

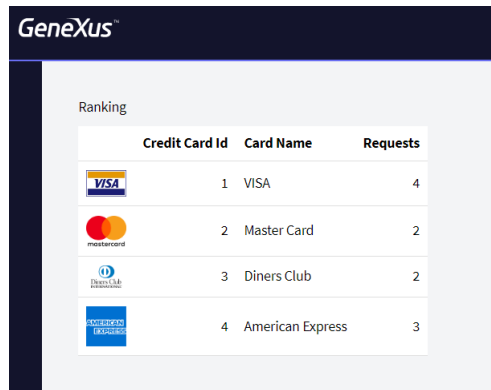
Carga de um SDT coleção no grid de um Web Panel

GeneXus™





Muitas vezes precisamos exibir o conteúdo de uma coleção de dados, e um web panel é uma opção.

Veremos um exemplo de carga de um SDT coleção no grid de um Web panel, para depois poder realizar alguma ação sobre suas linhas.

Example: Credit cards Ranking

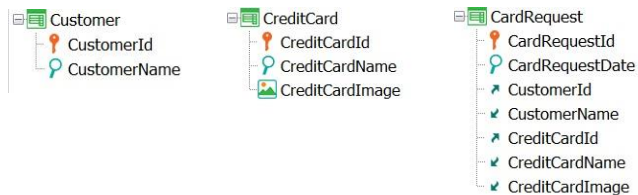


The screenshot shows a web application interface with a dark blue header containing the 'GeneXus' logo. Below the header, the word 'Ranking' is displayed. A table with three columns: 'Credit Card Id', 'Card Name', and 'Requests' is shown. The table contains four rows of data, each with a credit card logo in the first column, a rank and card name in the second, and the number of requests in the third.

Credit Card Id	Card Name	Requests
	1 VISA	4
	2 Master Card	2
	3 Diners Club	2
	4 American Express	3

Suponhamos então que a Agência de viagens oferece a possibilidade de registrar as solicitações dos clientes para a emissão de diferentes cartões de crédito com os quais trabalha a agência. Assim, queremos implementar um ranking de cartões de crédito, de acordo com a quantidade de solicitações registradas no ano 2020. Por se tratar de um ranking, a informação deverá ser apresentada ordenada da maior para a menor pela referida quantidade.

Example: Credit cards Ranking



Name	Type	Description	Is Collection
SDTCardRequests		SDTCard Requests	<input checked="" type="checkbox"/>
SDTCardRequestsItem			<input type="checkbox"/>
• CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
• CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
• CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
• CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

Consideremos então o desenho de transações que estamos vendo e o tipo de dado estruturado que permitirá carregar a coleção de todos os cartões, cada um com sua correspondente quantidade de solicitações registradas no ano 2020.

Example: Credit cards Ranking


```
SDTCardRequests from CreditCard
{
  SDTCardRequestsItem
  {
    CreditCardId
    CreditCardImage
    CreditCardName
    CreditCardRequests = count(CardRequestDate,CardRequestDate.Year())=2020)
  }
}
```

Name	Type	Description	Is Collection
SDTCardRequests		SDTCard Requests	<input checked="" type="checkbox"/>
SDTCardRequestsItem			
• CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
• CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
• CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
• CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

Vamos ao GeneXus:

como realizamos a carga do SDT? Com um Data provider
Este Data provider carregará cada cartão de crédito com sua
correspondente quantidade de solicitações registradas no ano 2020, daí a
condição de cálculo declarada na fórmula.

Example: Credit cards Ranking

Ranking			
GRID			
	Credit Card Id	Card Name	Requests
	&CardRequests.item(0).CreditCardId	&CardRequests.item(0).CreditCardName	&CardRequests.item(0).CreditCardRequests

```

Event Start
  &CardRequests = DPRequests()
  &CardRequests.Sort("[CreditCardRequests]")
Endevent

```

Name	Type	Is Collection
Variables		
Standard Variables		
• CardRequests	SDTCardRequests	<input type="checkbox"/>

Então, a partir do evento Start de um web panel, recebemos a saída deste Data provider, e fazemos isso na variável &CardRequests que foi definida baseada no tipo de dado SDTCardRequests.

Bem, mas para que seja realmente um ranking, devemos ordená-la da maior para a menor pela quantidade de solicitações registradas, e para isso utilizamos o método Sort.

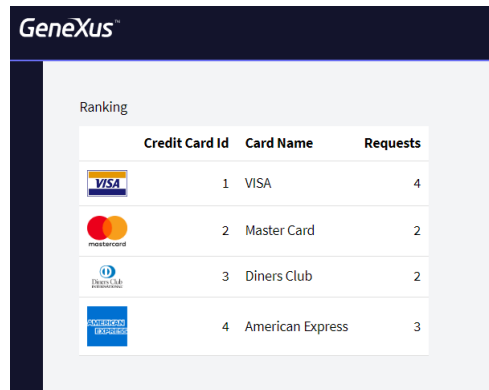
Está faltando mostrar efetivamente este conteúdo em um grid.

Para isto, a partir do form do web panel, vamos inserir a variável &CardRequests. Como se trata de uma variável estruturada e coleção, automaticamente GeneXus associará a um grid.





Mas é uma variável, e já sabemos que as variáveis em um web panel são dados de entrada. Portanto, para que não possa ser modificado seu conteúdo, vamos indicar que é Read only.

Observemos que também podemos escolher se queremos que seu conteúdo seja exibido em um Free style grid. No caso de não marcar esta opção, será exibida em um grid comum.

Em execução:



The screenshot shows the GeneXus application interface. At the top, the GeneXus logo is visible. Below it, the word "Ranking" is displayed. A table with three columns: "Credit Card Id", "Card Name", and "Requests" is shown. The table contains four rows of data, each with a credit card logo in the first column, a rank and card name in the second, and a number of requests in the third.

Credit Card Id	Card Name	Requests
	1 VISA	4
	2 Master Card	2
	3 Diners Club	2
	4 American Express	3

Pensemos agora no seguinte:

Existe alguma diferença entre este grid carregado a partir de uma variável estruturada coleção e outro grid carregado com variáveis da maneira habitual? Não, em qualquer caso se trata de um grid sem tabela base.

Vamos ver em execução. Pressionemos F5:

Bem, já implementamos nosso ranking. Agora queremos resolver o seguinte:

Precisamos poder selecionar um destes cartões de crédito, e que seja exibida uma lista pdf com as solicitações registradas para poder obtê-lo.

Voltemos ao GeneXus.

Já definimos a lista CardRequestList

```
▢ Parm(in:CreditCardId, in:&CreditCardName);  
▢ output_file("List", "pdf");
```

```
▢ printblock1  
.....  
.....&CreditCardName.....  
.....Card requests.....  
.....  
▢ printblock2  
.....CardRe CustomerName.....  
.....
```

```
1 print printblock1  
2  
3 For each CardRequest  
4   print printblock2  
5 endfor
```

Para facilitar, já definimos o procedimento CardRequestList que recebe em sua regra Parm o identificador e o nome do cartão de crédito, e exibe as solicitações registradas.

Propriedades Allow Selection e Allow Hovering

The screenshot shows a web panel with a 'Ranking' header and a 'GRID' below it. The grid has three columns: 'Credit Card Id', 'Card Name', and 'Requests'. The data row contains the following values: '&CardRequests.item(0).CreditCardId', '&CardRequests.item(0).CreditCardName', and '&CardRequests.item(0).CreditCardRequests'. A blue arrow points from the grid to a 'Behavior' table.

Behavior	
Sortable	True
Allow Drop	False
Allow Drag	False
Notify Context Chan	False
Allow Collapsing	False
Allow Selection	True
Allow Hovering	True

Bem, voltamos então ao nosso web panel.

Precisamos poder selecionar uma linha do grid. Então, vamos ver as propriedades do grid e vamos habilitar a propriedade Allow Selection com o valor True.

Desta forma, também é habilitada a propriedade Allow Hovering, que permite ir realçando com cor à medida que passamos pelas linhas. Vale mencionar que as cores padrão podem ser modificadas através do Theme aplicado, editando as classes atribuídas às propriedades Selected Row Class e Hover Row Class no nível da classe do Grid;

Bom. Mas também precisamos de um evento que capture a linha selecionada e permita realizar uma ação. Esse evento é o evento OnlineActivate associado ao grid.

Então, vamos ao setor dos eventos e escrevemos Event e o nome do grid, que neste caso é Grid1, ponto, e selecionamos o evento onlineActivate

O que queremos fazer é chamar a lista que vimos anteriormente, e passar por parâmetro o Id e o Nome do cartão selecionado na linha.

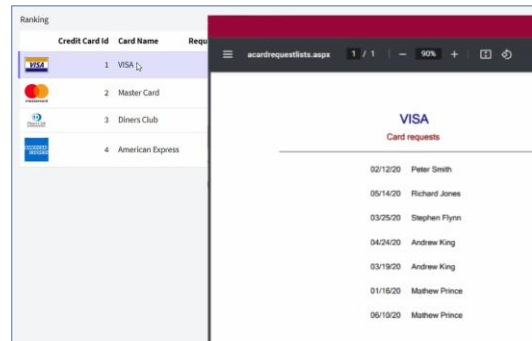
Propriedade CurrentItem

```

Event Start
  &CardRequests = DPRequests()
  &CardRequests.Sort("[CreditCardRequests]")
EndEvent

Event Grid1.OnLineActivate
  &CreditCardId = &CardRequests.CurrentItem.CreditCardId
  &CreditCardName = &CardRequests.CurrentItem.CreditCardName
  CardRequestLists.Popup(&CreditCardId, &CreditCardName)
EndEvent

```



Como fazemos?

Como o grid mostra a carga de uma coleção, devemos nos posicionar no item selecionado da coleção para poder obter então o id e o nome. Para isso, vamos utilizar a propriedade CurrentItem.

Então, vamos definir a variável &CreditCardId... vemos que ficou baseada automaticamente no atributo de mesmo nome, e vamos atribuir a variável coleção &CardRequests ponto, CurrentItem, ponto, CreditCardId, para obter desta forma o Id do cartão selecionado na linha.

Faremos o mesmo para obter o Nome.

Definimos a variável &CreditCardName, verificamos que ficou baseada no atributo de mesmo nome, e definimos a atribuição correspondente.

Finalmente, chamaremos a lista definida para que se abra como uma janela pop-up, e passamos os parâmetros necessários: a variável &CreditCardId, e a variável &CreditCardName.

Vamos ver em execução. Pressionemos F5:

Collection SDT in the grid of a Panel object

- The collection is received in the Refresh event.
- The variable is inserted in the form and GeneXus will automatically create the grid with the selected fields.
- To perform any action on a selected line, we must define an event and use the CurrentItem property
- This event should be defined as the default action of the grid (Default Action property).

Para finalizar, vale mencionar que se estivesse trabalhando com um objeto do tipo Panel, o processo é similar:

No evento Refresh, é recebida a coleção em uma variável.

É inserida esta variável no form e GeneXus criará automaticamente o grid associado com os campos selecionados.

No caso de querer também realizar alguma ação sobre uma linha selecionada, deverá ser definido um evento e utilizar a propriedade CurrentItem

Finalmente, deverá ser definido este evento como a ação padrão do grid, utilizando para isto a propriedade Default Action.

GeneXus[™]

training.genexus.com
wiki.genexus.com