

Duas formas de devolver uma coleção com um Data Provider

GeneXus™ 16

Para que um Data Provider carregue uma coleção de elementos:

The screenshot illustrates the configuration of a Data Provider to return a collection of elements. It is divided into three main sections:

- SDT Definition Table:**

Name	Type	Descript...	Is Collection
SDTCountries		SDTCou...	<input checked="" type="checkbox"/>
SDTCountriesItem			<input type="checkbox"/>
Id	Id	Id	<input type="checkbox"/>
Name	Name	Name	<input type="checkbox"/>
CountryAttractionsQuantity	Numeric(4,0)	Country ...	<input type="checkbox"/>
- Source Code:**

```

1 SDTCountries
2 {
3   SDTCountriesItem
4   {
5     Id = /*Id value*/
6     Name = /*Name value*/
7     CountryAttractionsQuantity = /*Coun
8   }
9 }

```
- Properties Window:**

Data Provider: RankingCountriesWithAttractionsQty

Name	RankingCountriesWithAttractionsQty
Description	Ranking Countries With Attractions Qty
Expose as Web Service	False
Module/Folder	Root Module
Qualified Name	RankingCountriesWithAttractionsQty
Object Visibility	Public
Output	
Infer Structure	No
Output	SDTCountries
Collection	False

Os Data Providers são objetos versáteis que, com uma linguagem simples e declarativa, nos permitem facilmente carregar estruturas de dados, tanto de itens simples como coleções de itens.

Como vimos no vídeo anterior, quando queríamos que um Data Provider carregasse uma coleção de elementos, o que fazíamos era definir uma variável baseada no tipo de dados estruturado (SDT) definido como coleção e depois arrastávamos essa SDT para o source do Data Provider. Fazendo isso, a propriedade output do mesmo se configurava automaticamente para carregar esse tipo de dados.

Agora faremos o mesmo, porém quando o SDT não é uma coleção

Objetivo: ranking de países

2	France	4
5	Egypt	3
4	United States	2
3	China	2
6	Uruguay	1
1	Brazil	1

1) Criamos o SDTCountry que não é uma coleção



Agora veremos que um Data Provider também possibilita devolver uma coleção de elementos, mesmo quando o tipo de dados estruturado **não é uma coleção**.

A agência de viagens havia solicitado um relatório que mostre um ranking dos países com mais atrações turísticas. Nosso exemplo implica em criar uma coleção de países, onde cada elemento é de um tipo de dados estruturado que armazena os dados de um só país.

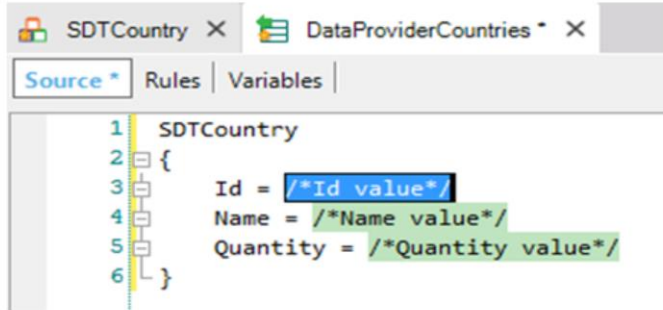
Para começar, vamos criar um tipo de dados estruturado (SDT) de nome SDTCountry, com os itens Id, Name e Quantity, no qual guardaremos a quantidade de atrações turísticas do país. Não definiremos esse SDT como coleção. Dessa forma, só poderá armazenar os dados de um único país.

2) Definimos uma variável baseada em SDTCountry e a marcamos como coleção

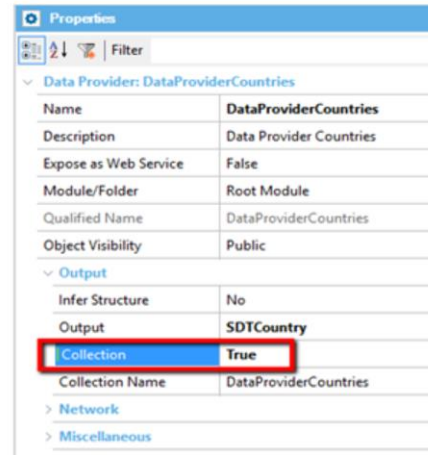


Depois, no objeto procedimento que implementa o ranking, definimos uma variável CountriesCollection do tipo SDTCountry e a marcamos como coleção.

3) Criamos o data provider e arrastamos o SDTCountry para o source



4) Configuramos a propriedade Collection como True



Para obter os dados de todos os países para construir o ranking, criamos um objeto Data Provider de nome DataProviderCountries. Depois arrastamos o SDTCountry para o Source do Data Provider.

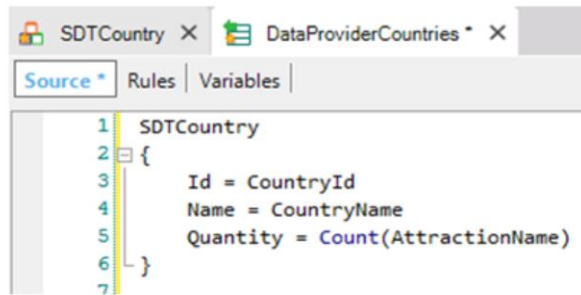
Vemos que foi criado automaticamente um código que reflete a estrutura do SDTCountry. Porém, não encontramos nenhum grupo repetitivo, já que não existe um grupo definido por chaves que esteja dentro de outro. Dessa forma, com a estrutura definida no Source, o Data Provider será capaz de carregar os dados de um só país. Porém, nós queremos carregar os dados de todos os países...

Felizmente o Data Provider permite resolver este problema, construindo a coleção por nós. Para isso, vamos às propriedades do Data Provider e, dentro do grupo Output, setamos a propriedade **Collection** como True.

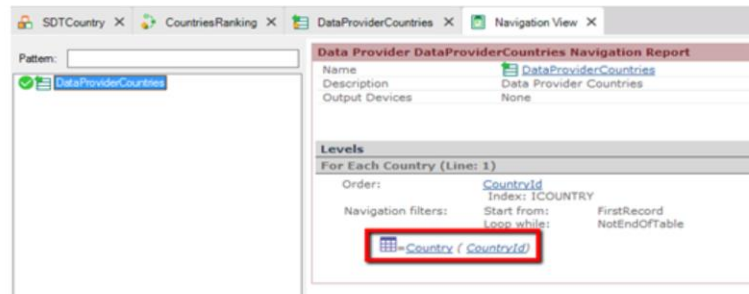
Notemos que, quando arrastamos o SDTCountry para o Source do Data Provider, a propriedade Output se configurou automaticamente com o nome do SDT, que já sabemos ser capaz de armazenar um único elemento.

Porém, ao definir a propriedade Collection como True, estamos indicando ao Data Provider que queremos que nos devolva uma coleção de elementos do tipo de SDTCountry. Também observamos que apareceu a propriedade Collection Name e que um nome foi definido automaticamente para a coleção.

5) Completamos a definição do data provider...



..e conferimos a lista de navegação

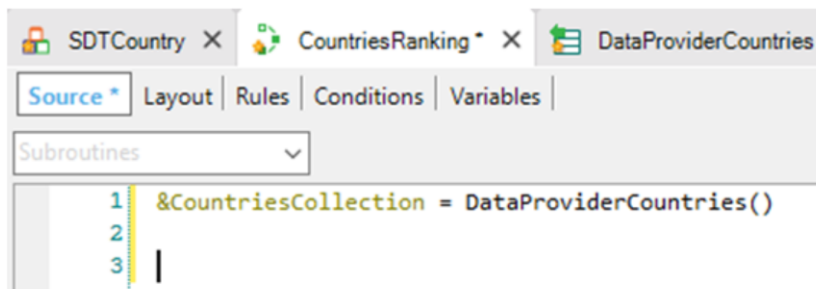


Para completar a configuração, vamos escrever de onde queremos que o data provider busque as informações, que em nosso caso, é da tabela de países. Para isso, do lado direito do sinal de igual (=) definimos os atributos CountryId para o Id e CountryName para o Name. Para Quantity, vamos definir uma fórmula Count que contará a quantidade de atrações do país.

Nosso Data Provider está pronto para ser executado.

Se clicarmos com o botão direito sobre a aba do Data Provider e escolhermos View Navigation, vemos que GeneXus especifica o objeto e nos mostra o relatório de navegação. Podemos observar que ele percorrerá a tabela Country para obter os dados dos países, que é exatamente o que queremos.

6) Carregamos o conteúdo da variável coleção



Vamos definir o básico para carregar o conteúdo da variável coleção CountriesCollection.

Em primeiro lugar, adicionamos a variável &CountriesCollection no Source e a carregamos com DataProviderCountries.

Quando fazemos a chamada do Data Provider, o mesmo percorrerá todos os registros da tabela Country e devolverá uma coleção já carregada, onde cada elemento da coleção é do tipo SDTCountry. Esta coleção será armazenada na variável coleção &CountriesCollection, que também definimos como coleção de elementos do mesmo tipo.

Observemos que a sintaxe da chamada do Data Provider não muda. A diferença é que a variável que recebe o resultado do Data Provider, em vez de estar baseada num SDT que já é coleção, está baseada em um SDT simples, e a variável é definida como coleção, marcando o check box.

Resumindo...

Temos duas formas para que um data provider nos devolva uma coleção:

- Definindo um SDT como coleção e o data provider se configura automaticamente

Name	Type	Is Collection
SDTCountries		<input checked="" type="checkbox"/>
SDTCountriesItem		<input type="checkbox"/>
Id	Id	<input type="checkbox"/>
Name	Name	<input type="checkbox"/>
CountryAttractionsQuantity	Numeric(4,0)	<input type="checkbox"/>

```

1 SDTCountries
2 {
3   SDTCountriesItem
4   {
5     Id = //Id.value//
6     Name = //Name.value//
7     CountryAttractionsQuantity = //CountryAttractionsQuantity.value//
8   }
9 }
  
```

Output	
Infer Structure	No
Output	SDTCountries
Collection	False

- Definindo um SDT que não seja uma coleção e depois através das propriedades do data provider, configuramos para que o mesmo monte a coleção para nós.

Name	Type	Is Collection
SDTCountry		<input type="checkbox"/>
Id	Id	<input type="checkbox"/>
Name	Name	<input type="checkbox"/>
Quantity	Numeric(4,0)	<input type="checkbox"/>

```

1 SDTCountry
2 {
3   Id = CountryId
4   Name = CountryName
5   Quantity = Count(AttractionName)
6 }
  
```

Output	
Infer Structure	No
Output	SDTCountry
Collection	True
Collection Name	DataProviderCountries

Resumindo, vimos que temos duas formas de devolver uma coleção de elementos através de um data provider :

- definindo um tipo de dados estruturado (SDT) do tipo coleção e, ao arrasta-lo sobre o Source do Data Provider, o mesmo se configura automaticamente para retornar uma coleção desse tipo.
- ou definindo um tipo de dados estruturado que não seja uma coleção e depois, através das propriedades do Data Provider, configurar o mesmo Data Provider para montar a coleção.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications