

GeneXus[™]
The power of doing.

Panel for Smart Devices

Behavior Development

GeneXus™ 16

Panel for Smart Devices

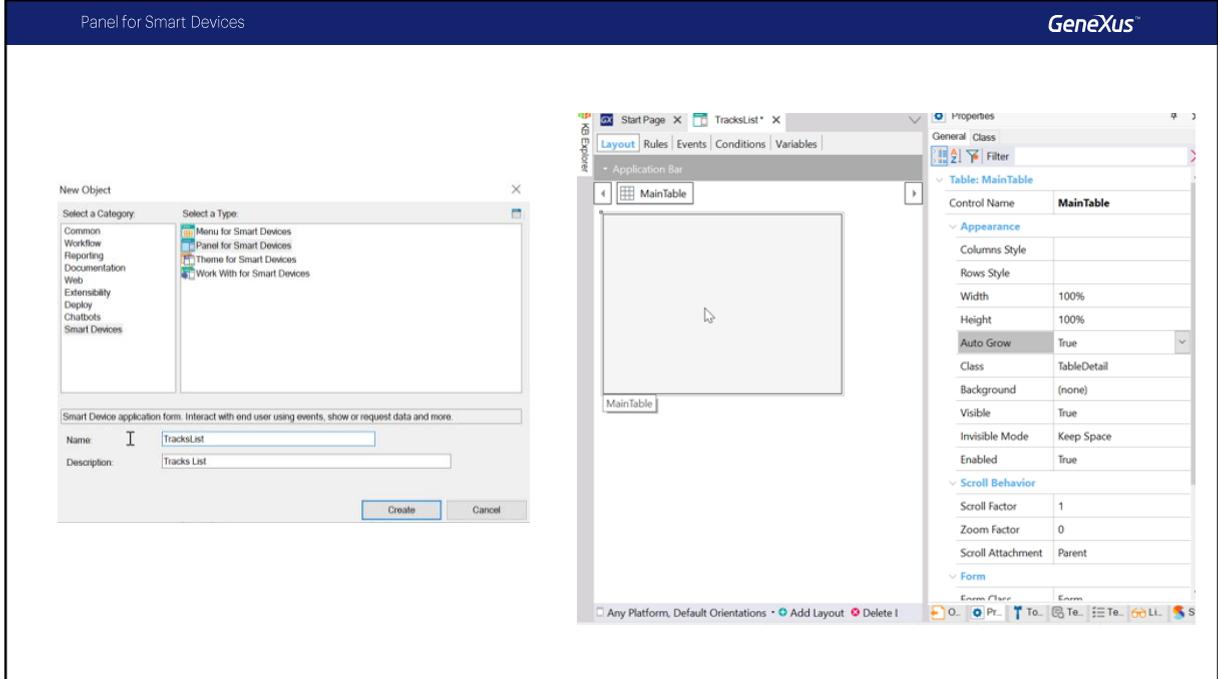
- Flexible Layout
- Show or Ask Data
- Complex Screens (Wizards)
- Menu (app Entry Point – Main Object)
- Reusable Components
- Load data from Data Providers or REST Services
- Multiple Grids
- And many more...

En este video vamos a ver en detalle algunas de las posibilidades que nos brindan los objetos Panel for Smart Devices y que lo hacen tan flexibles.

- La característica mas importante es justamente que el Layout es flexible, de hecho vimos que cuando creamos este objeto esta completamente vacío por lo que toda la implementación queda en manos del desarrollador.
- Por ejemplo podríamos utilizar un panel para mostrar información al usuario o para que el usuario ingrese información o incluso una combinación de ambas opciones al mismo tiempo.
- Podemos realizar pantallas complejas, por ejemplo para crear un Asistente con varios pasos
- Podríamos usarlo para crear un Menú ya que también puede ser Main por lo tanto un Entry Point de nuestra aplicación.
- Podemos utilizarlo como componentes reutilizables (esto lo veremos en detalle en el nivel avanzado)
- Podemos utilizarlo para mostrar información devuelta por Data Providers o Servicios REST
- Podemos implementar múltiples Grids
- Y muchos otros usos que ustedes encontraran además de estos que mencionamos.

Demo: Panel for Smart Devices

Veamos un Ejemplo en GeneXus.



En esta demo vamos a ver la flexibilidad que nos ofrecen los objetos Panel for Smart Devices.

Entonces vamos a crear un Panel desde cero, el cual usaremos para mostrar una lista con los Temas que mas conferencias tienen asociadas.

Crearemos un objeto de tipo Panel for Smart Devices y le pondremos de nombre TracksList, como ya habíamos mencionado, el panel se crea vacío y queda en manos del desarrollador toda la implementación.

Panel for Smart Devices GeneXus™

GetMyTracks x

Source Rules Variables

```

1 TrackItem [Count = &Qty]
2 order (TrackSessionCount), TrackDescription when &Qty <> 9999
3 order TrackDescription when &Qty = 9999
4 {
5   TrackId
6   TrackDescription
7   TrackClassID
8 }
9

```

GetMyTracks x

Source Rules Variables

```

1 parm(in:&Qty);
2

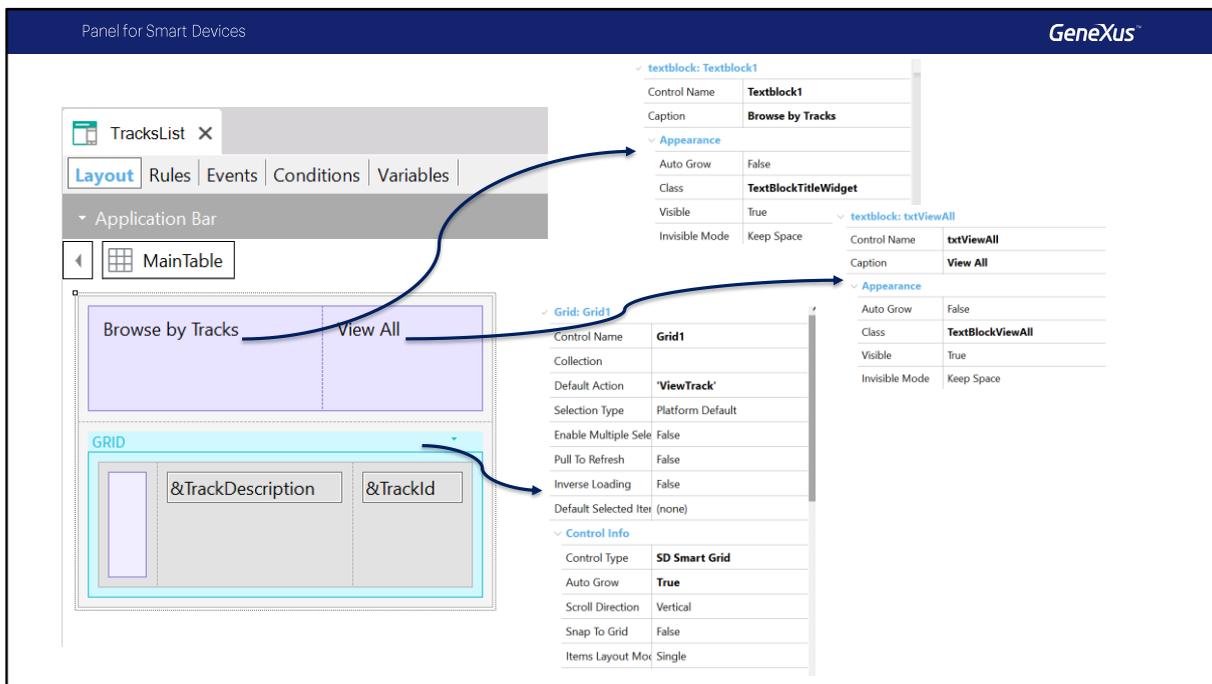
```

Formula	count(SessionId, SessionActive = Status.Active)
---------	---

Output

Infer Structure	No
Output	TrackItem
Collection	True
Collection Name	GetMyTracks

Vamos a usar un Data Provider , GetMyTracks, que resuelve este caso, este Data Provider recibe como parámetro la cantidad que deseamos mostrar, con esa cantidad vamos a hacer un count, para solo obtener esa cantidad de registros, y vamos a ordenar los resultados por un atributo Formula, TrackSessionCount, veamos como esta definido, cuenta la cantidad de conferencias activas de un Tema, pero solo en caso de que esa cantidad sea distinta de 9999, sino se ordenara ordenará por descripción. Este panel retornara una colección del tipo TrackItem.



Volvamos al Panel.

Para mostrar la grilla, como vamos a usar un data provider, vamos a crear unas variables que nos servirán para mostrar los datos en la grilla justamente. &TrackId basada en el atributo y &TrackDescription también basada en el atributo.

Vamos al Layout, aquí vamos a agregar una grilla, fijense que podemos usar cualquier tipo de control, seleccionamos las dos variables que recién creamos, ahora para poder utilizar un Stencil que ya tenemos, vamos a agregar una Tabla y movemos las variables adentro. El Stencil que usaremos es TrackView.

Vamos entonces a aplicar el Stencil, hacemos Search for Compatible Stencils, escribimos "track" y presionamos Swap.

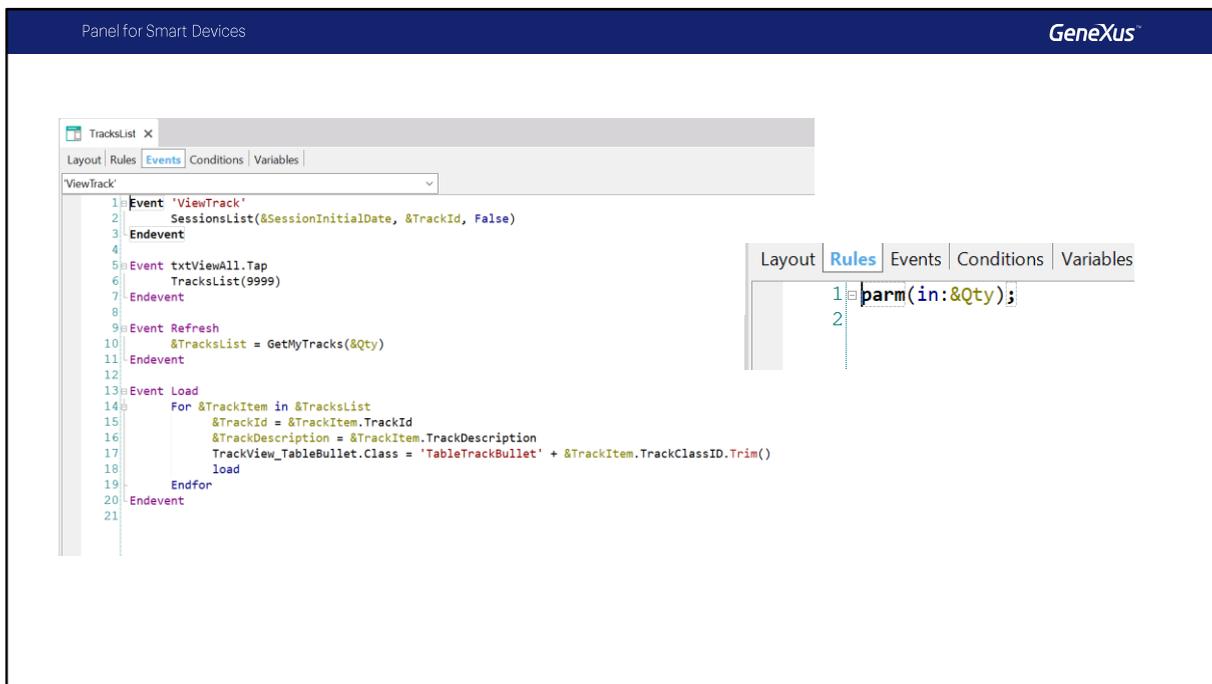
Para la grilla vamos a cambiar la propiedad Control Type, usaremos SD Smart Grid, y la propiedad Auto Grow en True.

También vamos a asignarle un Default Action , vamos elegir New y de nombre al evento le pondremos "ViewTrack".

Ahora vamos a agregar una Tabla arriba de la grilla, adentro vamos a agregar un TextBlock, le cambiamos el Caption, este va a ser el titulo. Vamos a ingresar "Browse By Tracks" y en la clase usaremos TextBlockTitleWidget.

Vamos a agregar otro TextBlock al lado derecho, le pondremos de nombre txtViewAll y en el Caption vamos a ingresar ViewAll, en la clase usaremos TextblockViewAll. Vamos a corregir la alineación vertical para que se vea en el medio y la alineación horizontal a la Derecha y al otro TextBlock vamos a ponerlo también igual al medio, bien.

Vamos a crear el evento Tap sobre este Textblock.



Primero vamos a programar el evento Refresh para que se ejecute el Data Provider, GetMyTracks, debemos pasarle a este data Provider la cantidad de Temas que queremos ver, utilizaremos una variable, esta variable la vamos a recibir como parámetro.

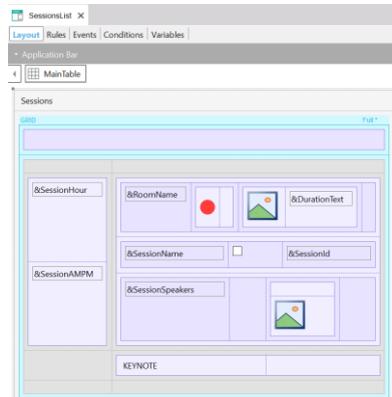
Este Data Provider retorna una colección del tipo TrackItem, vamos a asignar el resultado a la variable &TracksList de ese tipo de dato, TrackItem de tipo Collection "True".

En el Evento Load, vamos a recorrer la colección con el comando For In, y vamos a asignar los valores a las variables &TrackId desde el ítem, lo mismo para &Trackdescription, también recuperamos del ítem.

Para mostrar el color del track, primero necesito ver como se llama la tabla, ok TrackView_TableBullet, entonces usamos la propiedad Class de esta tabla y le asignamos una clase distinta para cada uno, como hacemos esto, concatenando el nombre de la clase, que es TableTrackBullet y le concatenamos el nombre con el ClassID del ítem, porque las clases terminan en 01, 02 etc.

Luego ponemos el comando Load, para cargar el ítem y terminamos.

En el evento txtViewAll.TAP vamos a llamar a este mismo objeto "TracksList", pero le pasaremos como parámetro el valor 9999, esto es para que no filtre la cantidad de resultados.



```

2 Event Refresh
3     //Get Device Id (used to show Favorite Sessions)
4     &DeviceId = GeneXus.Client.ClientInformation.Id
5     //call DP to get Sessions for a Track
6     &SessionList = GetSessionsByTrack(&DeviceId, &TrackId, &SessionInitialDate)
7 Endevent

```

En el evento ViewTrack llamaremos al panel SessionList que ya tengo preparado, vamos a verlo.

El panel muestra un titulo y una grilla con las conferencias.

Este panel recibe 3 parámetros, una Fecha, un Track Id, que es lo que necesitamos, y una bandera si es KeyNote o no, nosotros lo vamos a usar justamente para filtrar por Track Id.

Veamos los eventos.

En el evento Refresh capturamos primero el Id del dispositivo usando una API, y luego llamamos a un Data Provider que retorna una lista de sesiones filtradas por Tema, por eso le pasamos como parámetro el Id de dispositivo, el id del track que necesitamos y la fecha. Noten que siempre que usemos una grilla basada en una colección, lo adecuado es llamar al Data Provider en el evento Refresh, que se ejecuta antes del evento Load, y luego en el Load recorreremos la colección.

```

99: Event Load
100: //Load grid items from SDT
101: for &SessionItem in &SessionList
102:   &SessionTitle = &SessionItem.SessionName.Trim()
103:   &SessionStartTimeText = &SessionItem.SessionInitialTime.ToString().Trim()
104:   &SessionId = &SessionItem.SessionId
105:   &RoomName = &SessionItem.RoomName.Trim()
106:   &SessionName = &SessionItem.SessionName.Trim()
107:   &SessionSpeakers = &SessionItem.SessionSpeakers.Trim()
108:   &SessionHour = &SessionItem.SessionHour.Trim()
109:   &SessionMins = &SessionItem.SessionMins
110:   &Favorite = &SessionItem.SessionIsFavorite
111:   //show or hide table if session is a keynote
112:   if &SessionItem.SessionIsKeynote=True
113:     SessionInGrid_TableKeynote.Visible=True
114:   else
115:     SessionInGrid_TableKeynote.Visible=False
116:   endif
117:   //Show or Hide information if Session is a Live Session
118:   if &SessionItem.SessionIsLive = True
119:     SessionInGrid_ImageView.Visible=True
120:     SessionInGrid_Textblock1.Visible=True
121:     SessionInGrid_TableLive.Visible=True
122:   else
123:     SessionInGrid_ImageView.Visible=False
124:     SessionInGrid_Textblock1.Visible=False
125:     SessionInGrid_TableLive.Visible=False
126:   endif
127:   //set the duration icon if necessary
128:   do case
129:     case &SessionItem.SessionDuration = 30
130:       &DurationImage.FromImage(Time30)
131:     case &SessionItem.SessionDuration = 45
132:       &DurationImage.FromImage(Time45)
133:     case &SessionItem.SessionDuration = 60
134:       &DurationImage.FromImage(Time60)
135:     otherwise
136:       &DurationImage.Visible = False
137:   endcase
138:   //set Favorite icon
139:   if &Favorite
140:     &FavImage.FromImage(ScheduleFavOn)
141:   else
142:     &FavImage.FromImage(ScheduleFavOff)
143:   endif
144:   //load record on grid
145:   load
146: endfor
147: endevent

```

```

60: Event ClientStart
61: composite
62: //set Form Title
63: do case
64:   case not &TrackId.IsEmpty()
65:     &TrackDescription = getTrackDescription.Udp(&TrackId)
66:     Title.Caption = Format("MI Sessions", &TrackDescription.ToString())
67:   case not &SessionInitialDate.IsEmpty() and &TrackId.IsEmpty() and &IsKeynote = False
68:     Title.Visible = False
69:     Title.Caption = ''
70:   case &IsKeynote = True
71:     Title.Caption = 'Featured Sessions'
72:     &Qty = 3
73:   case not &SessionInitialDate.IsEmpty()
74:     Title.Visible = False
75:     Title.Caption = ''
76:   otherwise
77:     Title.Caption = 'Sessions'
78:     &Qty = 9999
79:   endcase
80: endcomposite
81: Endevent

```

```

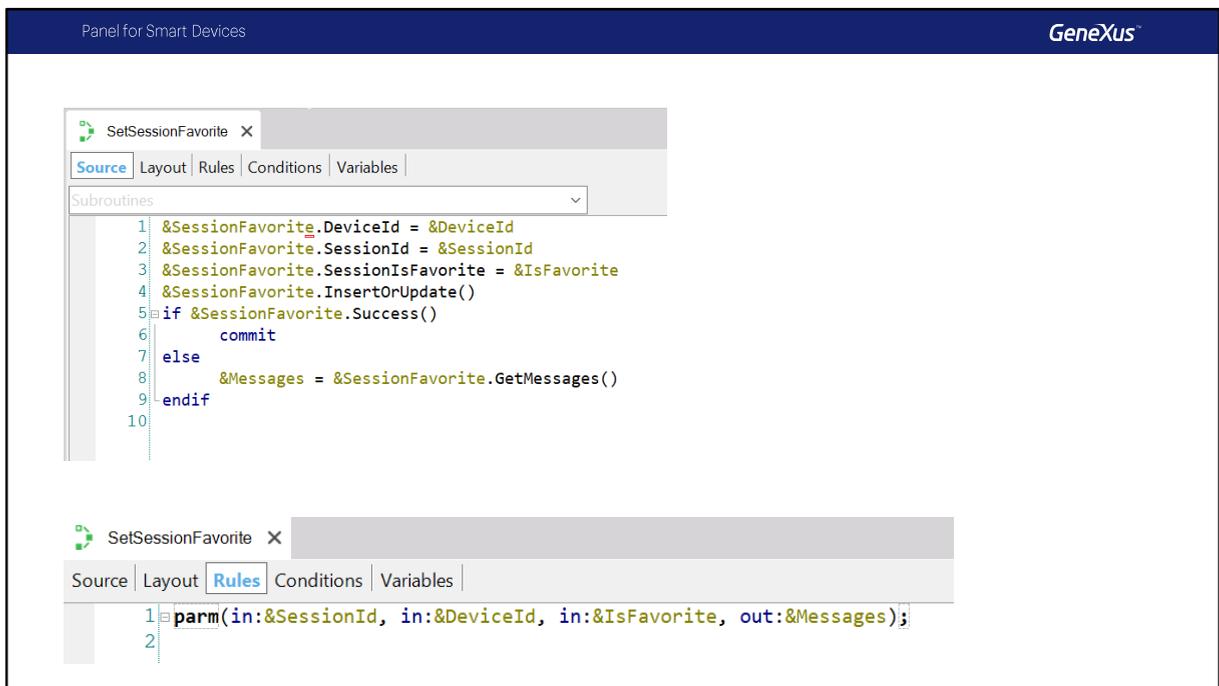
85: Event &FavImage.Tap
86: Composite
87: //Set Session as favorite
88: &DeviceId = GeneXus.Client.ClientInformation.Id
89: if &Favorite
90:   SetSessionFavorite.Call(&SessionId, &DeviceId, False)
91: else
92:   SetSessionFavorite.Call(&SessionId, &DeviceId, True)
93: endif
94: refresh
95: Endcomposite
96: Endevent
97:

```

En el evento Load vamos a recorrer la colección y asignamos los valores a las variables de la grilla y configuramos otros controles. al finalizar el comando Load.

En el evento Client_Start configuramos el titulo.

En el evento Tap sobre el icono de favoritos, llamamos a un proceso que marca la conferencia como favorita o la desmarca, según sea el caso.

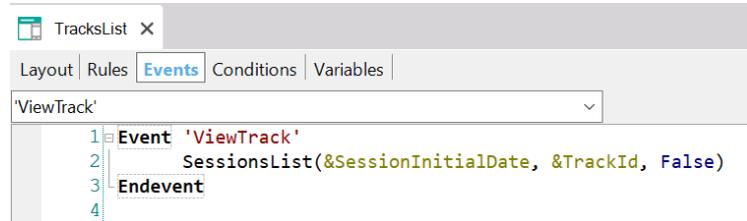


Veamos este Proceso, fíjense que acá usamos un Business Component para insertar o actualizar la marca de favorito del usuario. Este panel retorna los mensajes del Business Component, pero fíjense que no lo envío como parámetro, esto es algo que se resuelve solo para este parámetro, cuando es el ultimo y estamos en un evento del lado del cliente, es el comando Composite que va a encargarse de mostrar los mensajes si es necesario.

```
85 Event &FavImage.Tap
86 Composite
87     //Set Session as favorite
88     &DeviceId = GeneXus.Client.ClientInformation.Id
89     if &Favorite
90         SetSessionFavorite.Call(&SessionId, &DeviceId, False)
91     else
92         SetSessionFavorite.Call(&SessionId, &DeviceId, True)
93     endif
94     refresh
95 Endcomposite
96 Endevent
97
98
99 Event 'ViewSession'
100     //show Session Detail
101     WorkWithDevicesSession.Session.Detail(&SessionId)
102 Endevent
```

Al finalizar el evento se hace un Refresh, esto es para que veamos el cambio en el icono, de lo contrario no lo veríamos.

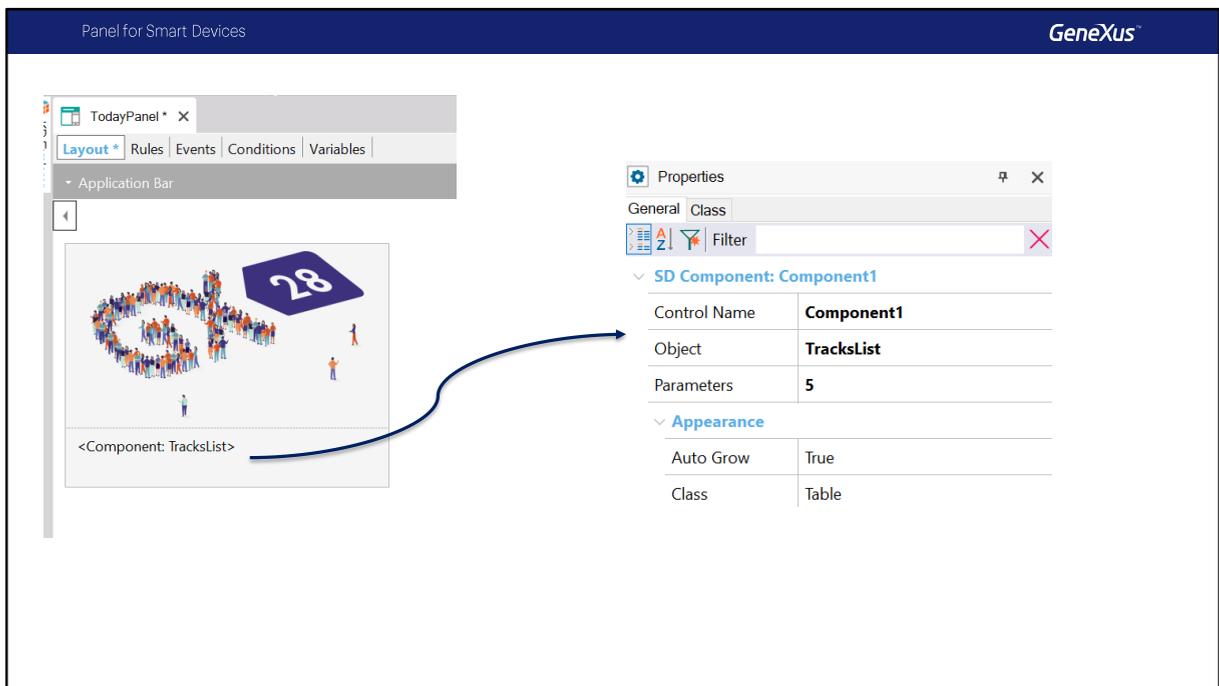
Por ultimo tenemos un evento ViewSession para acceder al detalle de la sesión, acá llamamos al View del WorkWith que ya conocemos.



```
1 Event 'ViewTrack'  
2   SessionsList(&SessionInitialDate, &TrackId, False)  
3 Endevent  
4
```

Volvamos a TracksList, vamos a poner el call a SessionsList, le pasamos los parámetros &SessionInitialDate, &TrackId, y el valor False para que no filtre las que son Key Note.

Este panel, lo vamos a llamar dentro de otro, vamos a ver como hacemos esto.

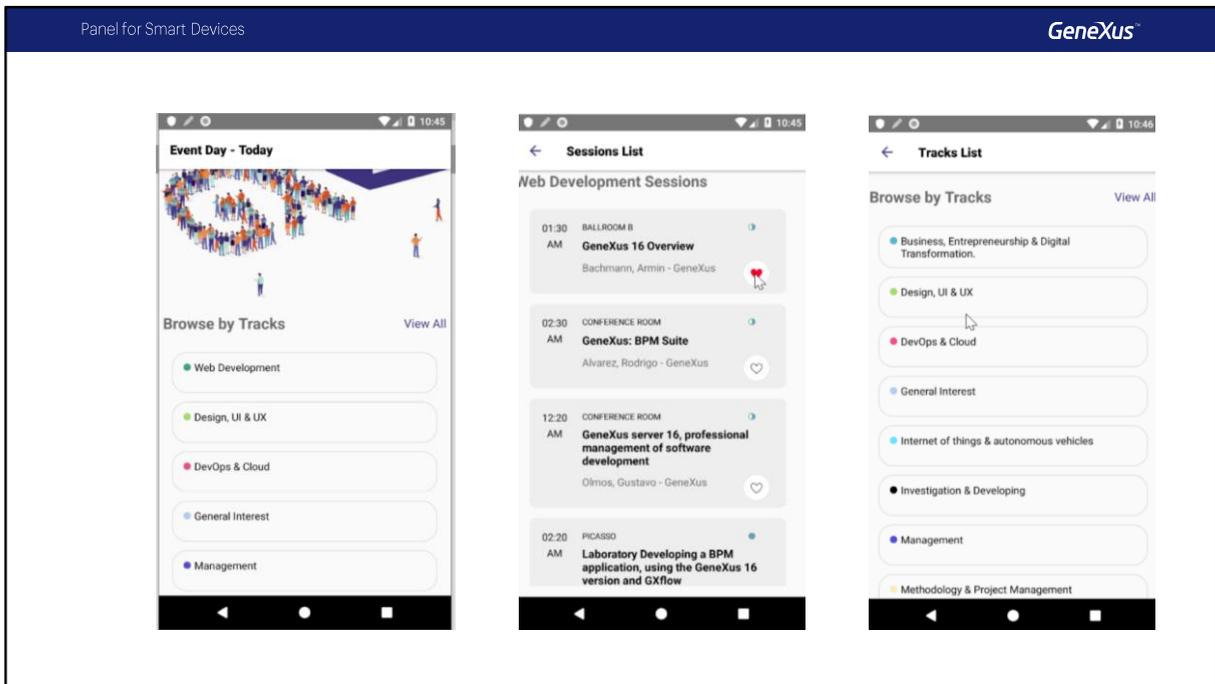


Vamos a usar un panel que ya teníamos creado, TodayPanel, y vamos a agregar un “Component”, estos controles son similares a los WebComponent que se usan en Web. Vamos a verlo.

Arrastramos Component debajo de la imagen, y en las propiedades, en el objeto, vamos a seleccionar entonces TracksList y como parámetro le pasaremos 5 que es la cantidad de temas que queremos mostrar en este panel.

Habíamos dicho que podíamos ejecutar un panel como si fuera un menú, entonces, vamos a configurar el panel para que sea Main, ahora podemos ver las propiedades de los objetos Main, y las propiedades de Android, tal cual las habíamos visto en el objeto Menu for Smart Devices.

Vamos a correr esta aplicación, podríamos ponerlo Startup Object pero vamos a usar la opción RUN.



Bien, vamos al emulador.

Ahora vemos los Tracks, acá solo veremos los 5 temas con mas conferencias asociados. Vamos a hacer TAP en el primero, tiene 4 conferencias, podemos marcar una como favorita, hacemos Tap en el Icono y ahí queda marcada. Volvamos, accedamos a otro tema, este solamente tiene 2 conferencias, marquemos una mas como favorita. Bien.

Ahora veamos el link a Todos los Temas, aquí incluso vamos a ver algunos que no tienen conferencias, por ejemplo Research & Development, porque se ven aquí si no tienen conferencias?

Data Provider GetMyTracks Navigation Report

Name:	GetMyTracks	Environment:	Default (CR)
Description:	Get My Tracks	Spec. Version:	16_0_1-128648
Output Devices:	None	Form Class:	HTML
		Program Name:	GetMyTracks
		Parameters:	in: &Qty, out: TrackItem

LEVELS

For Each Track (Line: 1)

Order: `[TrackSessionCount] , TrackDescription WHEN &Qty <= 9999`
 No Index
 TrackDescription WHEN &Qty = 9999
 Index: UTRACK
 TrackId OTHERWISE
 Index: ITRACK

Navigation filters:
 Start from: FirstRecord
 Loop while: NotEndOfTable

Join location: Server
 Optimizations: Server Paging

Navigation:
 [Track (TrackId) INTO TrackId TrackClassId TrackDescription
 [TrackSessionCount.navigation (TrackId)

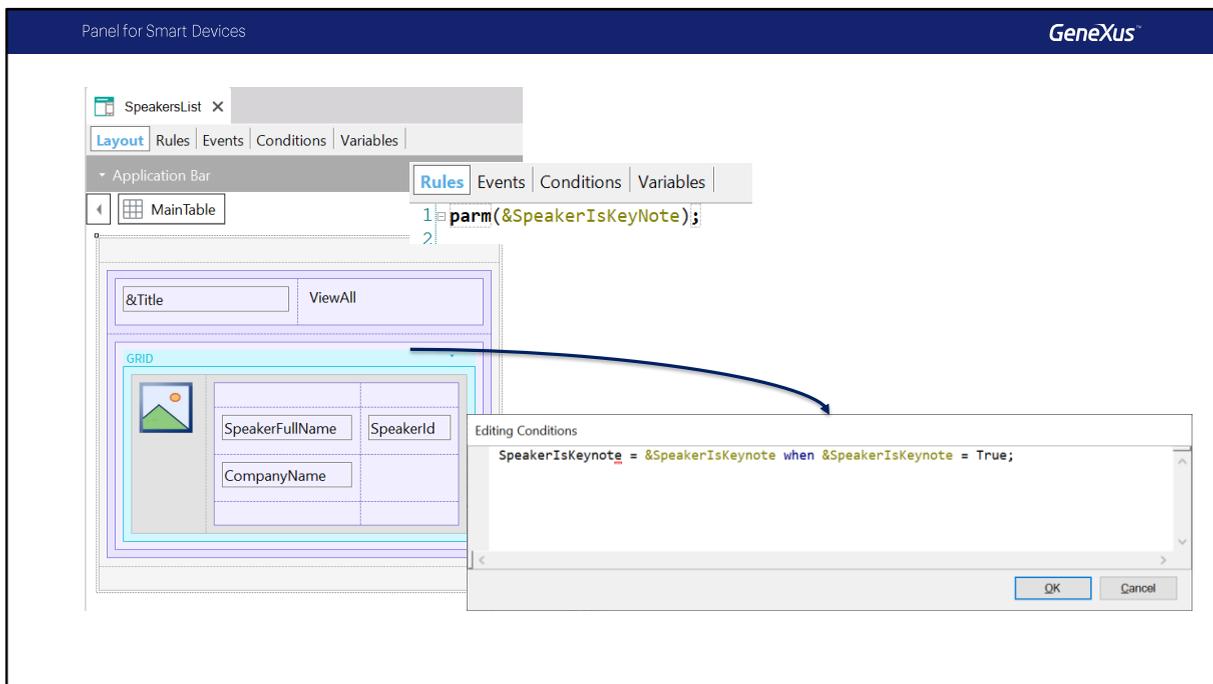
Formulas

Navigation to evaluate: TrackSessionCount
 Formula: `count(SessionId , SessionActive = Status.Active)`

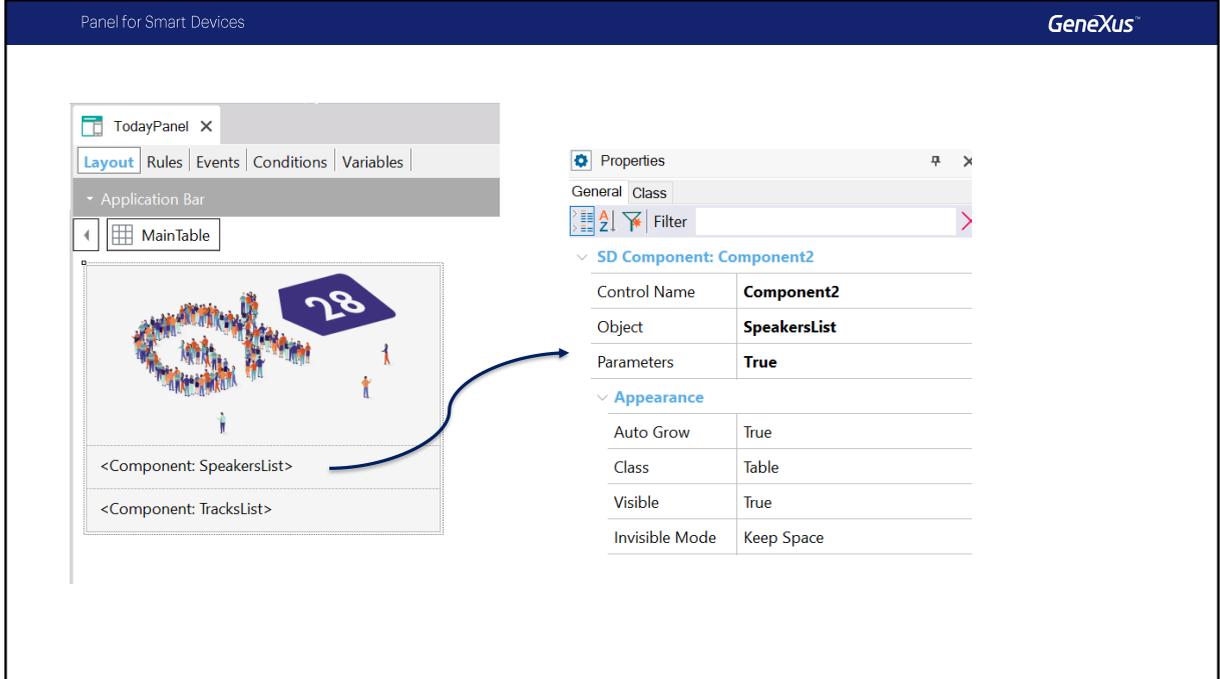
Where: `SessionActive = Status.Active`
 Given: TrackId
 Index: ITRACK
 Group by: TrackId

Navigation:
 [SessionTrack
 [Session (SessionId)

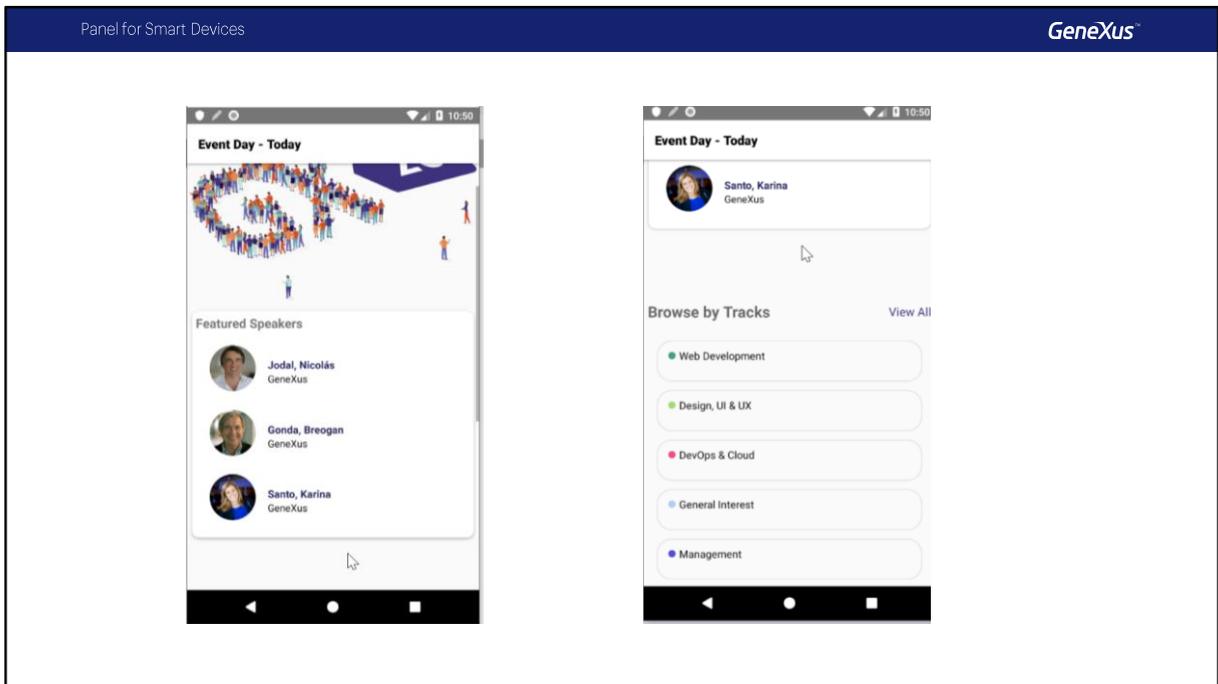
Veamos el listado de navegación del data provider que resuelve estos datos, y fíjense que la tabla base es Track y no a la tabla SessionTrack, por eso salen todos los temas y no solo aquellos que tienen una conferencia dada.



Bien, vamos a agregar otro panel, SpeakersList, este panel ya lo vimos varias veces, la diferencia es que ahora recibe una variable &IsKeyNote para filtrar en la grilla, si deseamos, solamente aquellos oradores que sean KeyNote cuando esta variable sea True.



Entonces vamos a agregarla a TodayPanel, agregamos otro componente, vamos a dejarlo arriba del anterior, elegimos el objeto SpeakersList y vamos a ponerle True (en el parámetro) porque queremos mostrar solamente los oradores principales. Grabamos y vamos a darle Run nuevamente.



Bien, ahora vemos este otro componente, que nos muestra los oradores principales.

A lo largo del curso , iremos desarrollando otros paneles para completar la aplicación y veremos otros usos de los paneles a medida que introduzcamos otros temas.

Con esto damos por finalizado el tema de Panels For Smart Devices. en el próximo video comenzaremos a ver la integración mediante el uso de APIS.

GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications