

GeneXus[™]
The power of doing.

Grid Control: Pull To Refresh & Inverse Loading

User Experience & User Interface

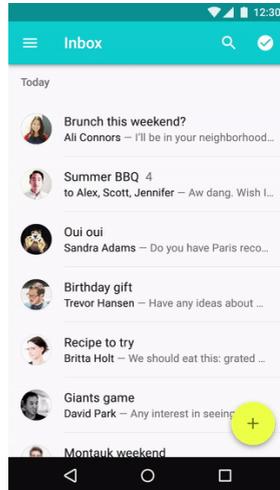
GeneXus™ 16

Grid Control

- 1 Pull To Refresh
- 2 Inverse Loading

En este video vamos a hablar de dos propiedades de las grillas muy útiles a la hora de hacer aplicaciones para Smart Devices, Pull To Refresh e Inverse Loading.

Pull To Refresh



Pull To Refresh

The pull-to-refresh pattern lets a user pull down on a list of data using touch in order to retrieve updated information.

Vamos a ver esta característica, que ya es un patrón de diseño en la mayoría de las aplicaciones móviles y con la cual todo usuario está familiarizado. Se trata de la propiedad Pull To Refresh.

El patrón Pull To Refresh permite al usuario, mediante un gesto de drag-down o swipe-down sobre una lista, recargar el contenido de la misma obteniendo información actualizada.

Cuando el usuario ejecuta la acción, se verá en la pantalla un pequeño círculo animado, por defecto, que representa el proceso de refrescar la información.

Pull To Refresh

Pull To Refresh

The screenshot shows the GeneXus IDE interface. On the left, a grid control named 'GRID' is visible, containing a table with columns for 'SpeakerFullName', 'SpeakerId', and 'CompanyName'. The 'Application Bar' contains 'MainTable' and 'GridSpeakers'. On the right, the 'Properties' window is open for the 'Grid: GridSpeakers' control. The 'Pull To Refresh' property is highlighted, and its value is 'False'. A dropdown menu is open, showing 'True' and 'False' options.

Control Name	GridSpeakers
Collection	
Default Action	<default>
Selection Type	Platform Default
Enable Multiple Selection	False
Pull To Refresh	False

En GeneXus esto se consigue con la propiedad del Grid Pull To Refresh.

Cuando está habilitada, el comportamiento default será preguntar al server si hay nueva información para devolver, en cuyo caso el grid de entrada será refrescado (y solo él, no el panel completo). Esto solo vale cuando el grid tiene table base.

Y que pasa en los casos en donde no se cuenta con un Grid con Tabla Base?

Pull To Refresh without Base Table

Pull To Refresh

```
SpeakersList* x Navigation View x
Layout | Rules | Events* | Conditions | Variables |
Events
1
2
3 Event GridSpeakers.PullRelease
4   //Look for updated info
5   DPGetSpeakers.call(&SpeakersSDT)
6 EndEvent
7 |
```

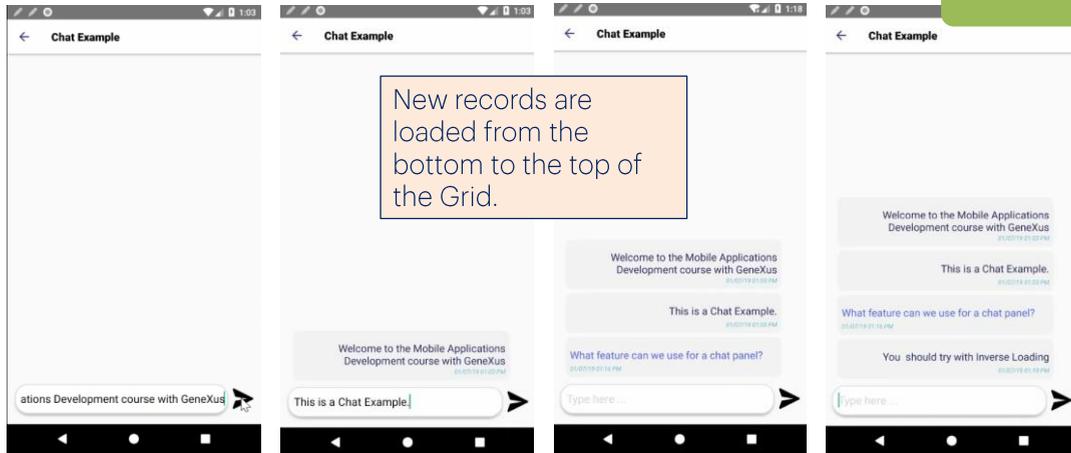
PullRelease Event is triggered when the user ends the swipe-down gesture

En los casos donde no tenemos una tabla base, por ejemplo porque la grilla esta basada en un SDT, en estos casos podemos utilizar el evento PullRelease. Este evento es disparado cuando el usuario final suelta la pantalla luego de hacer el gesto de swipe-down sobre la grilla.

Entonces en este evento podríamos llamar a un programa como vemos en pantalla, un Data Provider DPGetSpeakers, que ejecute en el servidor y retorne los resultados actualizados en un SDT.

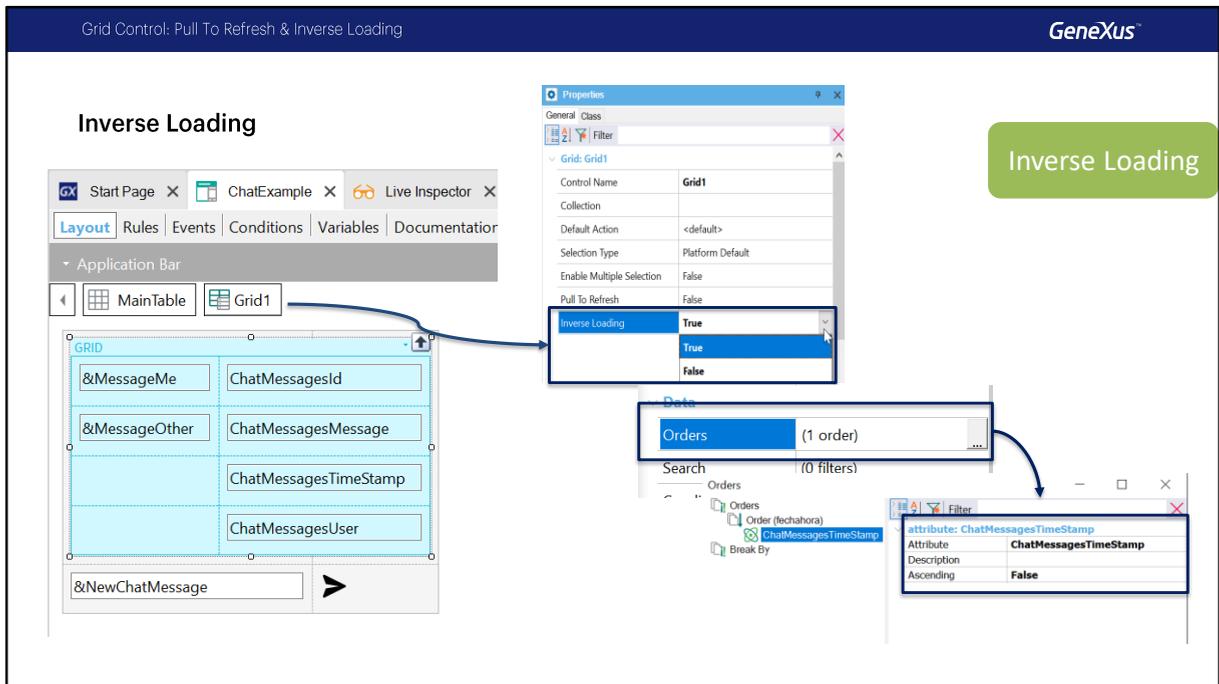
Inverse Loading

Inverse Loading



Otro patrón de diseño UI muy utilizado es la carga inversa.

Es muy utilizado por ejemplo en las aplicaciones de Chat, en estas aplicaciones generalmente los últimos mensajes aparecen debajo, al final de la pantalla, y la pantalla se va llenando desde abajo hacia arriba.



Para lograr este efecto en la aplicación, deberíamos usar la propiedad a nivel del grid llamada **Inverse Loading**.

Algo que debemos tener en cuenta es que necesitaremos ordenar la información, en el ejemplo usamos un atributo ChatMessageTimeStamp que almacene la fecha y hora del registro. De esta forma los registros más nuevos serán mostrados abajo del todo en el grid.

En el próximo video continuaremos con el estudio del control All Sections Content del Detail cuando usamos WorkWith for Smart Devices.

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications