

Selección múltiple en un grid y comandos para procesar las líneas seleccionadas

GeneXus™

| AttractionId | AttractionName      | CountryId | CityId | CategoryId |
|--------------|---------------------|-----------|--------|------------|
| 1            | Louvre Museum       | 2         | 1      | 1          |
| 2            | The Great Wall      | 3         | 1      | 2          |
| 3            | Eiffel Tower        | 2         | 1      | 2          |
| 4            | Forbidden city      | 3         | 1      | 2          |
| 5            | Christ the Redemmer | 1         | 2      | 2          |

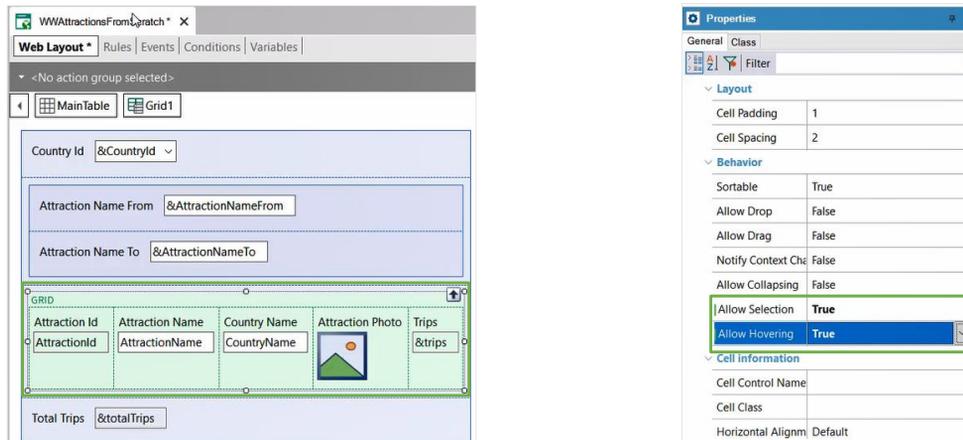
En diversas ocasiones necesitaremos trabajar con un conjunto de elementos previamente seleccionados, para realizar alguna acción luego con ellos.

En un grid, podemos visualizar muchos elementos, y es probable que queramos seleccionar solamente algunos.

A continuación, veremos cómo podemos realizar una selección múltiple de elementos de una forma sencilla, y cómo recorrer esos elementos para luego procesarlos.

Para este ejemplo utilizaremos la aplicación realizada para una agencia de viajes.

## Web Panel WWAttractionFromScratch



Tenemos en nuestra KB un Web panel creado el cual lista todas las atracciones ingresadas.

Supongamos que del listado necesitamos seleccionar solo una atracción para realizar alguna acción con ella luego, se hace necesario para esto, poder marcar esa fila del grid como seleccionada.

En una aplicación Web, si queremos poder marcar una fila del grid como seleccionada, vamos a las propiedades del grid y en la propiedad AllowSelection seleccionamos el valor True.

Al hacerlo se habilita la propiedad Allow Hovering, esta permitirá que al movernos por las filas se vayan marcando de un color, la dejamos en true, lo veremos ahora.

## WEB

The screenshot shows a web form with the following elements:

- A dropdown menu for "Country Id" with the value "&CountryId".
- Input fields for "Attraction Name From" (&AttractionNameFrom) and "Attraction Name To" (&AttractionNameTo).
- A grid with the following columns:
 

| Attraction Id            | Attraction Name | Country Name | Attraction Photo | Trips  |
|--------------------------|-----------------|--------------|------------------|--------|
| <input type="checkbox"/> | AttractionId    | CountryName  |                  | &trips |
- An input field for "Total Trips" with the value "&totalTrips".
- A button labeled "List of selected attractions".

```

Event 'List of selected attractions'
  For each line in Grid1
    if &selected
      &SelectedAttractionsIds.Add(AttractionId)
    endif
  Endfor
Endevent

```

Veámoslo en ejecución.

Ahora supongamos que necesitamos seleccionar una o más atracciones para luego imprimirlas en un listado PDF, por lo que necesitaremos poder realizar una selección múltiple de atracciones en dicho listado.

Poner la propiedad AllowSelection en True no nos sirve ya que esta no soporta selección múltiple.

Para una aplicación Web, si nuestro grid no tiene paginación, una de las opciones que tenemos para implementarlo es la siguiente: Primero que nada, creamos una variable del tipo boolean y la agregamos al grid en la primera columna. Esta la utilizaremos para poder seleccionar un registro.

Luego agregamos un botón, el cual invocará el procedimiento que listará las atracciones seleccionadas.

Para poder enviar estas atracciones al procedimiento, deberemos primero guardarlas en una colección.

Simplemente con guardar el ID de las atracciones seleccionadas ya es suficiente, por lo que creamos una variable del tipo ID (dominio del tipo numérico) y lo marcamos como colección.

Accedemos al evento asociado al botón, y necesitamos aquí recorrer cada una de las filas del grid, esto lo hacemos mediante la instrucción For Each Line, In, y el nombre del grid. Esto nos permitirá recuperar los valores de cada fila de la grilla.

Cabe aclarar que el comando For each line sirve para cualquier situación en la que en un evento de usuario deba recorrerse el grid. Como solamente nos interesan las atracciones seleccionadas, condicionamos la búsqueda con un If &selected (que es lo mismo a poner If &selected = true), y en caso que se cumpla esta condición agregamos el Id de la atracción a la variable colección creada.

SD

The screenshot shows the GeneXus IDE interface for a Smart Device (SD) application. The main window displays a grid control with the following columns: 'Attraction Id', 'AttractionId', an image placeholder, and 'AttractionName'. Below the grid is a button labeled 'List Selected Attractions'. The Properties panel for 'Grid: Grid1' is open, showing the following settings:

|                          |                  |
|--------------------------|------------------|
| Control Name             | Grid1            |
| Collection               |                  |
| Default Action           | <default>        |
| Selection Type           | Platform Default |
| Enable Multiple Selected | True             |
| Pull To Refresh          | False            |
| Inverse Loading          | False            |
| Default Selected Item    | (none)           |
| Show Selector            | Always           |

Below the Properties panel, an event handler is defined:

```

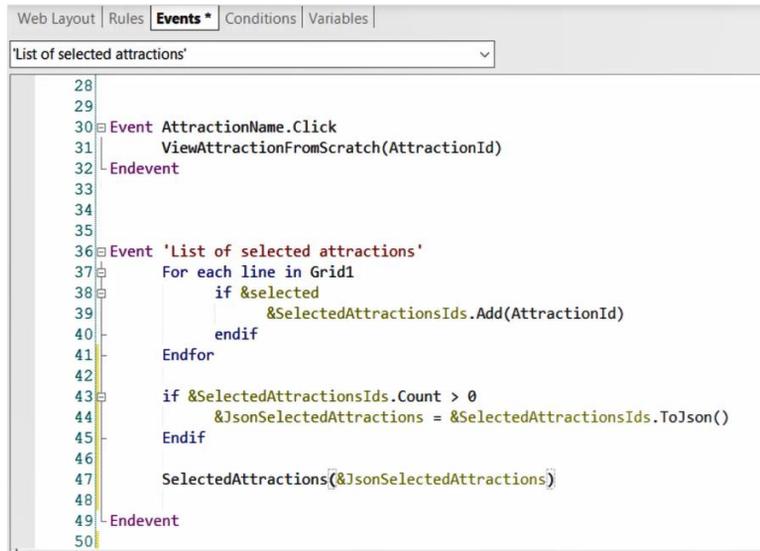
Event 'List Selected Attractions'
  Composite
    For each selected line in Grid1
      &SelectedAttractionsIds.Add(AttractionId)
    Endfor
  EndComposite
Endevent

```

Si se tratase de una aplicación para Smart Devices, es posible configurar la grilla para que permita selección múltiple mediante una propiedad del grid de nombre Enable Multiple Selection, deberemos ponerla en true. Y para que nos aparezca el checkbox que nos permite seleccionar la fila, dejamos con el valor Always la propiedad Show Selector. Estas propiedades nos ahorran tener que declarar una variable booleana, como lo hicimos recién en web.

Luego para recorrer las filas se debe utilizar el comando For Each selected line. El cual deberá estar agrupado dentro de un bloque Composite. Lo que hará este comando, es que si una de las instrucciones falla, el resto no se ejecutará.

## WEB



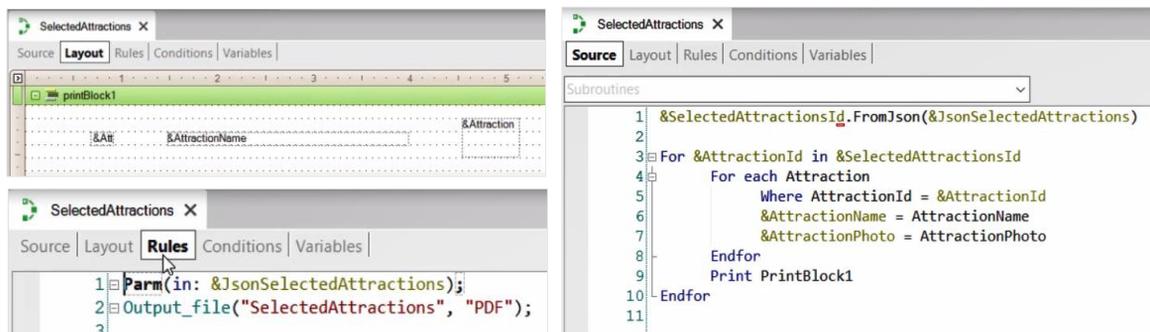
```
Web Layout | Rules | Events * | Conditions | Variables
'List of selected attractions'
28
29
30 Event AttractionName.Click
31   ViewAttractionFromScratch(AttractionId)
32 -Endevent
33
34
35
36 Event 'List of selected attractions'
37   For each line in Grid1
38     if &selected
39       &SelectedAttractionsIds.Add(AttractionId)
40     endif
41   Endfor
42
43   if &SelectedAttractionsIds.Count > 0
44     &JsonSelectedAttractions = &SelectedAttractionsIds.ToJson()
45   Endif
46
47   SelectedAttractions(&JsonSelectedAttractions)
48
49 -Endevent
50
```

Retomando con nuestro Web Panel, una vez recorridas todas las filas del grid, deberemos pasarle la colección de atracciones al procedimiento. No podemos pasar por parámetro directamente, sino que deberemos serializar su contenido, generar un archivo de texto de formato estructurado, como un JSON o un XML. Ambos son formatos utilizados para el intercambio de datos. En este caso utilizaremos JSON.

Para esto creamos una variable con este nombre, del tipo LongVarchar. Y el valor que tomará será el de la colección luego de aplicado el método ToJson.

Por último, invocamos al procedimiento que listara las atracciones seleccionadas, al que llamamos SelectedAttractions, y le pasamos por parámetro la variable que contiene el JSON que acabamos de obtener.

## Objeto Procedimiento



Veamos ahora como el procedimiento llamado, recibe y procesa toda esta información.

En el Layout ingresamos solamente estas tres variables para mostrar info de las atracciones seleccionadas.

En la sección rules del procedimiento, declaramos en la regla Parm una variable de entrada, del tipo LongVarChar, encargada de recibir la lista de atracciones que le estamos pasando.

En el Source, declaramos una variable del tipo colección, a la que le cargaremos el contenido de la variable que contiene el valor recibido por parámetro, aplicándole el método FromJson.

De esta manera, los Ids de las atracciones seleccionadas quedan guardadas en esta variable colección. Ahora deberemos recorrer la misma para acceder a ellos. Lo haremos mediante la siguiente instrucción.

A continuación, recorreremos la tabla Attraction, y por cada AttractionId obtenemos el nombre de la atracción y su foto, cargándolos en estas dos variables creadas para esto.

Luego que cerramos este for each, imprimimos el printblock que contiene las variables que mostrarán el Id, el nombre y la foto de nuestras atracciones.

## En ejecución

Country Id

Attraction Name From

Attraction Name To

| Attraction Name                                    | Country Name | Attraction Photo  | Trips |
|--|--------------|---|-------|
| <input checked="" type="checkbox"/> Louvre Museum  | France       |  | 0     |
| <input type="checkbox"/> The Great Wall            | China        |  | 0     |
| <input checked="" type="checkbox"/> Eiffel Tower   | France       |  | 0     |
| <input type="checkbox"/> Christ the Redeemer       | Brazil       |  | 0     |
| <input checked="" type="checkbox"/> Matisse Museum | France       |  | 0     |
| <input checked="" type="checkbox"/> Forbidden city | China        |  | 0     |

Total Trips 0

|   |                |   |
|---|----------------|---|
| 1 | Louvre Museum  |  |
| 3 | Eiffel Tower   |  |
| 6 | Matisse Museum |  |
| 7 | Forbidden city |  |

Probemos ahora esta funcionalidad que acabamos de programar.

Recordemos que, para este ejemplo, queremos seleccionar algunas atracciones y que se nos imprima un listado PDF con ellas.

Accedemos al web panel que nos muestra el listado de todas las atracciones ingresadas, y seleccionamos algunas de ellas. Luego presionamos el botón para ejecutar esta acción. Y vemos que nos aparece el listado con las atracciones que fueron seleccionadas.

## Resumen

- **Seleccionar una fila del Grid**
  - Propiedad **AllowSelection** en true
- **Seleccionar multiples filas del Grid**
  - Web
    - Agregar variable booleana al Grid
    - Se recorren las filas con For each line
  - SD
    - Propiedad **Enable Multiple Selection** en true
    - Se recorren las filas con **For each selected line**.
- **Pasaje de datos por parametro**
  - Serializar el contenido al formato Json mediante el metodo ToJson()
  - Al recibir los datos se deberan cargar en una variable colección mediante el método FromJson()

Resumiendo lo que vimos en este video.

Si queremos seleccionar solamente una fila de nuestro grid, simplemente cambiando a True la propiedad AllowSelection del grid y luego asignándole un color desde el tema es suficiente.

Si como acabamos de ver en el ejemplo planteado, necesitamos seleccionar más de una fila, no nos servirá esta propiedad, por lo que deberemos utilizar una variable booleana en el grid para guardar la selección de la fila.

Luego, para recorrer las filas de la grilla, debemos utilizar el comando For each line. Este comando itera sobre las filas cargadas en el grid, y por cada iteración obtendremos los valores de las columnas del grid por cada fila recorrida.

Para saber si la fila fue seleccionada o no, chequearemos el valor de la variable booleana.

Si estamos en una aplicación para Smart Devices, el grid sí cuenta con una propiedad para esta funcionalidad, la propiedad Enable Multiple Selección.

Luego para recorrer estas filas se utiliza la propiedad show selector junto al comando For each selected line.

Luego para pasar la colección de atracciones al procedimiento, se deberá serializar el contenido. Y una vez recibido por parámetro desde el otro objeto, para poder manipular los registros, creamos una variable colección y la cargamos con estos datos, mediante el método FromJson.

Para mas información los invitamos a indagar en nuestra Wiki.

*GeneXus*<sup>™</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)