Multiple selection in grid and commands to process the selected lines

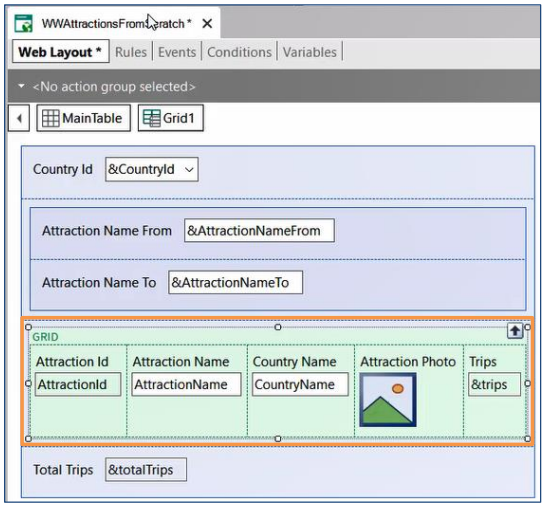
GeneXus™

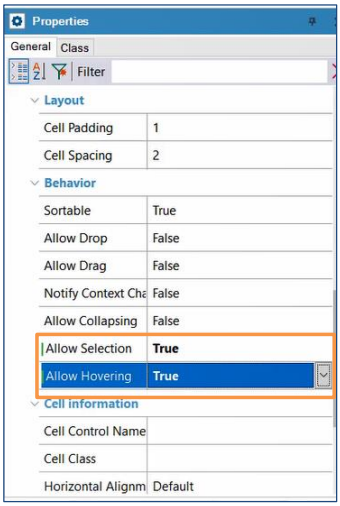
AttractionId	AttractionName	CountryId	CityId	CategoryId
1	Louvre Museum	2	1	1
2	The Great Wall	3	1	2
3	Eiffel Tower	2	1	2
4	Forbidden city	3	1	2
5	Christ the Redemmer	1	2	2

Em várias ocasiões precisaremos trabalhar com um conjunto de elementos previamente selecionados, para realizar alguma ação com eles posteriormente. Em um grid, podemos visualizar muitos elementos e é provável que queiramos selecionar apenas alguns.
A seguir, veremos como podemos fazer uma seleção múltipla de elementos de uma forma simples e como percorrer esses elementos para depois processá-los.

Para este exemplo, utilizaremos a aplicação feita para uma agência de viagens.

Web Panel WWAttractionFromScratch



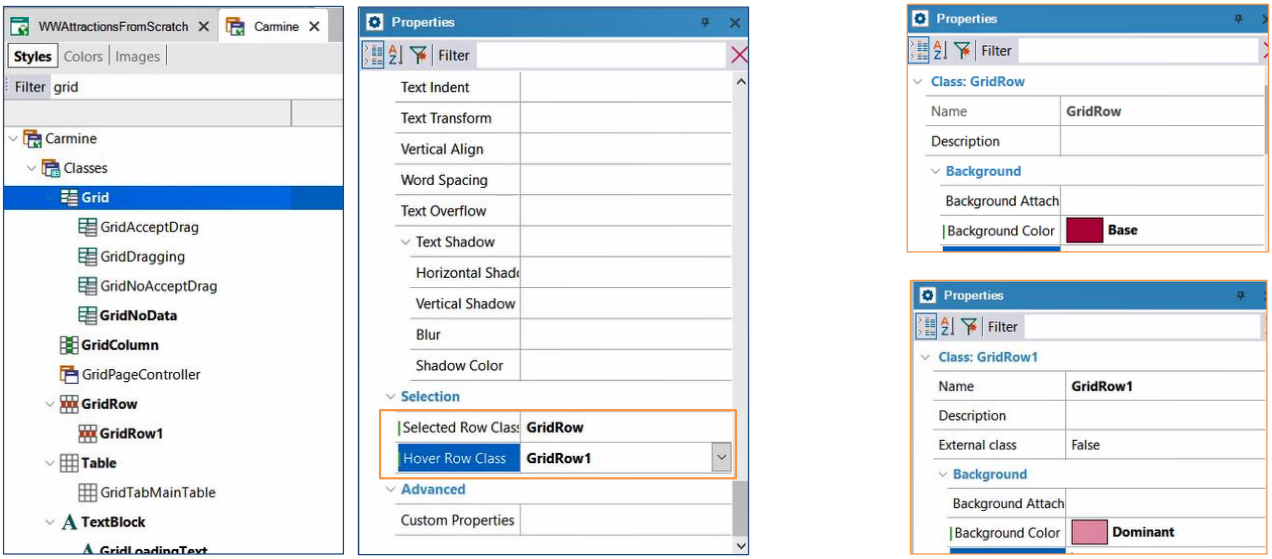


Temos em nossa KB um Web panel criado, que lista todas as atrações inseridas. Suponhamos que da lista precisamos selecionar apenas uma atração para realizar alguma ação com ela depois, é necessário para isto, ser possível marcar essa linha do grid como selecionada.

Em uma aplicação Web, se quisermos marcar uma linha do grid como selecionada, vamos às propriedades do grid e na propriedade Allow Selection selecionamos o valor True.

Ao fazer isso, é habilitada a propriedade Allow Hovering, esta permitirá que ao nos mover pelas linhas vão sendo marcadas com uma cor, a deixamos também em true.

Modificar propriedades no tema do Web panel



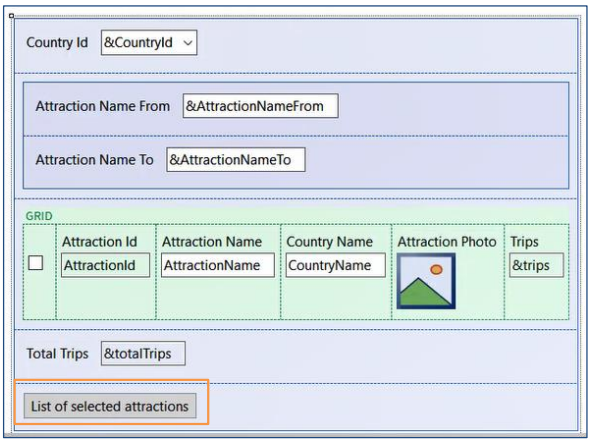
Para visualizá-lo em tela, resta configurar uma cor para cada funcionalidade. Fazemos isto inserindo o tema que tenha associado o web panel que contém o grid. Então, na classe Grid, temos a propriedade Selected Row Class, onde devemos atribuir uma classe que conterá a cor para quando selecionamos uma linha, escolhemos a classe GridRow. E a propriedade Hover Row Class, que atribuiremos a classe GridRow1, criada como descendente de GridRow, que conterá a cor que será atribuída à linha quando passemos sobre ela.

Vejamos em execução.

Agora, suponhamos que precisamos selecionar uma ou mais atrações para depois imprimi-las em uma lista PDF, portanto, precisaremos ser capazes de fazer uma seleção múltipla de atrações nesta lista.

Colocar a propriedade Allow Selection em True não nos serve, pois esta não suporta seleção múltipla.

WEB




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

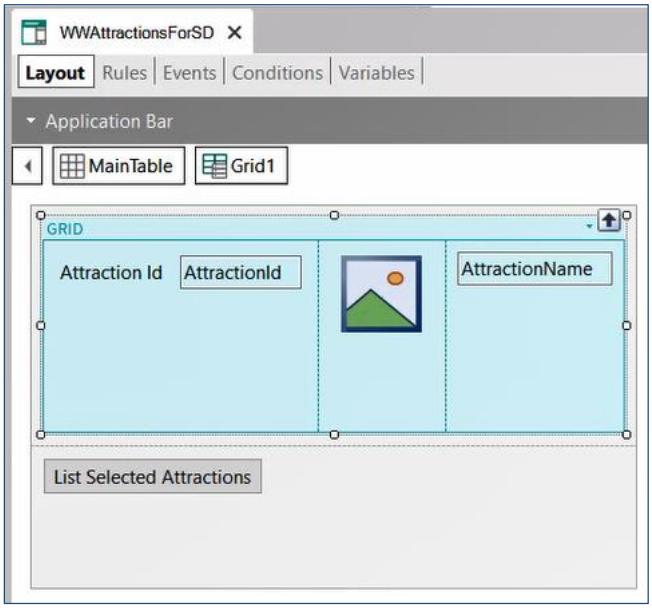
Para uma aplicação Web, se nosso grid não tiver paginação, uma das opções que temos para implementá-la é a seguinte:
Em primeiro lugar, criamos uma variável do tipo boolean e a adicionamos ao grid na primeira coluna. A utilizaremos para poder selecionar um registro.

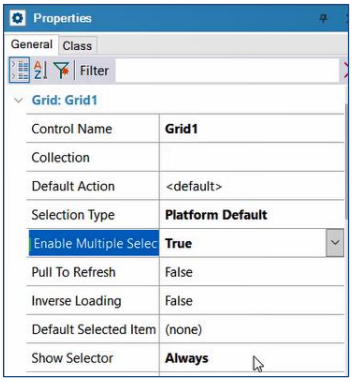
Em seguida, adicionamos um botão, que invocará o procedimento que irá listar as atrações selecionadas.
Para ser possível enviar estas atrações para o procedimento, devemos primeiro salvá-las em uma coleção.
Basta salvar o ID das atrações selecionadas, por isso criamos uma variável do tipo ID (domínio do tipo numérico) e a marcamos como coleção.

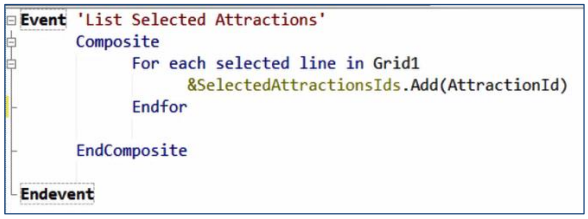
Acessamos o evento associado ao botão, e precisamos aqui percorrer cada uma das linhas do grid, fazemos isto por meio da instrução For Each Line, In, e o nome do grid. Isto nos permitirá recuperar os valores de cada linha do grid.

Vale esclarecer que o comando For each line serve para qualquer situação na qual em um evento de usuário deva ser percorrido o grid.
Como somente nos interessam as atrações selecionadas, condicionamos a pesquisa com um If &selected (que é o mesmo que colocar If &selected = true), e no caso que se cumpra esta condição adicionamos o Id da atração à variável coleção criada.

SD



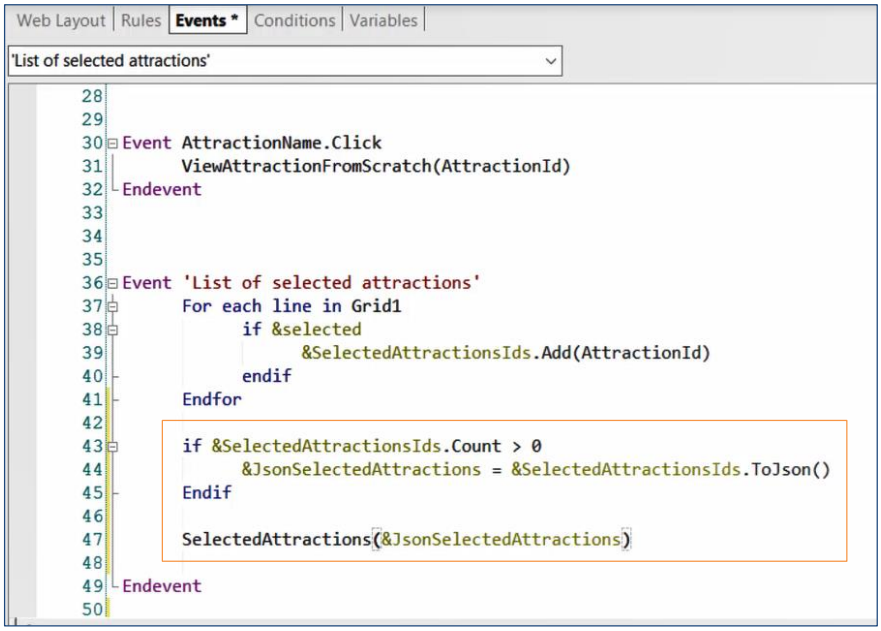




Se fosse uma aplicação para Smart Devices, é possível configurar o grid para que permita seleção múltipla através de uma propriedade do grid chamada Enable Multiple Selection, deveremos colocá-la como true. E para que nos apareça o checkbox que nos permite selecionar a linha, deixamos com o valor Always a propriedade Show Selector. Estas propriedades nos poupam de ter que declarar uma variável booleana, como acabamos de fazer em web.

Então, para percorrer as linhas, deve-se utilizar o comando For Each selected line. Que deverá estar agrupado dentro de um bloco Composite. O que este comando fará é que, se uma das instruções falhar, o resto não será executado.

WEB

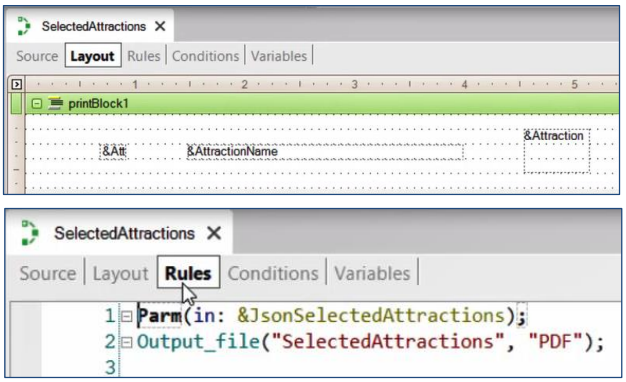


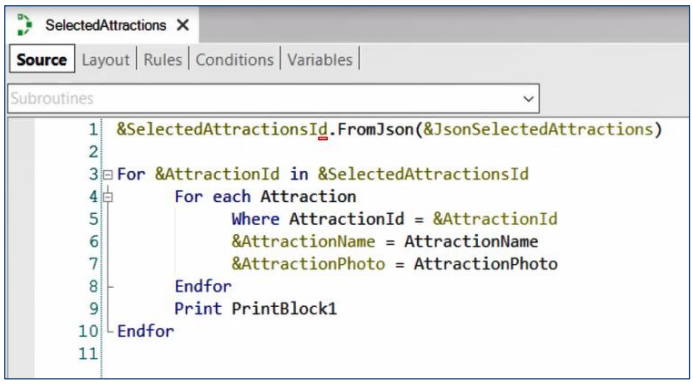
Voltando ao nosso Web Panel, uma vez percorridas todas as linhas do grid, deveremos passar a coleção de atrações para o procedimento.
Não podemos passar por parâmetro diretamente, mas deveremos serializar seu conteúdo, gerar um arquivo de texto em formato estruturado, como um JSON ou um XML. Ambos são formatos utilizados para troca de dados. Neste caso, utilizaremos JSON.

Para isto criamos uma variável com este nome, do tipo LongVarchar. E o valor que assumirá será o da coleção após a aplicação do método ToJson.

Por último, invocamos o procedimento que listará as atrações selecionadas, que chamamos de SelectedAttractions, e passamos por parâmetro a variável que contém o JSON que acabamos de obter.

Objeto Procedimento





Vejamos agora como o procedimento chamado recebe e processa todas estas informações.

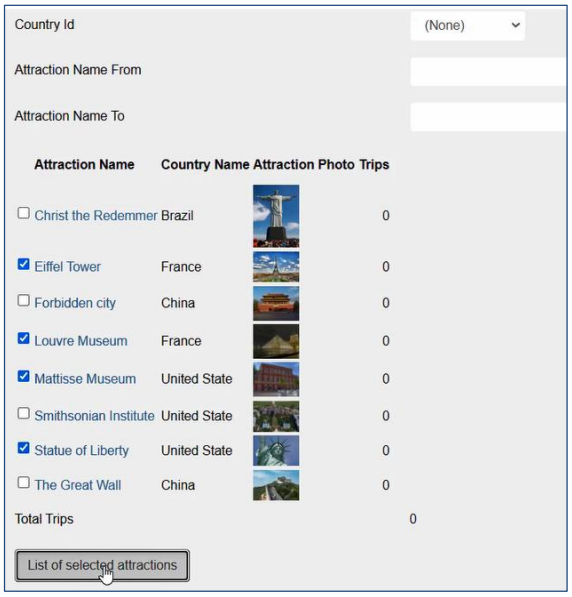
No Layout, inserimos somente estas três variáveis para mostrar informações das atrações selecionadas.
Na seção rules do procedimento, declaramos na regra Parm uma variável de entrada, do tipo LongVarChar, encarregada de receber a lista de atrações que estamos passando para ela.

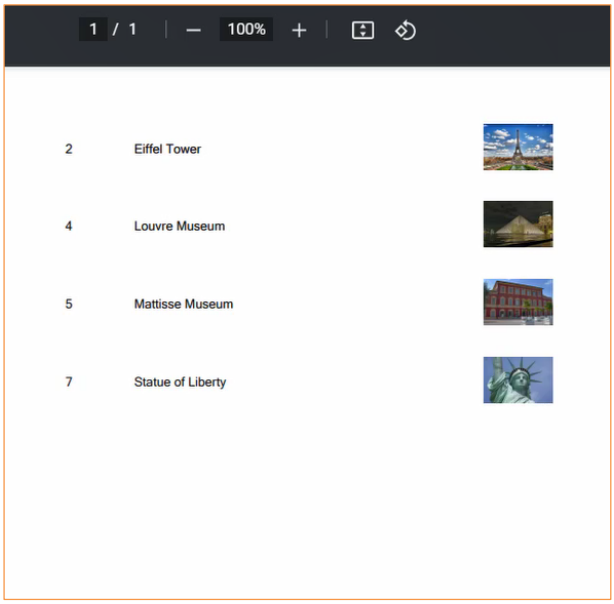
No Source, declaramos uma variável do tipo coleção, para a qual carregaremos o conteúdo da variável que contém o valor recebido por parâmetro, aplicando o método FromJson.

Desta forma, os Ids das atrações selecionadas ficam salvos nesta variável coleção. Agora deveremos percorrer a mesma para acessá-los. Faremos isso por meio da seguinte instrução.

A seguir, percorremos a tabela Attraction, e para cada AttractionId obtemos o nome da atração e sua foto, carregando-os nestas duas variáveis criadas para isto. Depois de fecharmos este for each, imprimimos o printblock que contém as variáveis que vão mostrar o Id, o nome e a foto de nossas atrações.

Em execução





Vamos testar agora esta funcionalidade que acabamos de programar.

Lembremos que, para este exemplo, queremos selecionar algumas atrações e que seja impressa uma lista PDF com elas.
Acessamos o web panel que nos mostra a lista de todas as atrações inseridas e selecionamos algumas delas. Em seguida, pressionamos o botão para executar esta ação. E vemos que aparece a lista com as atrações que foram selecionadas.

Resumo

- **Selecionar uma linha do Grid**
    - **Propriedade AllowSelection** em true
- **Selecionar múltiplas linhas do Grid**
    - Web
        - **Adicionar variável booleana ao Grid**
        - São percorridas as linhas com For each line
    - SD
        - **Propriedade Enable Multiple Selection** em true
        - São percorridas as linhas com **For each selected line**.
- **Passagem de dados por parâmetro**
    - Serializar o conteúdo para o formato Json usando o método ToJson()
    - Ao receber os dados, deverão ser carregados em uma variável coleção usando o método FromJson()

Resumindo o que vimos neste vídeo.

Se quisermos selecionar apenas uma linha de nosso grid, simplesmente alterando para True a propriedade Allow Selection do grid e, em seguida, atribuindo a ela uma cor a partir do tema é suficiente.

Se, como acabamos de ver no exemplo apresentado, precisarmos selecionar mais de uma linha, não nos servirá esta propriedade, então deveremos utilizar uma variável booleana no grid para salvar a seleção da linha.

Então, para percorrer as linhas do grid, devemos utilizar o comando For each line. Este comando itera sobre as linhas carregadas no grid e, para cada iteração, obteremos os valores das colunas do grid para cada linha percorrida.
Para saber se a linha foi selecionada ou não, verificaremos o valor da variável booleana.

Se estamos em uma aplicação para Smart Devices, o grid conta com uma propriedade para esta funcionalidade, a propriedade Enable Multiple Selection.

Em seguida, para percorrer estas linhas, é utilizada a propriedade show selector junto com o comando For each selected line.

Então, para passar a coleção de atrações para o procedimento, deverá ser serializado o conteúdo. E uma vez recebido por parâmetro a partir do outro objeto, para ser possível manipular os registros, criamos uma variável coleção e a carregamos com estes dados, usando o método FromJson.

Para mais informações, convidamos a consultar em nossa Wiki.

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