

Dos formas de devolver una colección con un Data Provider

GeneXus™ 16

Para que un Data Provider nos cargue una colección de elementos:

The screenshot displays the GeneXus IDE interface with three main components:

- Model Explorer (Top Left):** A tree view showing the data model. It includes a collection type `SDTCountries` and an item type `SDTCountriesItem` with attributes `Id` (Type: Id), `Name` (Type: Name), and `CountryAttractionsQuantity` (Type: Numeric(4.0)). The `SDTCountries` type has a red box around its `Is Collection` checkbox, which is checked.
- Source Editor (Bottom Left):** Shows the source code for the `RankingCountriesWithAttractionsQty` data provider. The code defines the `SDTCountries` collection and its items:


```

1 SDTCountries
2 {
3   SDTCountriesItem
4   {
5     Id = /*Id value*/
6     Name = /*Name value*/
7     CountryAttractionsQuantity = /*Coun
8   }
9 }
      
```
- Properties Window (Right):** Displays the configuration for the `RankingCountriesWithAttractionsQty` data provider. The `Output` property is set to `SDTCountries`, which is highlighted with a red box. Other properties include `Name` (RankingCountriesWithAttractionsQty), `Description` (Ranking Countries With Attractions Qty), `Expose as Web Service` (False), `Module/Folder` (Root Module), `Qualified Name` (RankingCountriesWithAttractionsQty), `Object Visibility` (Public), `Infer Structure` (No), and `Collection` (False).

Los Data Provider son objetos versátiles que con un lenguaje declarativo nos facilitan la carga de estructuras, tanto ítems simples como colecciones de ítems.

Como vimos en un video anterior, cuando queríamos que un Data Provider nos cargue una colección de elementos, lo que hacíamos era definir una variable basada en un tipo de datos estructurado que sea colección y luego arrastrar esa variable al source del data provider, con lo cual el output del mismo se configuraba automáticamente para cargar ese tipo de datos.

Ahora haremos lo mismo, pero cuando el SDT no es colección

Objetivo: ranking de países

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

1) Creamos el SDTCountry que no es colección



Ahora veremos la posibilidad que dispone un Data Provider para devolver una colección de elementos, cuando el dato estructurado **no es una colección**.

La agencia de viajes nos había solicitado un reporte que muestre un ranking de los países con más atracciones turísticas. Nuestro ejemplo implica crear una colección de países, donde cada elemento es de un tipo de dato estructurado que almacena los datos de un solo país.

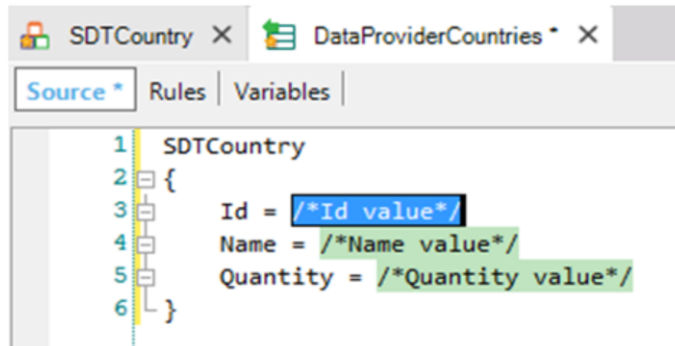
Para comenzar vamos a crear un tipo de dato estructurado de nombre SDTCountry, con un ítem Id, otro Name y otro Quantity en el que almacenaremos la cantidad de atracciones turísticas del país. A este tipo de datos no lo marcamos como colección, con lo cual será capaz de almacenar los datos de un único país.

2) Definimos una variable basada en SDTCountry y la marcamos colección

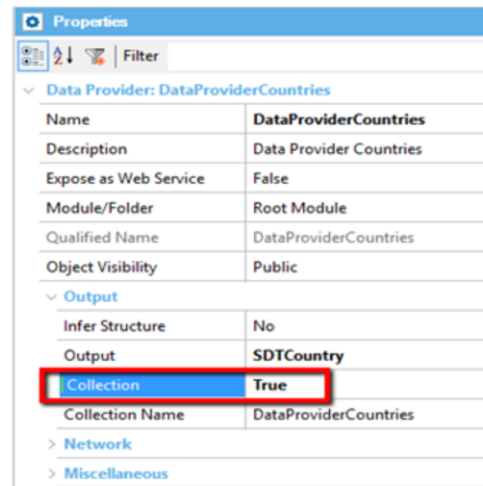


Luego, en el objeto procedimiento que implementa el ranking, definimos una variable CountriesCollection del tipo SDTCountry y la marcamos como colección.

3) Creamos el data provider y arrastramos el SDTCountry al source



4) Configuramos la propiedad Collection en True



Para obtener los datos de todos los países para confeccionar el ranking, creamos un objeto data provider de nombre DataProviderCountries. Luego arrastramos el SDTCountry al source del data provider.

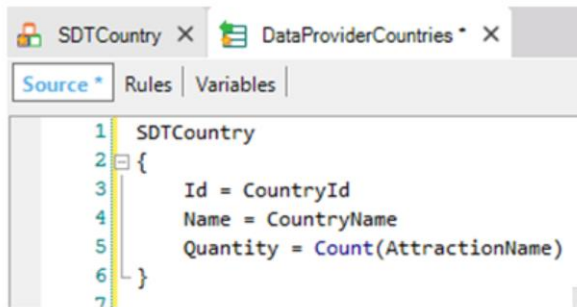
Vemos que se creó automáticamente un código que refleja la estructura del SDTCountry. Sin embargo, no encontramos ningún grupo repetitivo, ya que no hay un grupo definido por llaves que esté dentro de otro, por lo que esta definición escrita en el source, sólo será capaz de cargar los datos de un país. Pero nosotros queremos cargar los datos de todos los países...

Afortunadamente el data provider nos resuelve este problema, construyendo la colección por nosotros. Para eso vamos a las propiedades del data provider y en el grupo Output, ponemos la propiedad **Collection** en el valor True.

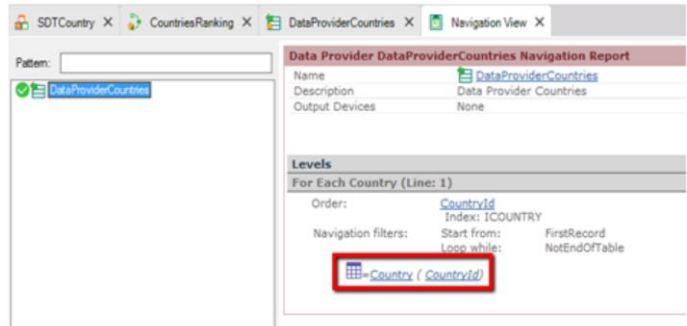
Observemos que cuando arrastramos el SDTCountry al source del data provider, la propiedad Output se configuró automáticamente con el nombre del SDT, que sabemos que almacena un único elemento.

Pero al asignar la propiedad Collection en True, le estamos indicando al data provider que queremos que nos devuelva una colección de elementos del tipo del SDTCountry. También observamos que apareció la propiedad Collection Name y que se asignó automáticamente un nombre para la colección.

5) Completamos la definición del data provider...



..y corroboramos el listado de navegación

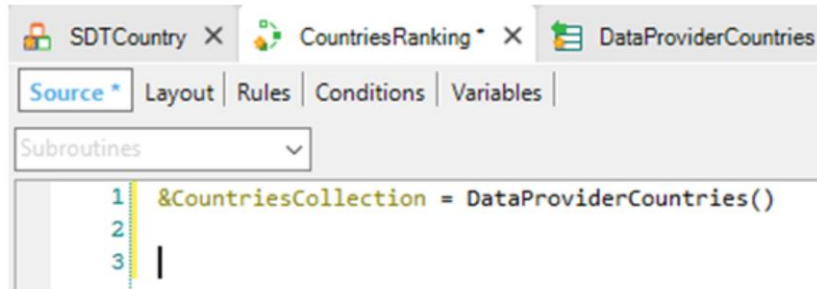


Para completar la definición, vamos a escribir de dónde queremos que el data provider obtenga los datos, que en nuestro caso es de la tabla de países. Así que asignamos a la derecha de las asignaciones los atributos CountryId para el Id y CountryName para el Name. Luego a Quantity le asignamos una fórmula Count que contará la cantidad de atracciones del país.

Nuestro data provider queda así pronto para ser ejecutado.

Si damos botón derecho sobre la solapa del nombre del data provider y elegimos View Navigation, vemos que GeneXus especifica al objeto y nos muestra el reporte de navegación. En el mismo observamos que el mismo recorrerá la tabla Country para obtener los datos de los países, que es lo que pretendíamos.

6) Cargamos el contenido de la variable colección



Definamos lo necesario para cargar el contenido de la variable colección CountriesCollection.

En primer lugar agregamos la variable &CountriesCollection al source y le asignamos el DataProviderCountries.

Al invocarse el data provider, el mismo recorrerá todos los registros de la tabla Country y devolverá cargada una colección donde cada elemento es del tipo SDTCountry. Esta colección quedará almacenada en la variable colección &CountriesCollection, que definimos también como colección de elementos del mismo tipo.

Observemos que la sintaxis con que se invoca al data provider no cambia, sólo que la variable que recibe el resultado del data provider en vez de estar basada en un SDT que ya es colección, está basada en un SDT simple, y a la variable la hacemos colección marcando el check box.

Resumiendo...

Tenemos dos formas que un data provider nos devuelva una colección:

- Definiendo un SDT colección y el data provider se configura automáticamente

The screenshot shows the GeneXus IDE interface for a data provider named 'RankingCountriesWithAttractionsQty'. On the left, the 'SDTCountries' SDT is defined with properties: Id (Id), Name (Name), and CountryAttractionsQuantity (Numeric(4,0)). The 'Is Collection' checkbox is checked. In the center, the 'Source' tab shows a rule that returns an array of SDTCountriesItem objects. On the right, the 'Output' tab shows the configuration: 'Infer Structure' is No, 'Output' is SDTCountries (highlighted with a red box), and 'Collection' is False.

- Definiendo un SDT que no sea una colección y luego mediante las propiedades del data provider configuramos que el mismo nos arme la colección.

The screenshot shows the GeneXus IDE interface for a data provider named 'DataProviderCountries'. On the left, the 'SDTCountry' SDT is defined with properties: Id (Id), Name (Name), and Quantity (Numeric(4,0)). The 'Is Collection' checkbox is unchecked. In the center, the 'Source' tab shows a rule that returns a single SDTCountry object. On the right, the 'Output' tab shows the configuration: 'Infer Structure' is No, 'Output' is SDTCountry, 'Collection' is True (highlighted with a red box), and 'Collection Name' is DataProviderCountries.

Resumiendo, vemos que tenemos dos formas de que un data provider nos devuelva una colección de elementos:

- definiendo un tipo de datos estructurado del tipo colección y al arrastrarlo al source del data provider automáticamente se configura el mismo para retornar una colección de ese tipo,
- o bien definiendo un tipo de datos estructurado que no sea una colección y luego mediante las propiedades del data provider podemos configurar que el mismo data provider nos arme la colección.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications