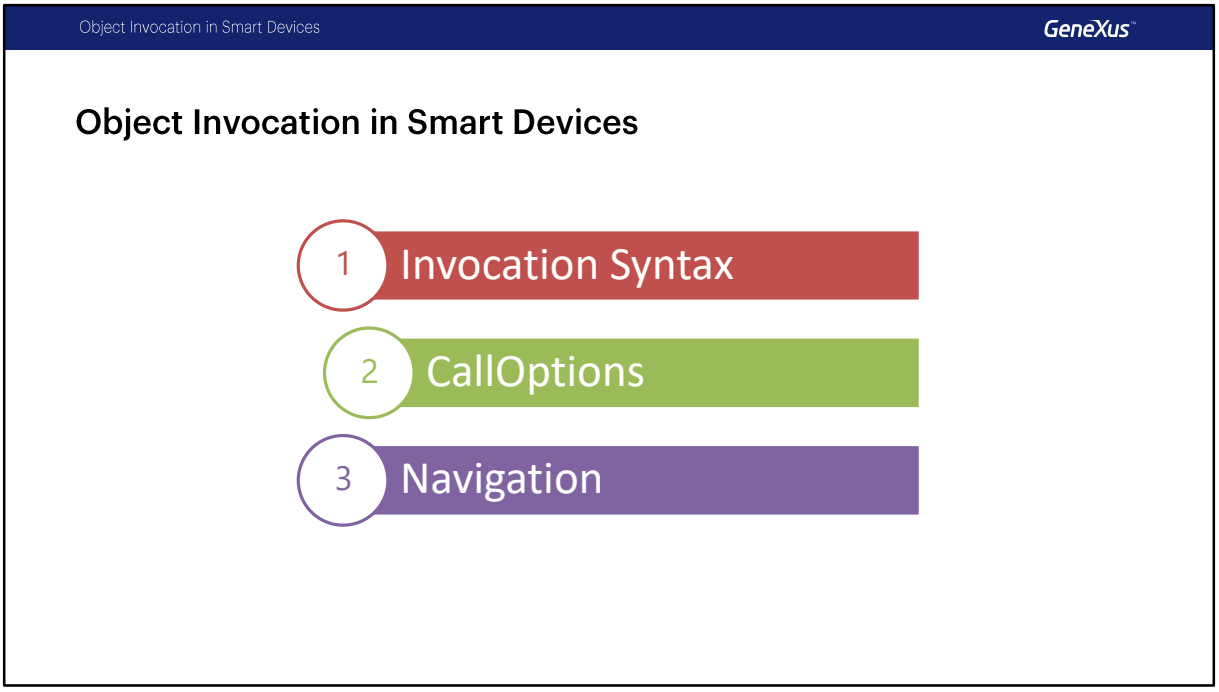

GeneXus™
The power of doing.

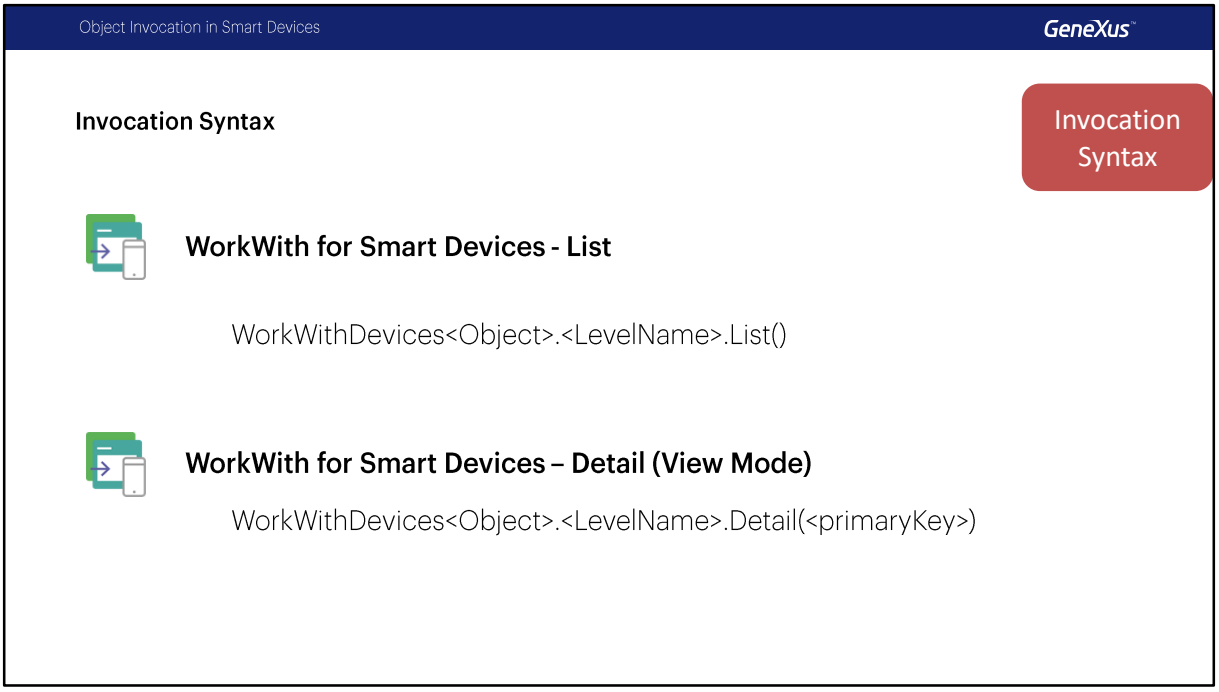
Object Invocation in Smart Devices

Behavior Development

GeneXus 16



En este video estudiaremos los objetos que se pueden invocar, cómo y cual será la sintaxis en cada caso.
Luego veremos el uso de CallOptions donde estudiaremos qué opciones de invocación tenemos, los tipos de invocación que podemos conseguir.
Por ultimo veremos como usar el Objeto Externo Navigation provisto por GeneXus.

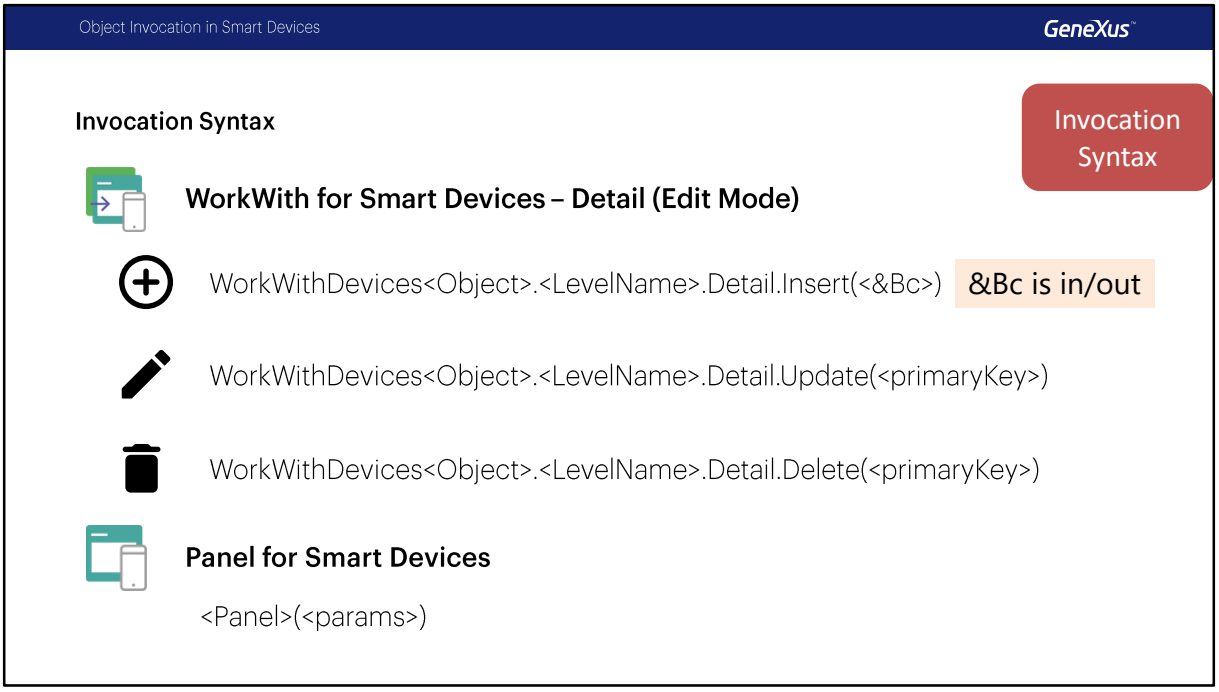


Primero vamos a hacer un repaso de la sintaxis de las invocaciones a los WorkWith y a los paneles, que son los principales objetos con interfaz en las aplicaciones para Smart Devices.

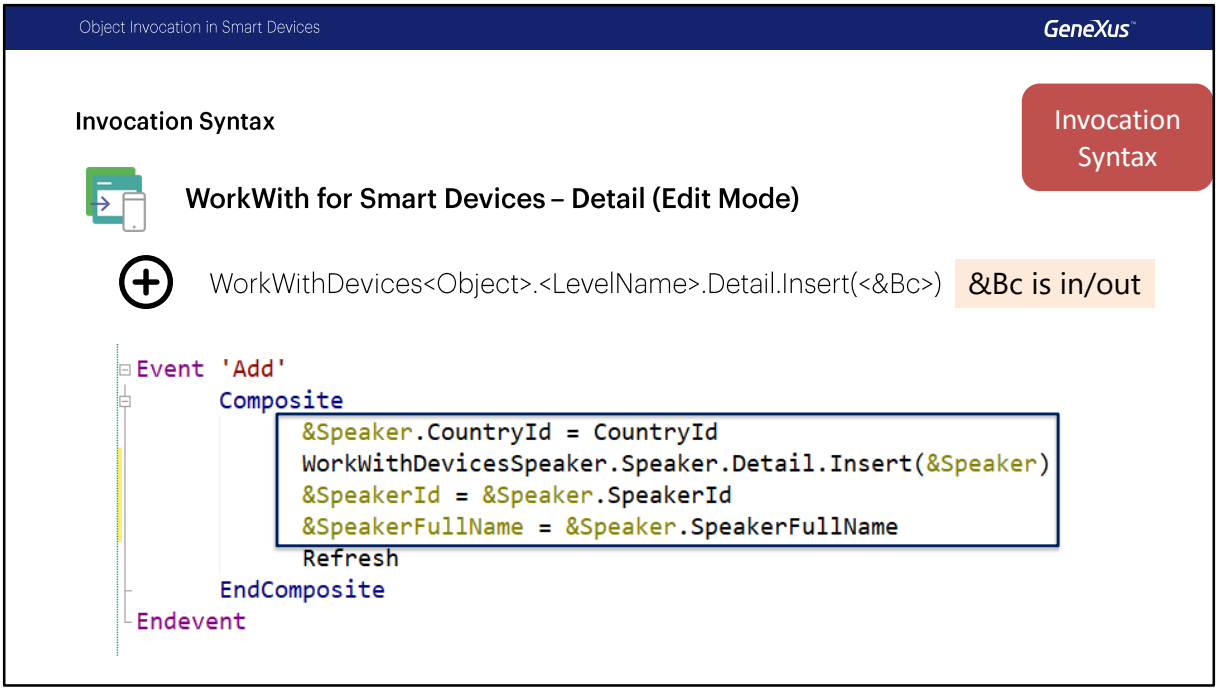
El List del WorkWith habíamos visto que se invocaba de esta manera: con el nombre del WorkWith Devices del objeto que correspondiera, el nombre del nivel, y luego el método List sin parámetros.

Para el Detail en modo View, era esta otra, en lugar de List usamos el nodo Detail, y necesitamos pasar la primary key, para identificar de quién queremos mostrar ese detalle.

El parámetro puede ser un atributo, una variable o un valor fijo, en definitiva una expresión que sea del tipo de dato de la clave primaria que se espera recibir.



Para el caso del Detail en modo Edit, debemos indicarle el modo:
En modo insert llamamos al nodo Detail.Insert y como parámetros podemos pasarle ninguno u opcionalmente un Business Component del tipo de la transacción que se trate.
si queremos actualizar los datos debemos usar el método Update, y le tenemos que pasar la clave primaria de el registro que queremos actualizar.
Por ultimo , si queremos eliminar, usamos el método Delete, y también debemos pasar la clave primaria del registro que deseamos eliminar.



Veamos el caso del modo Insert que es especial, primero que el Business Component va a ser un parámetro de in-out, de entrada y de salida; va a volver cargado con todo lo que el usuario ingresó en esa pantalla.
También, si necesitábamos pasarle algún parámetro a esa pantalla del Detail, antes de llamar al Detail en modo Insert podemos inicializar los valores que le queremos pasar en ese Business Component y luego esos valores van a aparecer inicializados en la pantalla de Insert, el usuario va a modificar los valores que le interesen, y va a grabar los datos.

Por ejemplo, en este caso de la llamada a insert de un Orador la clave primaria de Speaker es autonumerada, supongamos que necesitamos saber el valor que se le dio a esa clave primaria o el nombre completo, entonces en ese caso podemos utilizar el Business Component luego de la invocación, que retorna con los valores asignados.

Y por otro lado tenemos la invocación a los paneles, que es exactamente igual a la invocación a cualquier otro objeto GeneXus.

Invocation Syntax

Invocation Syntax

WorkWith for Smart Devices – Detail (Edit Mode)

WorkWithDevices<Object>.<LevelName>.Detail.Insert(<&Bc>) &Bc is in/out

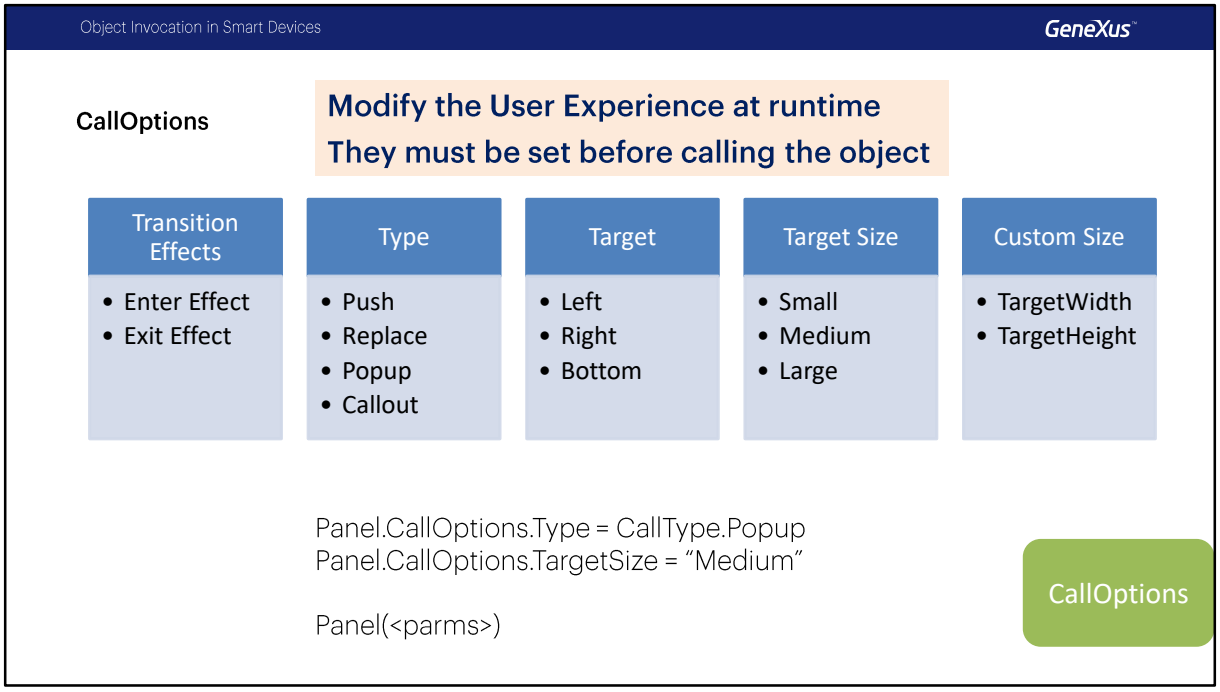
WorkWithDevices<Object>.<LevelName>.Detail.Update(<primaryKey>)

WorkWithDevices<Object>.<LevelName>.Detail.Delete(<primaryKey>)

Panel for Smart Devices

<Panel>(<params>)

Y por otro lado tenemos la invocación a los paneles, que es exactamente igual a la invocación a cualquier otro objeto GeneXus.



Las CallOptions nos permitirán modificar en runtime ciertas propiedades que tienen que ver con la experiencia del usuario, estas configuraciones deberán ser realizadas un instante antes de invocar al objeto.

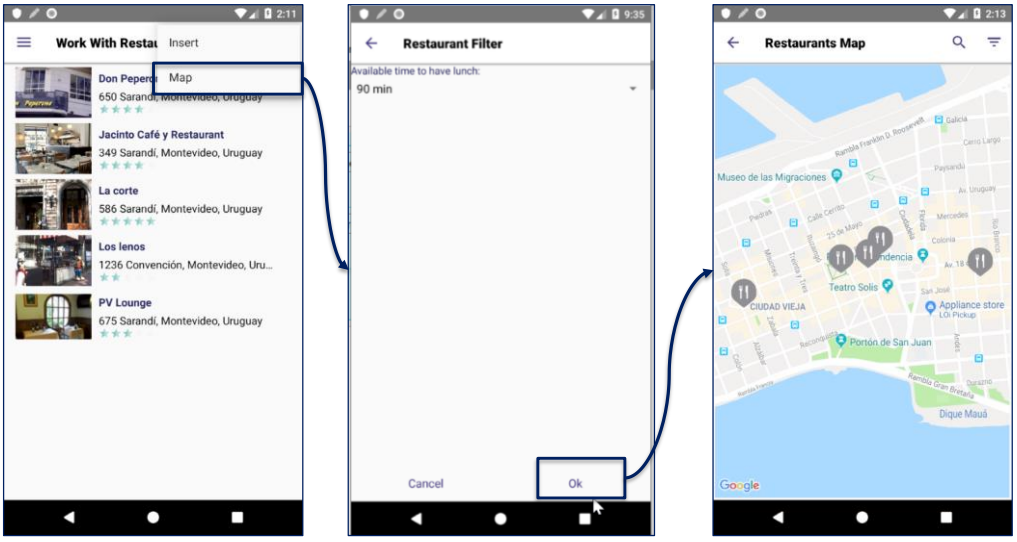
Las propiedades que podemos configurar son:

- Los efectos de transición, que pueden ser el de Entrada o el de Salida (los valores posibles son los del dominio Effect)
- El tipo de llamado, con los tipos Push, Replace, Popup y Callout (este ultimo solo disponible en iOS), acá se utiliza el dominio enumerado CallType
- El target donde se desplegara, que puede ser Izquierda, Derecha, y Abajo (este ultimo solo para iOS)
- El tamaño, con las opciones Small, Medium y Large
- y de manera personalizada el tamaño especificando el Ancho y Alto, en dips o porcentajes, usando las propiedades TargetWidth y TargetHeight

Por ejemplo si quisiéramos llamar un panel como Popup con un tamaño Medium podríamos configurar las propiedades Type y TargetSize como vemos en pantalla y luego hacer el llamado al objeto normalmente.

Demo: Object Invocation & CallOptions

Vamos a ver un ejemplo en GeneXus.



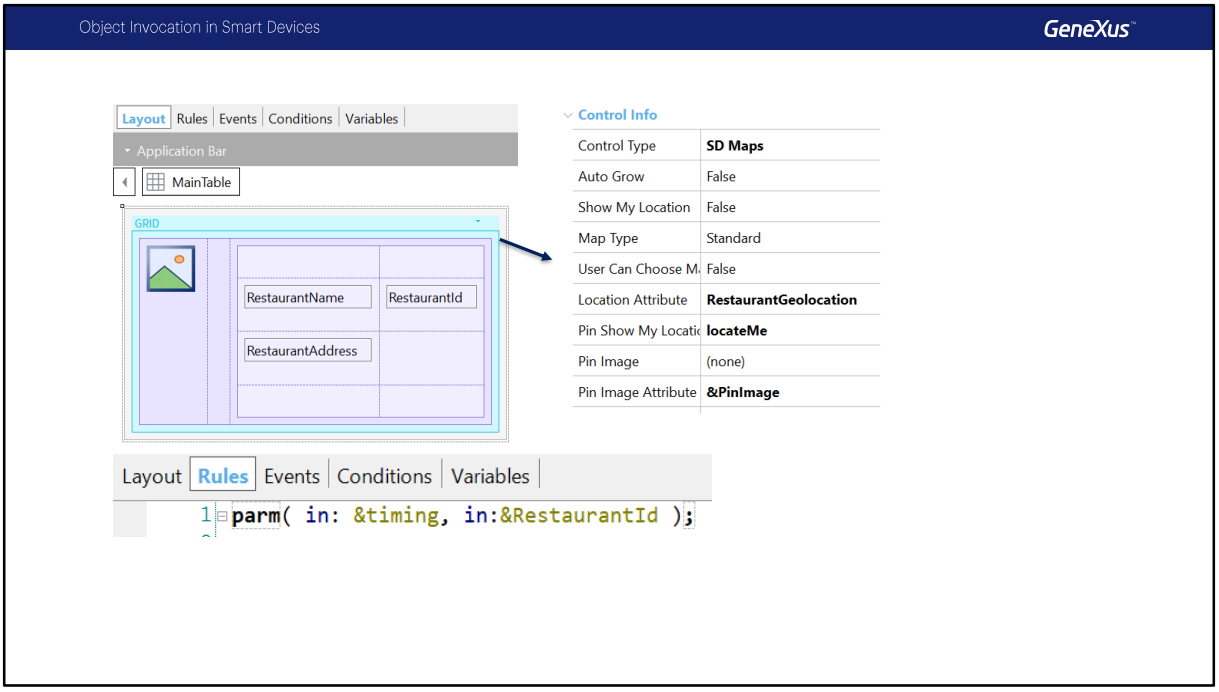
Recordaran que en otro video, cuando vimos el uso de Action Group y personalizamos el Aplication Bar habíamos estado trabajando sobre el WorkWithDevicesRestaurant, habíamos agregado un botón Map en ese ejemplo pero nunca programamos dicho botón, bueno, en este caso les voy a mostrar como completarlo.

Primero vamos a ver en ejecución que es lo que hicimos o lo que vamos a hacer.

Desde el WorkWith de Restaurant, cuando accedamos al botón MAP vamos a mostrar una pantalla intermedia que le permitirá al usuario seleccionar el tiempo disponible para almorzar, por ejemplo 90 minutos, y cuando le da OK el usuario se le van a mostrar en el mapa esas opciones y no todos los restaurantes, sino filtrado por el tiempo en los que el restaurante se compromete a dar el servicio.

Vamos a verlo por ejemplo con un valor mas restrictivo, 30 minutos, y ahí vemos que tenemos menos opciones. bien.

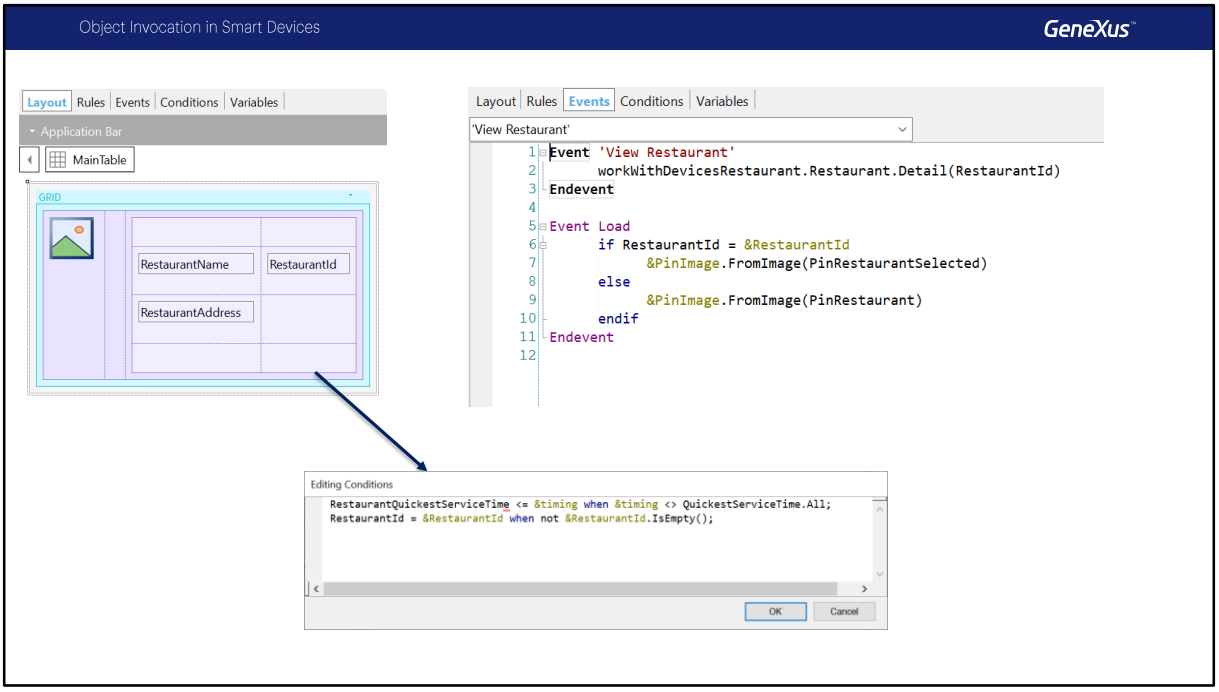
Además desde el mapa puedo ver cierta información, el nombre la dirección y si hago TAP sobre el elemento me lleva al view del detail de ese restaurante.



Bien, vamos a ver como programamos todo esto.

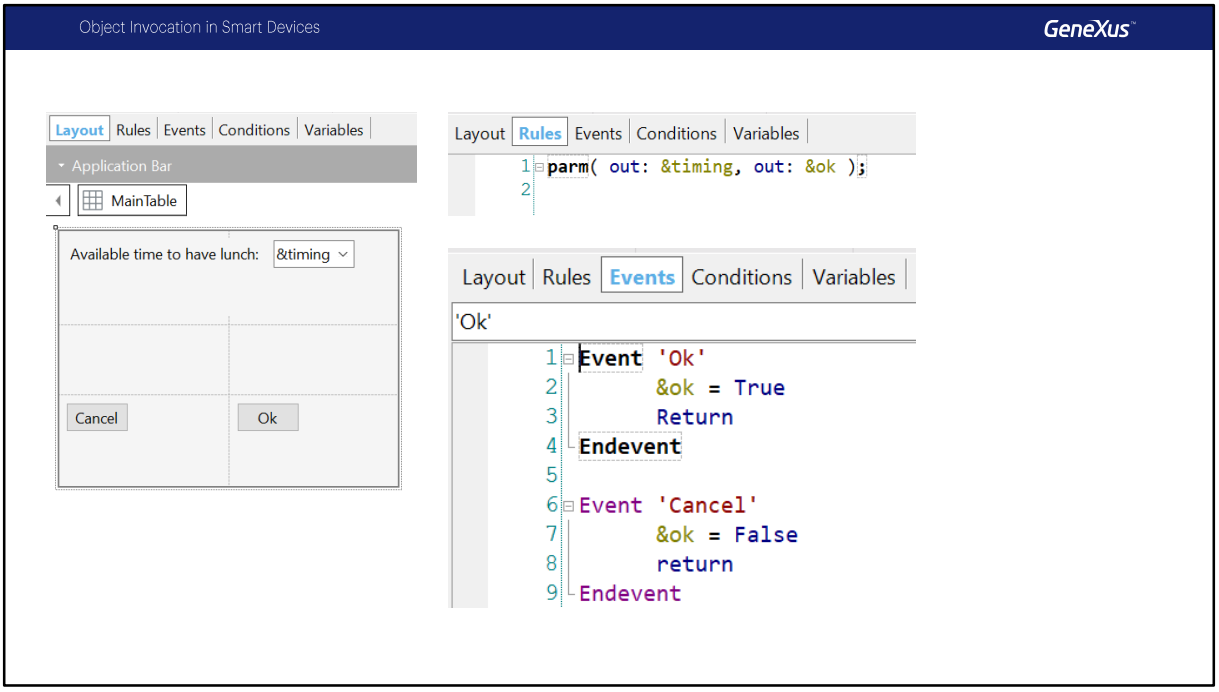
Vamos a empezar por el panel del mapa. Este panel tiene la misma grilla del list, exactamente la misma, hicimos un copy de ese elemento y lo pegamos en este layout, pero configuramos el Control Type SD MAPS, esto ya lo habíamos visto en otro video, cual es el Location Attribute, etc.

En los parámetros vamos a recibir para poder filtrar en una variable &timing el tiempo de servicio que quiere el usuario, es un dominio enumerado que tiene los valores 30,60, 90 y el 0 se usa para la opción todos. y también podemos recibir un Id de un Restaurant para mostrarlo resaltado en el mapa.

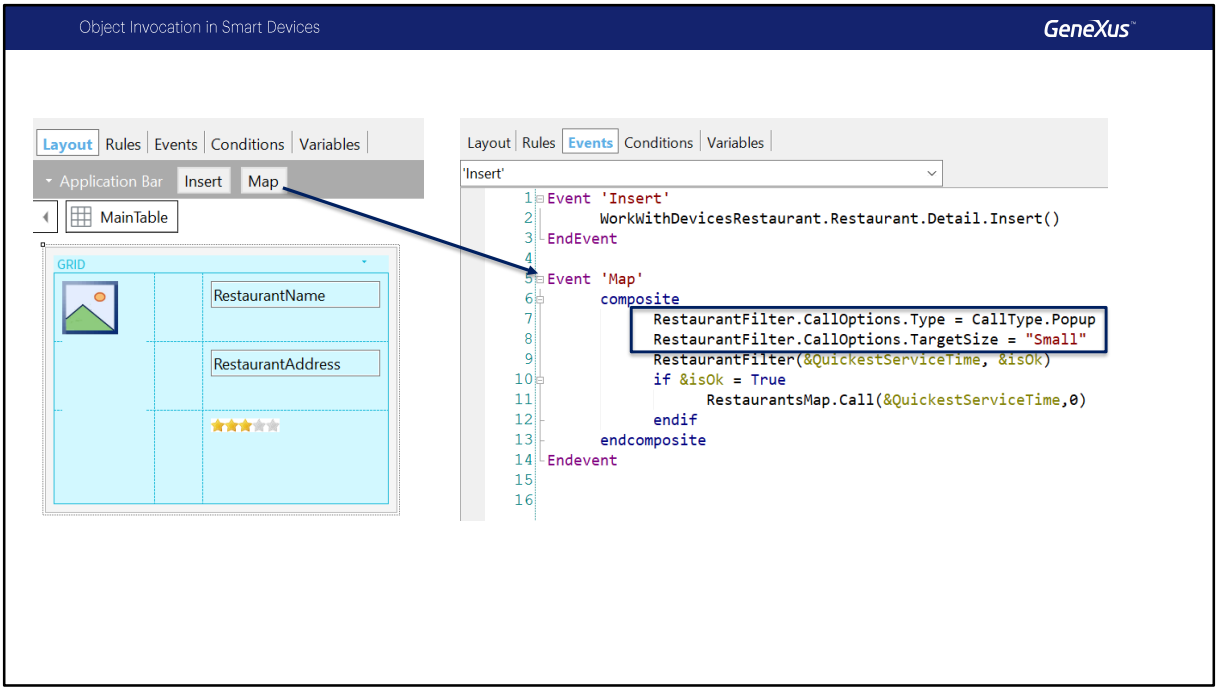


Vamos a ver los eventos, bueno, el evento View Restaurant es el que me permite cuando hago Tap sobre el elemento este sobre la tabla, que es el default Action del Grid, View Restaurant, para ir al Detail; y en el Load simplemente se cambia cuando hay un restaurante que viene indicado por parámetro, se cambia el icono y se muestra de otro color.

Por ultimo vamos a ver las Conditions de este panel, tenemos el tiempo de servicio al que se compromete el restaurante menor o igual a lo que desea el usuario, entonces si yo selecciono 90 se van a incluir los de 90, los de 60 y los de 30 obviamente, y este filtro se va a aplicar cuando la opción sea distinta de todos. Y el otro filtro es simplemente para filtrar por clave primaria cuando la variable tenga un valor que recibimos por parámetro.



Vamos a ver ahora el panel intermedio, que usamos para que el usuario seleccione el tiempo del que dispone.
el layout, de vuelta, tenemos la variable &timing, que esta basada en el domino al igual que el que recibía el mapa, y dos botones, uno para que el usuario cancele la operación y no se muestre el mapa y el OK para que retorne efectivamente el valor seleccionado.
En las Rules, este panel retorna el valor seleccionado por el usuario y una bandera &ok que retornara en True cuando presione OK o False cuando presione Cancel.

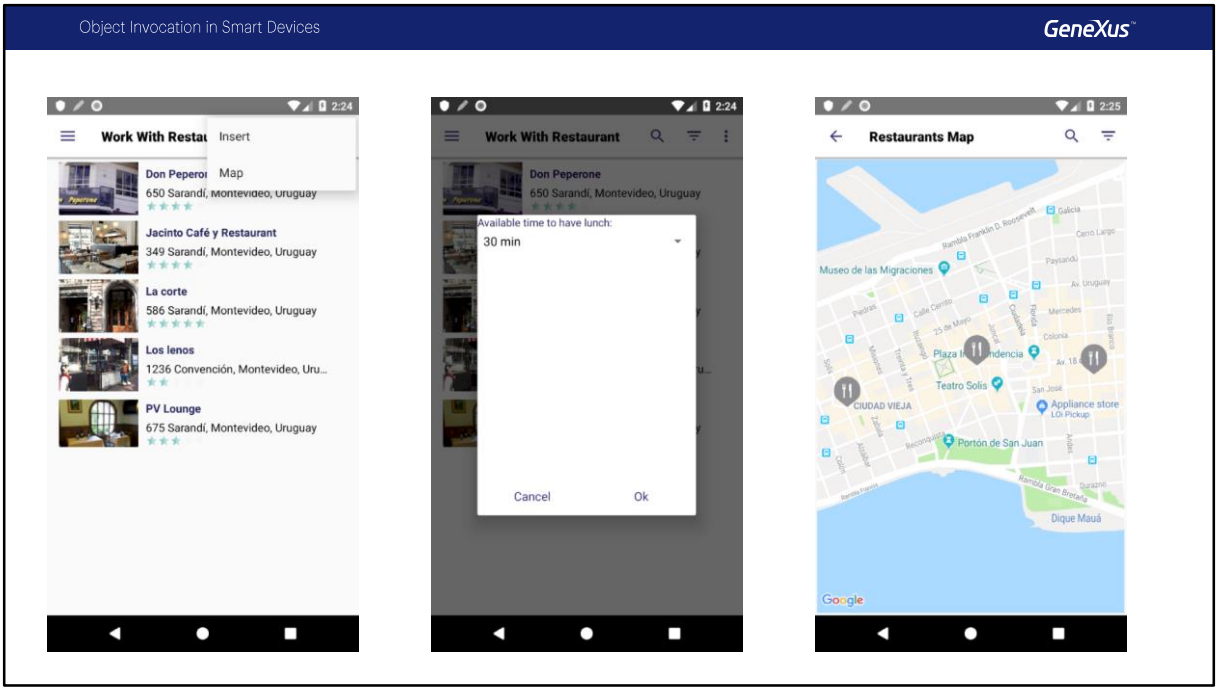


Vamos a ver ahora si el WorkWith, en el botón MAP lo que hicimos.

Llamamos al panel que muestra el filtro para que el usuario seleccione un valor, en el caso de que el usuario presione OK vimos que retornaba además la bandera en True y en ese caso vamos a llamar al mapa y vamos a pasarle la selección del usuario, si es False en ese caso no vamos a hacer nada.

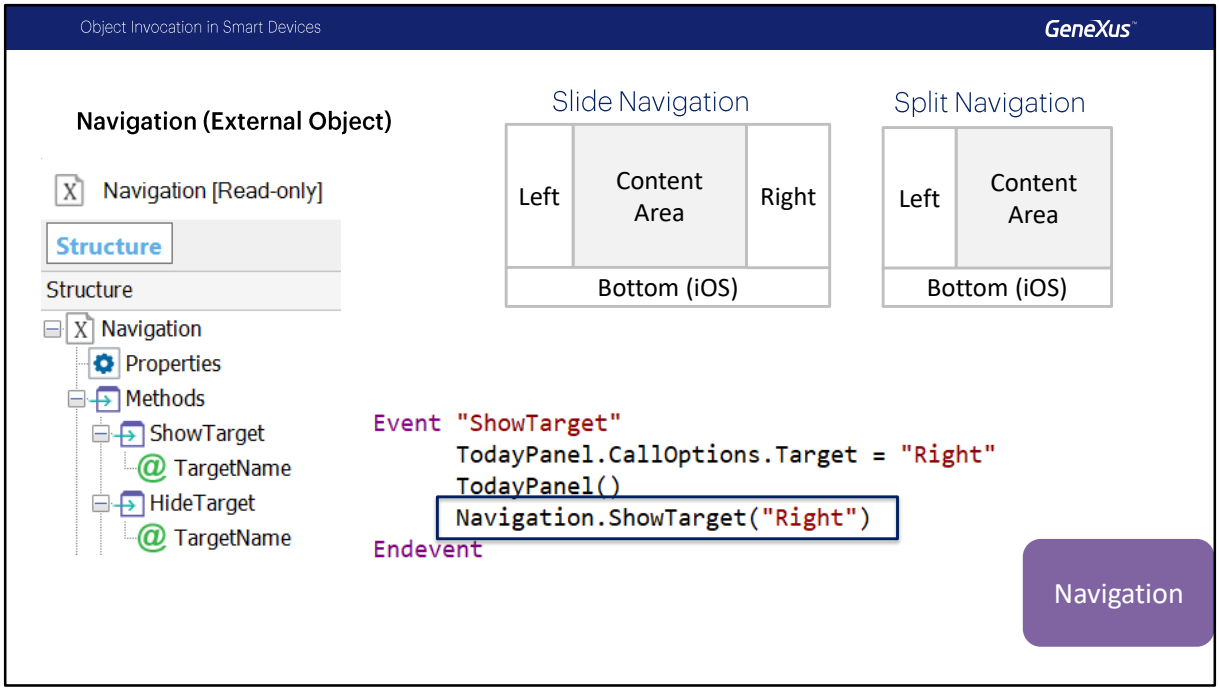
Un problema que tenemos es este panel ocupa toda la pantalla y deseamos que se vea como un PopUp.
Entonces , como podemos hacer esto?
Simplemente vamos a usar CallOptions, vamos a usar la propiedad Type , vamos a configurarla y vamos a usar el dominio enumerado CallType.PopUp.
Debajo ponemos de vuelta, el nombre del panel .CallOptions.TargetSize y para que no ocupe toda la pantalla vamos a usar el valor "small“.
Con estas dos líneas vamos a configurar como se va a llamar al panel , luego hacemos la invocación y se va a respetar lo que hayamos puesto.
En el caso del mapa si queremos que ocupe toda la pantalla.

Vamos a grabar y vamos a correr la aplicación.



Bien, vamos a ver entonces como quedo el panel , ingresamos al List de Restaurants, vamos a la opción Map, y ahí vean que el panel se ve como un PopUp, es modal: no puedo hacer en este caso Tap fuera de él, si están en iOS y usan la opción CallOut, se muestra como un PopUp no modal, en ese caso si yo hago Tap fuera de la pantalla va a volver a la pantalla de atrás.
Vamos a seleccionar para ver los registros en el mapa, y el mapa no toma la configuración de las CallOptions lo cual es correcto.

Volvamos a la presentación.



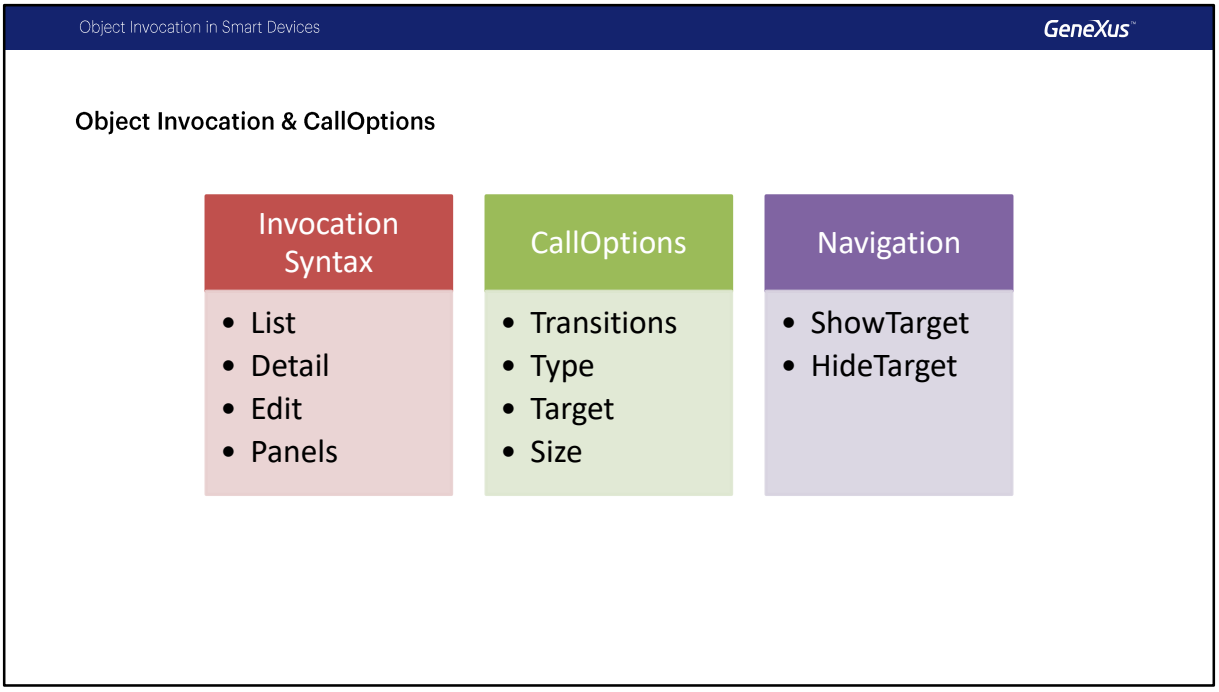
Para estilo de navegación Slide o Split , contamos con distintas regiones como “left”, “right”, y “bottom”, esto lo habíamos visto cuando estudiamos los estilos de navegación de la aplicación y como vimos en CallOptions con la propiedad Target hace un momento podíamos determinar donde queremos mostrar el panel.

El External Object Navigation, permite a los desarrolladores mostrar u ocultar contenido en forma dinámica en la región que se desee.

Si bien el objeto Navigation posee mas métodos y eventos definidos, en la imagen solo mostramos los métodos ShowTarget y HideTarget, que son los que tenemos disponibles para usar en Smart Devices, los demás Métodos y Eventos de este objeto están disponibles solo en entorno Web.

Por ejemplo, si el Navigation Style de la app es Slide y el panel central contiene un evento definido como se muestra, no sólo hay que cargar el panel en la región derecha, con la CallOption target que estamos viendo, sino que hay que desplegarla también, si no lo hacemos el panel se ejecutara pero no estará visible (a menos que estemos usando target “Content”.

Y para ello se utiliza entonces el método ShowTarget del external object Navigation.



Vamos a hacer un resumen de todo lo que vimos.

Primero vimos la sintaxis para llamar a los distintos objetos generados por el patrón WorkWith y a paneles, el pasaje de parámetros en cada caso.
También vimos las configuraciones que podemos utilizar por medio del uso de CallOptions, como Transiciones de entrada y salida, el tipo de llamada que podía ser Popup, Callout, Push que es el default y Replace . En otro video se estudiara en detalle las diferencias de Push y Replace.
Vimos también que podemos configurar el Target en donde se abrirá el objeto cuando usamos Split o Slide y Size que nos permite modificar el tamaño cuando usamos Popup o Callout.
Por ultimo vimos el uso del External Object Navigation y como podemos usarlo para mostrar u ocultar un panel en un sector determinado cuando usamos navegación de tipo Split o Slide.

Con esto terminamos el tema.

GeneXus™

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