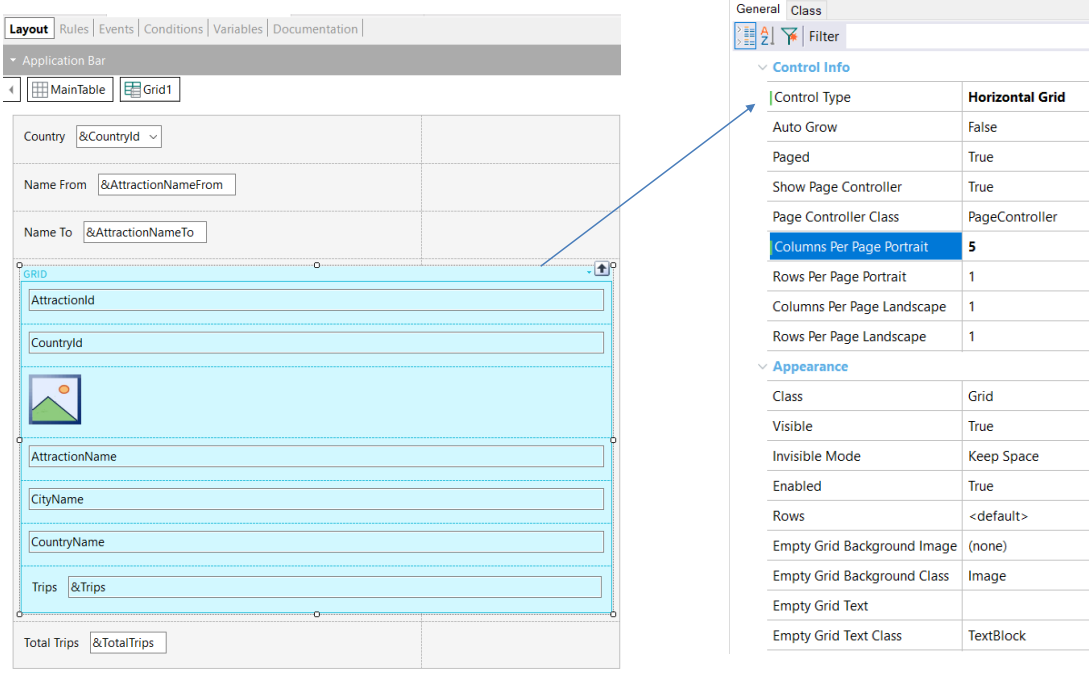
That is, the next row will bring back the same group of records. And this can also be defined as a fixed value (staggered or not), or depending on the width (minimum and maximum) that we want to give to each item.

That is, with the Smart Grid we will not achieve the behavior we need, which is to see all the attractions that fit in the available width of the screen and scroll to the right if there are still more that we can't see.

Horizontal Grid

We have already seen several options of the standard grid to display information.
We will now see another control called Horizontal Grid.

Horizontal Grid



In the Control type property, we can see several controls that we can use to display the information. We choose the Horizontal Grid value.

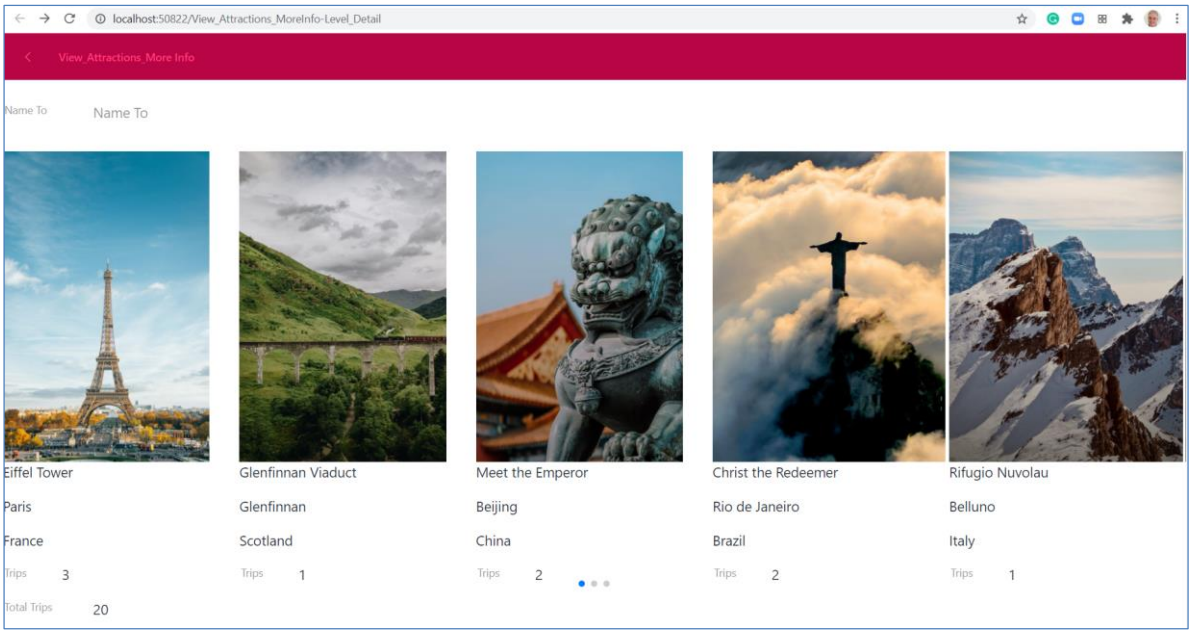
By assigning this value, a series of properties will be enabled that will allow us to configure how the grid will look like. For example, the Show Page Controller property determines whether the pager will be shown and the Page Controller Class property will allow us to define the class on which this pager is based.

This class is like a template to which we can change its properties, and all the controls based on the class will inherit those values.
However, GeneXus 17 offers a much more flexible, scalable, and powerful way to do this, which is through a Design System Object that we will study later on.

Going back to the properties that allow us to customize the horizontal grid, we can define how many columns we want to see and at the same time how many rows we want on each page. We are going to set the Columns Per Page Portrait property to value 5 and leave the Rows Per Page Portrait property to value 1 because we want to see only one row per page.

We run it.

Using an Horizontal Grid Control



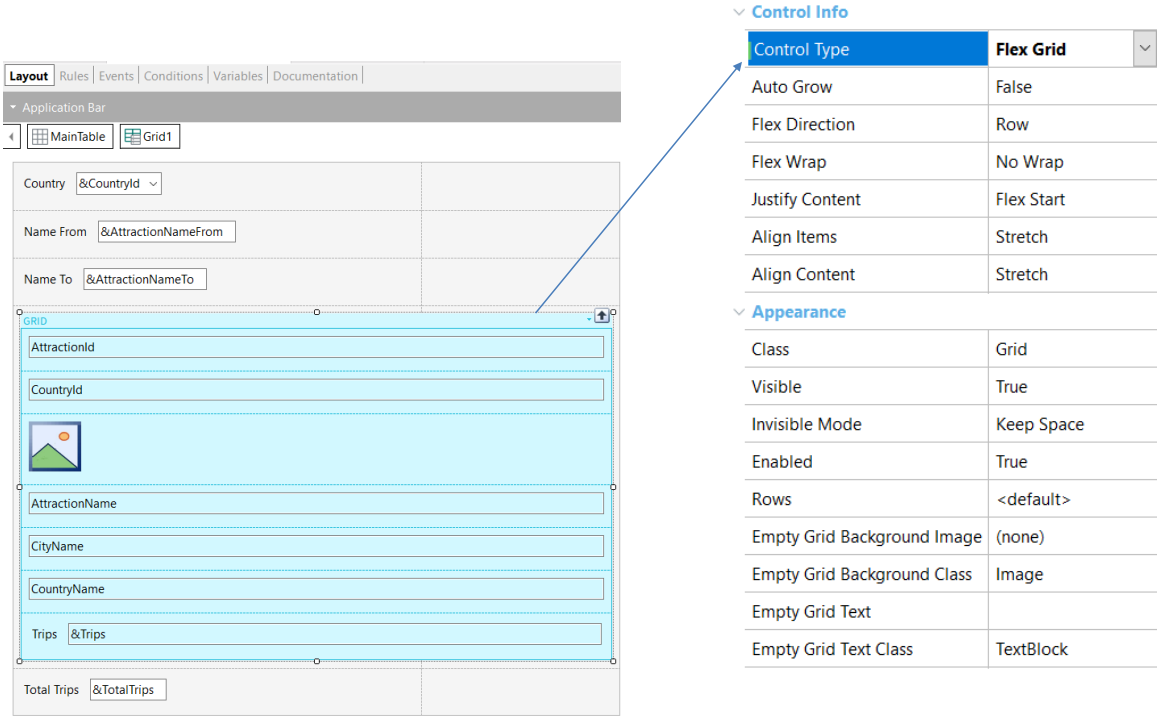
So, now we're getting very close to the way we wanted it to look. We can page to the right and we will see the attractions taking up the entire page, with 5 attractions at a time.

If we had also changed the Rows Per Page Portrait property, we would see several rows of attractions, but in this case we are not interested.

At the bottom of the page, note the paging control, which tells us which page we are on.

Flex Grid and Flex Control

We have already achieved an acceptable result for our application. Now we will see another grid that is even more flexible and a flexible container that allows you to organize content even if you don't know its size.



In this example, we first select Save As for the View_Attractions_MoreInfo panel and name it View_Attractions_MoreInfo 2.

To use the flexible grid, we go to the Control Type property of the grid and choose Flex Grid.
Flex Grid is a type of FreeStyle grid, where we can load as many elements as we want, dynamically.

The way the information is displayed depends on what we set in the five properties displayed on the screen: Flex Direction, Flex Wrap, Justify Content, Align Items, and Align Content.

Control Info

Control Type	Flex Grid
Auto Grow	False
Flex Direction	Row
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

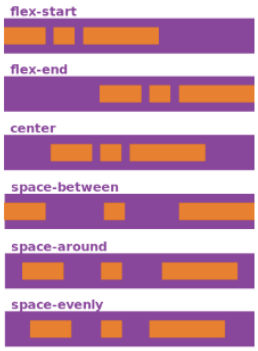
Flex Direction

Source: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Flex Wrap

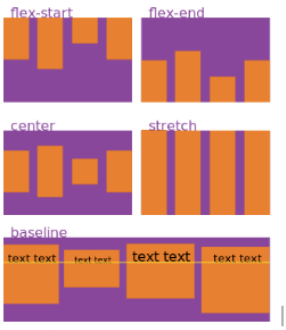
Source: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Justify Content



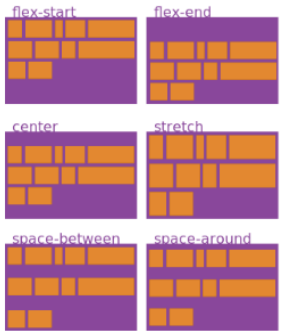
Source: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Align Items



Source: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Align Content



Source: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Flex Direction: Specifies the direction of the elements, which can be placed horizontally (Rows or Inverse Rows) or vertically (Column or Inverse Column).

Flex Wrap: Sets how elements will be arranged when they exceed the row length and not all of them can be displayed. The Wrap value sorts them on the next line.

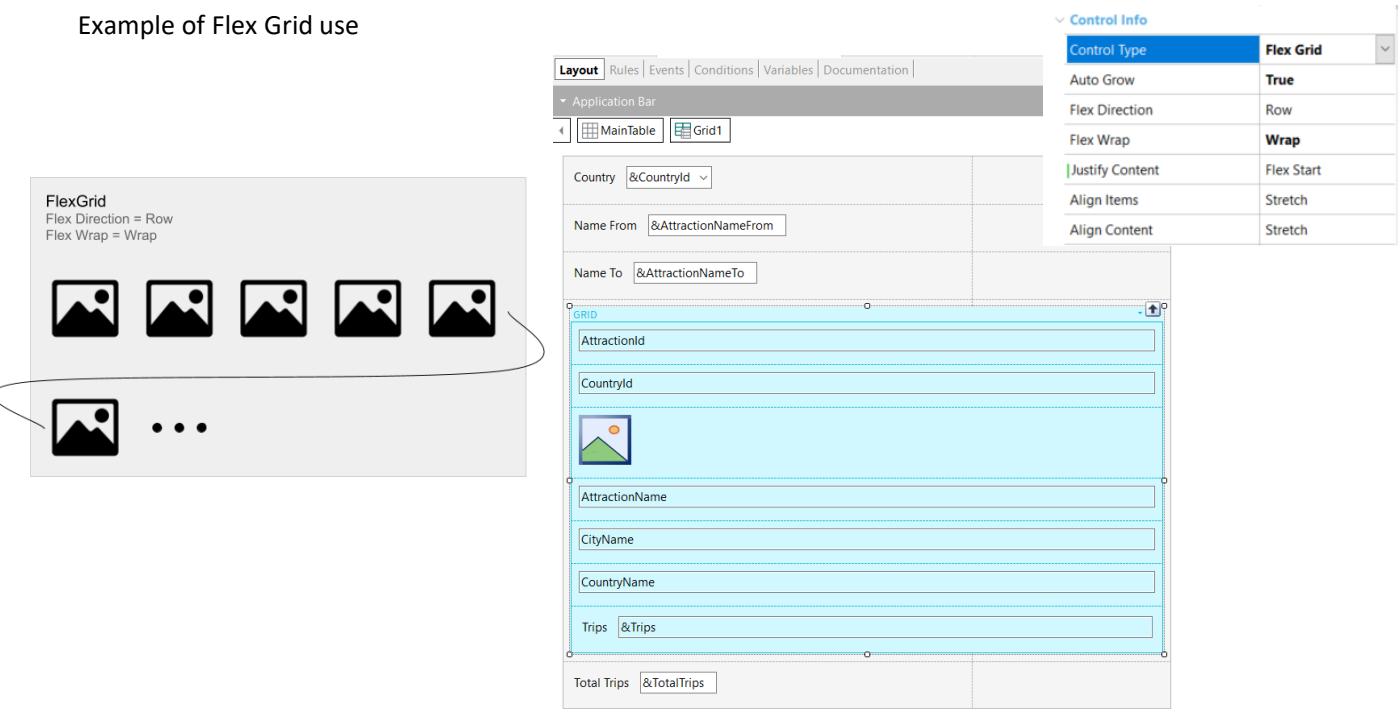
Justify Content: Defines the alignment of the elements in the rows (horizontal alignment if Flex Direction is specified as Row and vertical alignment if Flex Direction is specified as Column). If the alignment is horizontal, it allows setting how the content will be aligned. It can be left to right, right to left, centered, adjusting the spacing to take up the entire available width, or adding spaces between elements.

Align Items: Defines the alignment of the elements in the rows (vertical alignment if Flex Direction is specified as a row and horizontal alignment if Flex Direction is specified as a column).

Align Content: It allows aligning the row when there is additional space in the container.

These properties also apply to the Flex Control, which is a flexible container that we will see in a moment.

Example of Flex Grid use

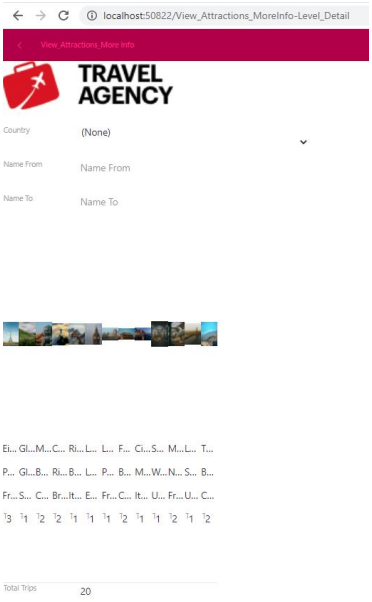


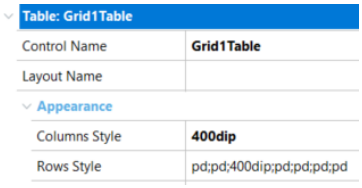
Suppose we want to see a photo gallery of the attractions, with the data of each attraction below the photo. We want the photos to flow horizontally and automatically move to a new row to avoid horizontal scrolling.

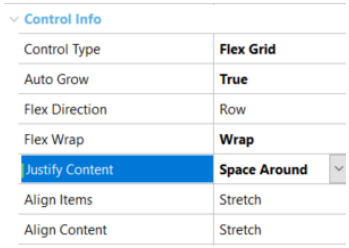
As we saw before, some photos are in landscape format and others in portrait format, so their dimensions are unknown at design time.

To achieve this, we will use the Flex Grid, with the Flex Direction property set to "Row" value and Flex Wrap set to "Wrap" value. We leave the rest of the default properties and execute.

Example of Flex Grid use

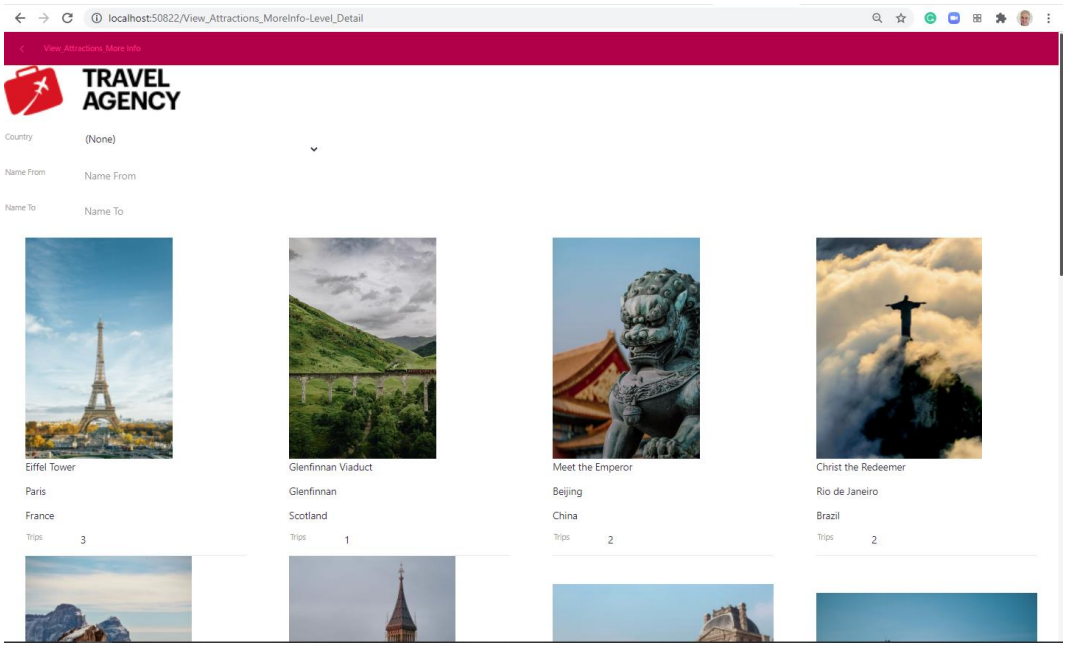






Note that there was an attempt to make a wrap, but it was not done because there was enough space.
We can also see that not enough space was left to see the attraction data clearly.
To fix this, let's give a value of 400 dips to the columns of the grid table—Grid1Table—so that it takes up more space and the Flex Grid must do the wrap.
We also change the value of the Justify Content property of the Flex grid to Space Around so that it adds spaces between the attractions and they are not stick to each other.
We run it.

Example of Flex Grid use



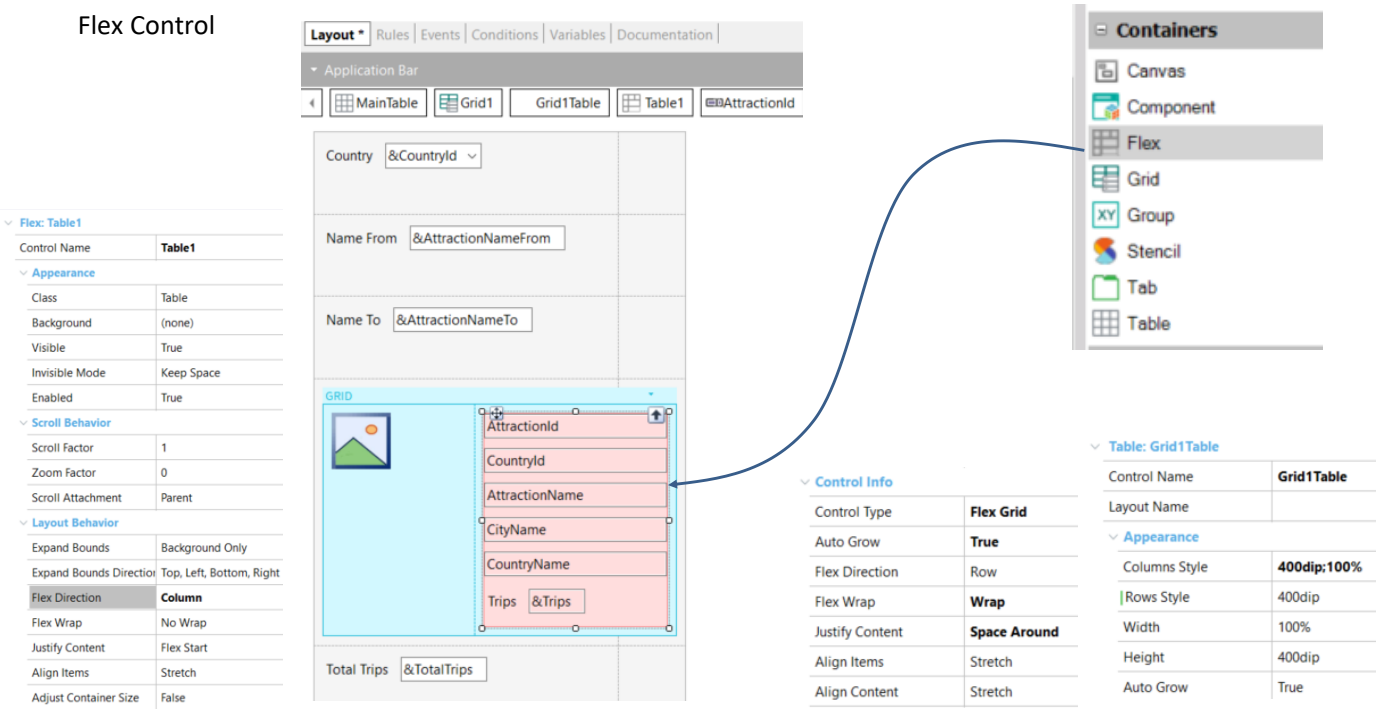
Now each column takes more space and the wrap is done, showing the attractions in a downwards direction automatically once the available space in the row is taken.

We also see that the separation between attractions is adequate and that the information is displayed correctly.

Suppose we wanted the attraction name, city, country, and total trips to be on the right side of the photo and not below it.

If we want to change the layout of each item in the grid, we must use a Flex Control.

Flex Control

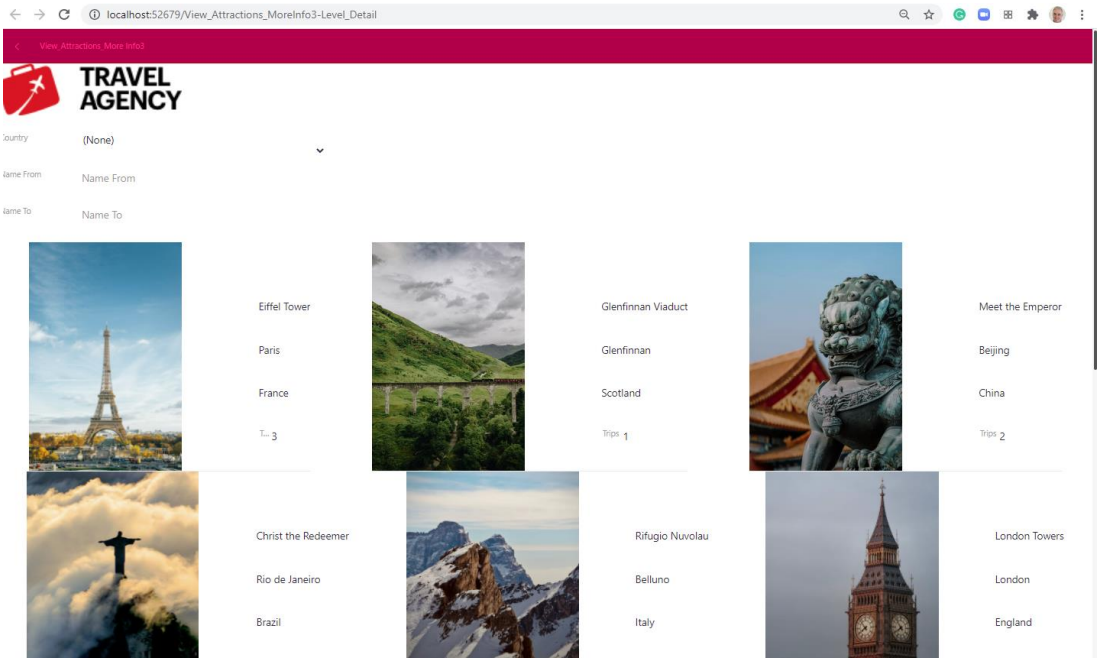


The Flex Control (also called Flex Table) is a container for controls, such as Tables or Responsive Tables. It aims to provide new ways to organize controls in the user interface, even if the sizes of the contained controls are unknown.

To continue working with the proposed example, we save the View_Attractions_MoreInfo2 panel with the name View_Attractions_MoreInfo3. And drag a Flex Control to the grid, to the right of the other controls.
It is called Table1, and if we go to its properties, we can see the same properties we had for the Flex Grid.

We set the Flex Direction property to Column and move all the attraction's data except the photo, to the Flex Table. The Flex Grid still has the same properties as in the previous example, and in the grid table we change the row value from pd to 400 dips.
We run it.

Example of Flex Control use



Now we see the attraction's data on the right. If we want, we can later adjust the values of the Flex Control properties to improve the way the data is displayed.

In this video, we saw how we can use different types of grids and configure them to view the information in different ways.

In the following videos, we will continue to improve the user experience of our application.

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