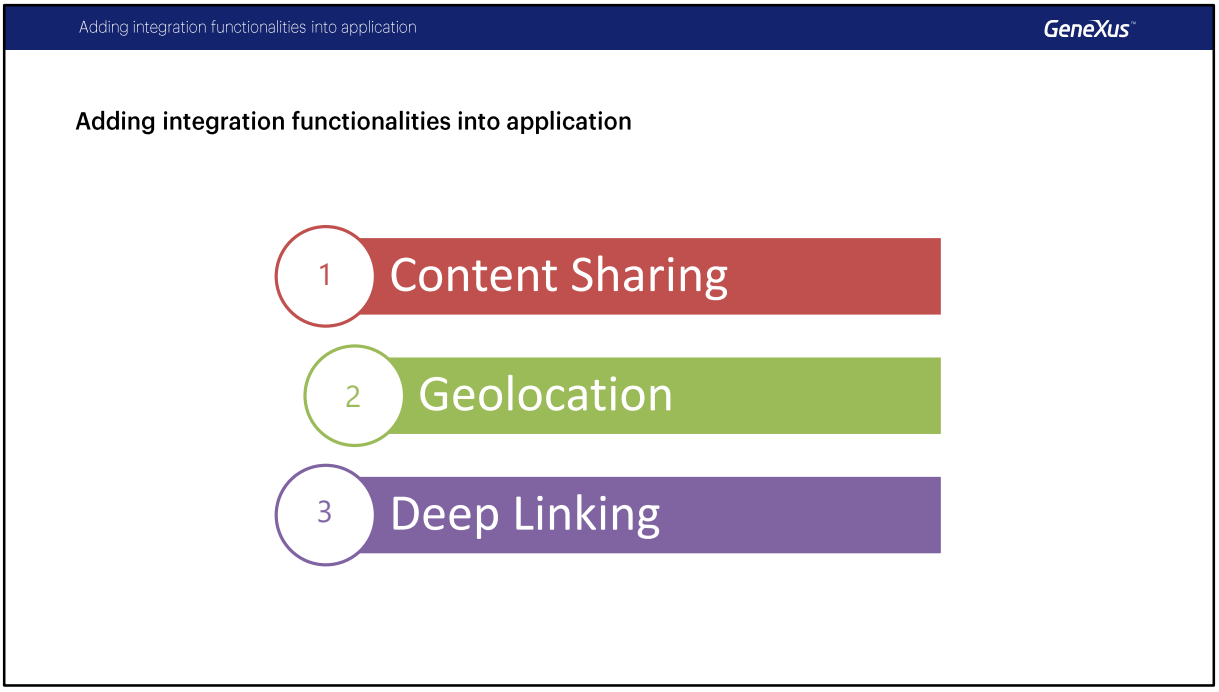

GeneXus™
The power of doing.

Adding integration functionalities into application

Integration Development

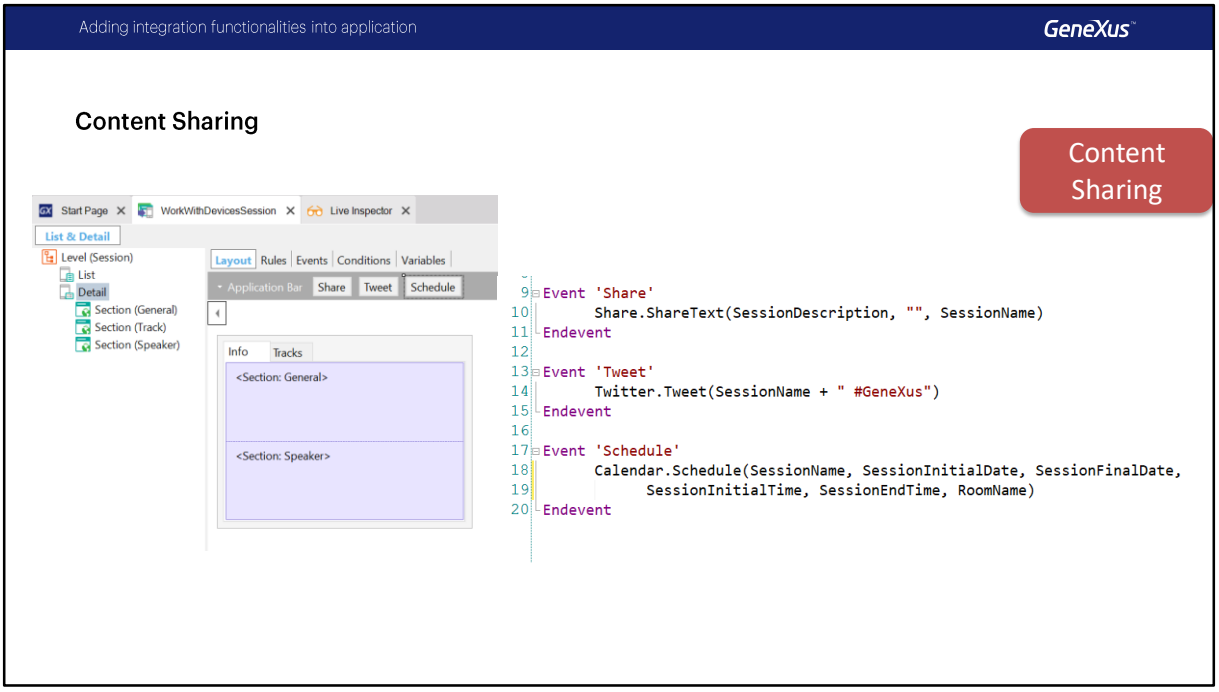
GeneXus™ 16



En este video vamos a ver como integrar nuestra aplicación haciendo uso de las APIs que nos ofrece GeneXus.
Las funcionalidades que desarrollaremos nos permitirán compartir contenido usando distintas aplicaciones y además permitiremos buscar restaurantes que se encuentren en un radio determinado de nuestra posición, por ultimo veremos como podríamos integrar nuestra aplicación SD con una aplicación Web usando Deep Linking.

Demo: Adding integration functionalities into application

Vamos a ver como hacer esto en GeneXus.



Vamos a trabajar primero sobre el objeto WorkWithDevicesSession.

En el View, agregaremos un botón en el Application Bar, vamos a poner de nombre al evento “Share”.
Vamos a definir el evento.
Lo que deseamos compartir es el titulo y la descripción de la conferencia, para eso vamos a utilizar la API Share, tiene 2 métodos uno para compartir texto y otro para compartir imágenes, su utilización es muy simple, escribimos Share.Text, este método recibe 3 parámetros: el Texto, una URL que es opcional y un titulo.
Como Texto compartiremos la descripción de la conferencia SessionDescription, en URL no ingresaremos nada y en titulo pondremos el nombre de la conferencia, SessionName.

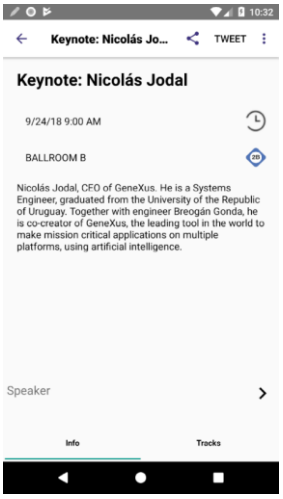
Ahora agregaremos otro botón mas, para permitir compartir la información por Twitter, el nombre del evento vamos a ingresar Tweet.
Vamos al evento.
En este caso nos interesa inicializar el tweet con el nombre de la sesión y vamos a agregar también un hashtag #GeneXus al final.
vamos a utilizar la API Twitter y el método Tweet que solo recibe un parámetro con el texto que queremos.
Vamos a escribir entonces Twitter.Tweet, el texto que deseamos, SessionName y vamos a

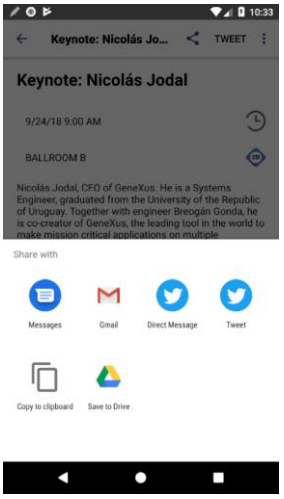
concatenarle al final un Hashtag “ #GeneXus". OK.

Por ultimo, vamos a brindar también la posibilidad de agregar una conferencia a la agenda del usuario. Para esto agregaremos un ultimo botón, vamos a ponerle de nombre Schedule. En el evento vamos a utilizar la API Calendar y el método Schedule. Vamos a ver esta Api, recibe varios parámetros.
Escribimos entonces Calendar.Schedule, en titulo pondremos SessionName, en StartDate utilizaremos SessionInitialDate, en EndDate utilizamos SessionFinalDate. En hora de inicio pondremos SessionInitialTime y en horario de finalización SessionEndTime, por ultimo podemos indicar el lugar, aquí usaremos el nombre de la sala RoomName.

Content Sharing: Share Text

Content Sharing



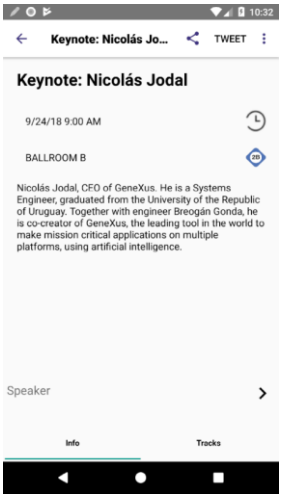


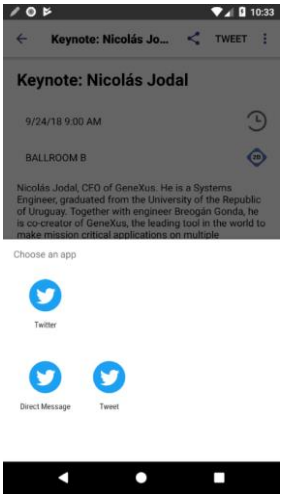
Bien, vamos a ver y a correr nuestra aplicación.

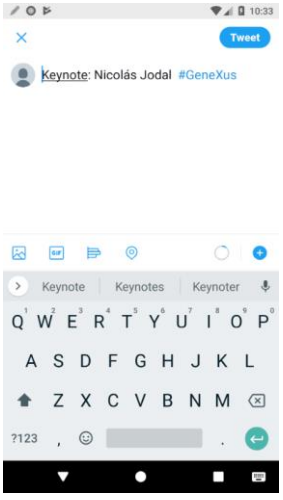
Bien, vamos al emulador.
Entramos a Sessions, vamos a entrar a la primera session y vamos a probar Share, fijense que cambio el icono, automaticamete lo pone GeneXus.
y aca nos permite compartir texto en cualquiera de las aplicaciones que manejan ese tipo de contenido.

Content Sharing

Content Sharing: Twitter

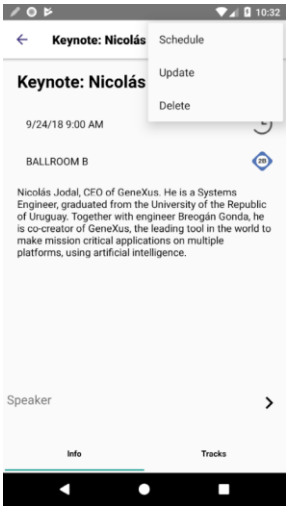


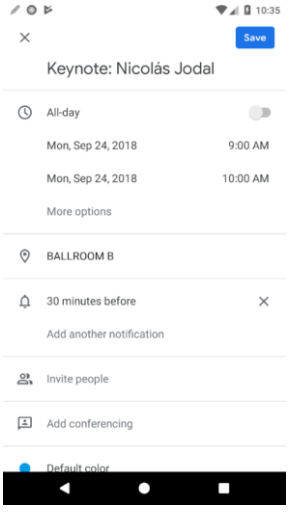




Tenemos la opción Tweet. En este caso el dialogo permite seleccionar si deseamos enviar un mensaje directo o simplemente escribir un tweet, seleccionamos esta ultima opción, se inicia un tweet con la información que ingresamos nosotros. Volvamos. vamos a borrarlo.

Content Sharing: Schedule

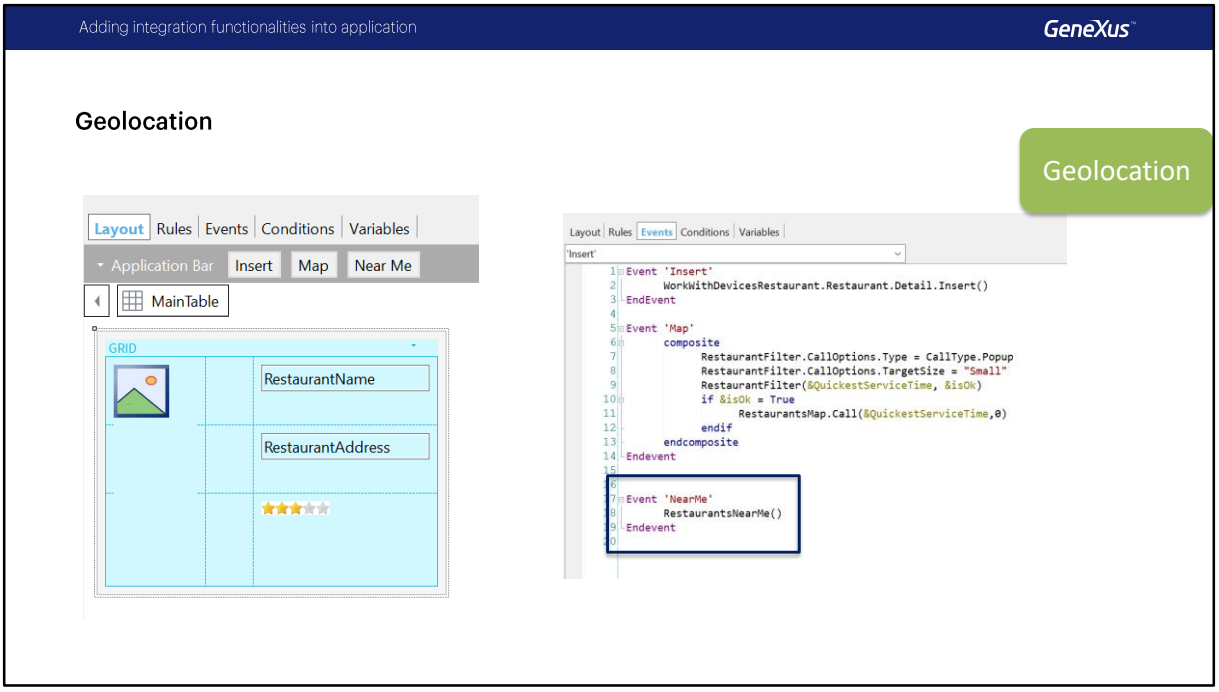




Content Sharing

Por ultimo, veamos la opción de agregar la conferencia a la agenda. Schedule. se abre la aplicación predeterminada y se crea el evento con los datos ya inicializados.

Bien, volvamos a GeneXus.

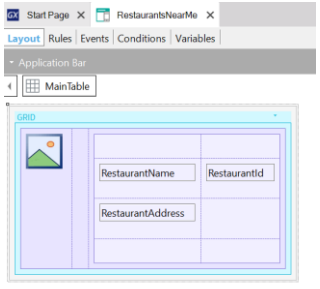


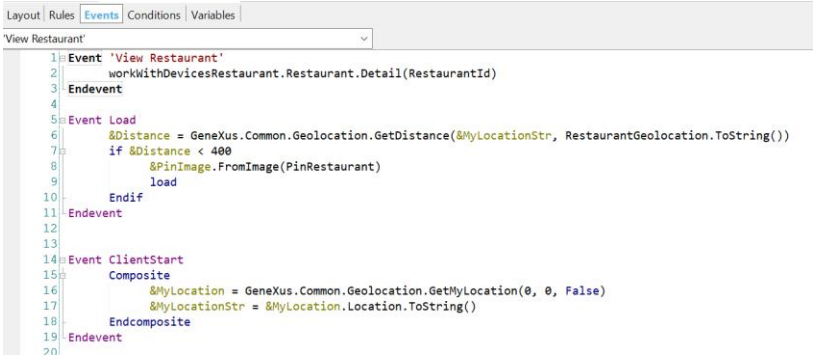
Ahora vamos a modificar el List de Restaurantes, lo que vamos a hacer es agregar una opción para que nos muestre el mapa con los restaurantes, pero solo con los mas cercanos, supongamos que a menos de 400 metros.

Para esto, vamos a usar una copia de RestaurantsMap, yo ya cree una copia de este panel y la nombre RestaurantsNearMe, este panel no recibe parámetros porque no los necesitaremos.
y en el Layout además agregue un botón "Near Me" para llamarlo.

Geolocation

Geolocation





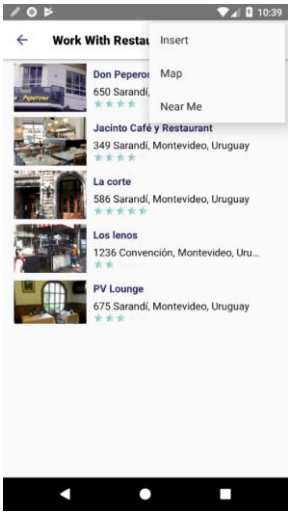
Vamos a ver el panel RestaurantsNearMe.
En el evento ClientStart vamos a capturar la ubicación actual del dispositivo.
vamos al evento ClientStart y vamos a usar la API Geolocation con un método GetMyLocation, que retorna un tipo de dato estructurado, yo ya cree &Mylocation, la variable basada en ese tipo, y escribimos GeoLocation.GetMyLocation, y acá recibe varios parámetros, vamos a dejar todos en 0, tenemos la precisión , el timeout y una bandera, acá les mostraba &Mylocation de tipo Geolocation.

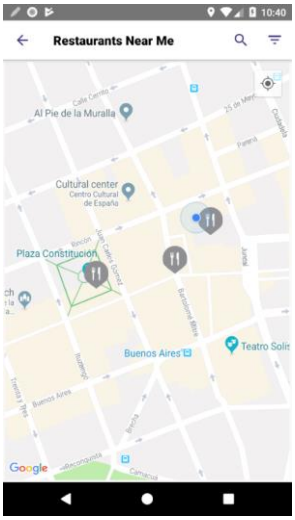
Vamos a guardar además la ubicación en un string, o sea el valor de latitud y longitud, entonces en el string &Mylocation.Location.Tostring() para almacenar ese valor. como tenemos mas de una línea vamos a escribir el bloque composite, endcomposite.

y ahora en el evento Load, vamos a usar nuevamente esta API Geolocation, pero esta vez usaremos el método GetDistance, que retorna una distancia que es un numérico, entonces escribimos Geolocation.GetDistance, este método recibe dos strings con las dos ubicaciones,
entonces Vamos a pasarle &MyLocationStr y la ubicación del restaurante, ResaurantGeoLocation, y esto nos va a dar, vamos a convertirlo a String, esto nos va a dar entonces la distancia entre esos dos puntos medida en metros.
entonces antes de cargar, vamos a preguntar si la distancia es menor a 400 , ahí si

asignamos la imagen y hacemos el load y eso seria todo.

Geolocation





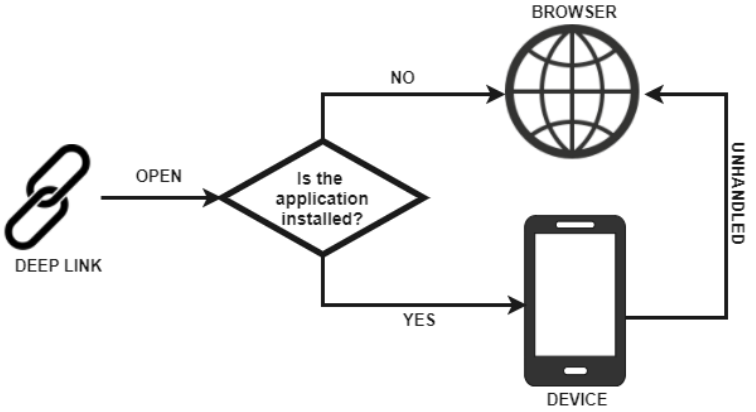
Geolocation

Bien, vamos a verlo en ejecución.
vamos al emulador.
Accedamos a Restaurants, vamos a la opción Map, primero para que veamos los restaurantes que tenemos , vamos a usar "todos", así vemos todos los restaurantes. vean que tenemos 5. volvamos.
y Ahora si entremos a la opción, Near Me, ahí recupero nuestra ubicación, fíjense que la ubicación esta en azul, esa es nuestra ubicación, y ahora solamente tenemos 3 restaurantes cercanos, bien, por ahora eso es todo.

Volvamos a la presentación.

Deep Linking

Deep Linking

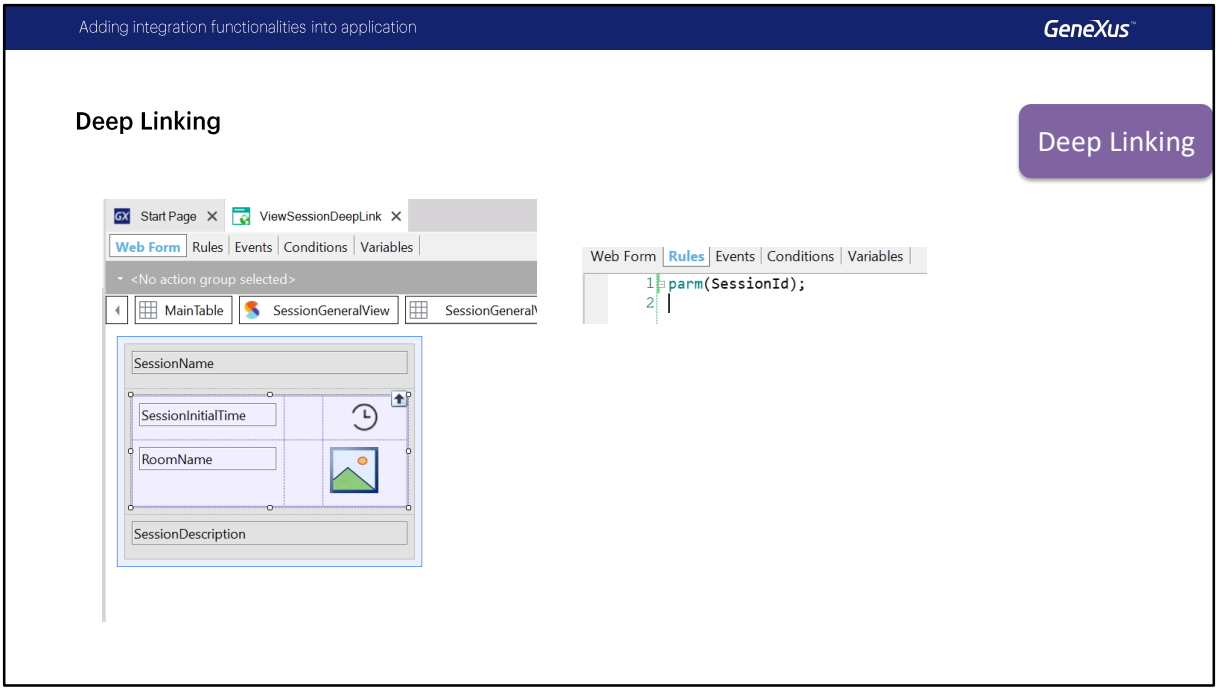


Vamos a ver ahora Deep Linking.

Esta característica nos permite, mediante el uso de una API especial DeepLink, establecer que determinados Links o URLs van a ser manejadas directamente por una app y no por el navegador web.
Automáticamente se va a chequear si existe esa APP en el dispositivo, en ese caso se va a ofrecer la posibilidad de abrirla y en caso de que no tengamos la app podemos navegar el sitio web normalmente. También es posible que ese link no sea debidamente manejado por la aplicación y en ese caso también nos lleva al navegador web.

En nuestro caso, supongamos que recibimos un mail con un link sobre una conferencia, cuando abramos el link,
en lugar de abrir el navegador web se debería abrir la aplicación y mostrar directamente la información de la conferencia, en caso de que no tuviéramos la aplicación el link será abierto directamente por el navegador del dispositivo.

Veamos como podemos implementarlo en nuestra aplicación.



Para poder utilizar esta característica necesitamos varias cosas.

Primero vamos a necesitar un panel web, que va a recibir por parámetro el atributo SessionId.

Deep Linking

Deep Linking

```
Event DeepLink.Handle( &URL, &IsHandled )
    Composite
        if &URL.Trim().ToLower().StartsWith("https://trialapps3.genexus.com/Idd7b6bb1b5ade0f353170746540991386/")
            and &URL.ToLower().Contains("viewsessiondeeplink")
            &Index = &URL.indexof( "?" )+1
            &Query = &URL.Substring( &Index)
            &SessionId = &Query.Trim().ToNumeric()
            &IsHandled = True
            ViewSessionSD(&SessionId)
        endIf
    EndComposite
EndEvent
```

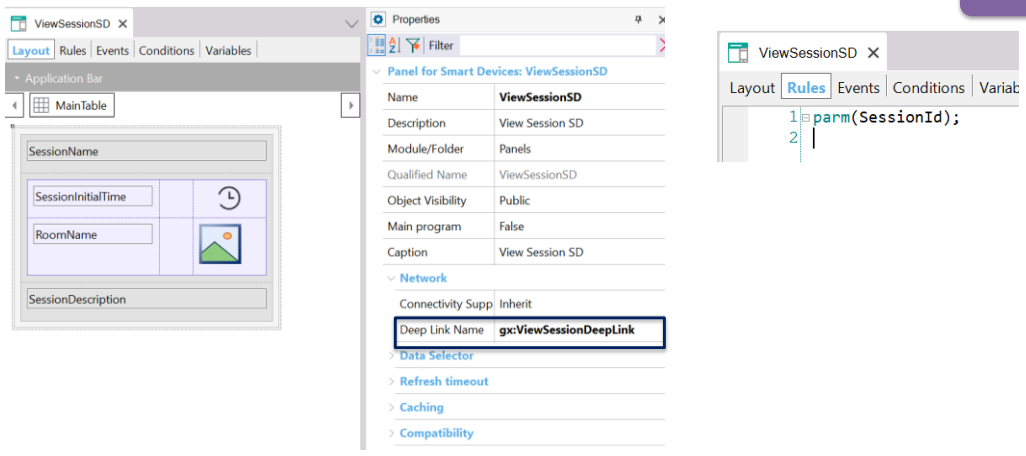
Main object properties

Application Title	**Event Day**
Application Shortcuts	(none)
Platform Overrides	
Default Layout Orientation	Default
Ads Provider	None
Deep Link Base URL	**https://trialapps3.genexus.com/Idd7b6bb1b5ade0f353170746540991386/**

Luego vamos a necesitar programar en el objeto main de nuestra App un evento especial, que utilizara la API DeppLink y el evento Handle, además de establecer la propiedad Deep Link Base URL con la URL de la aplicación web.

Deep Linking

Deep Linking



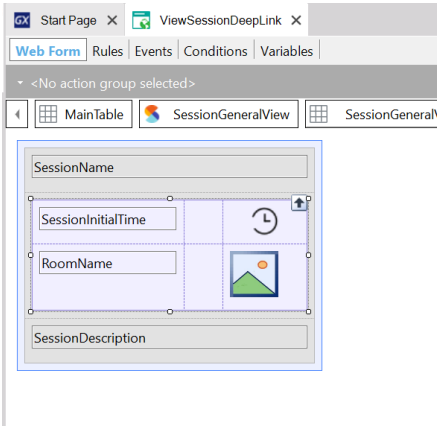
Por ultimo, vamos a crear un panel SD, va a ser igual al nodo Detail del WorkWithDevicesSession, este panel recibe como parámetro también a SessionId y posee la propiedad Deep Link Name donde es necesario especificar el nombre del panel web.

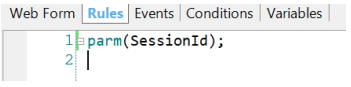
Demo: Deep Linking

Vamos a GeneXus.

Deep Linking: Web Panel ViewSessionDeepLink

Deep Linking

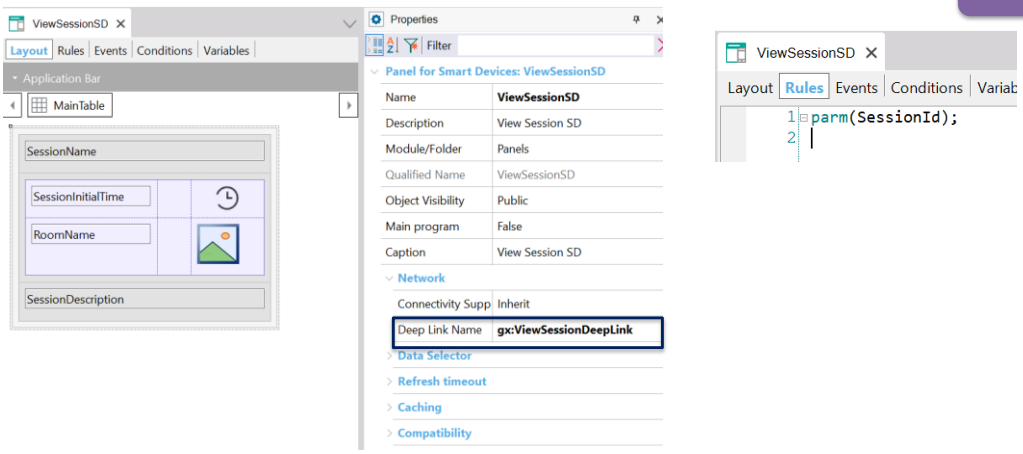




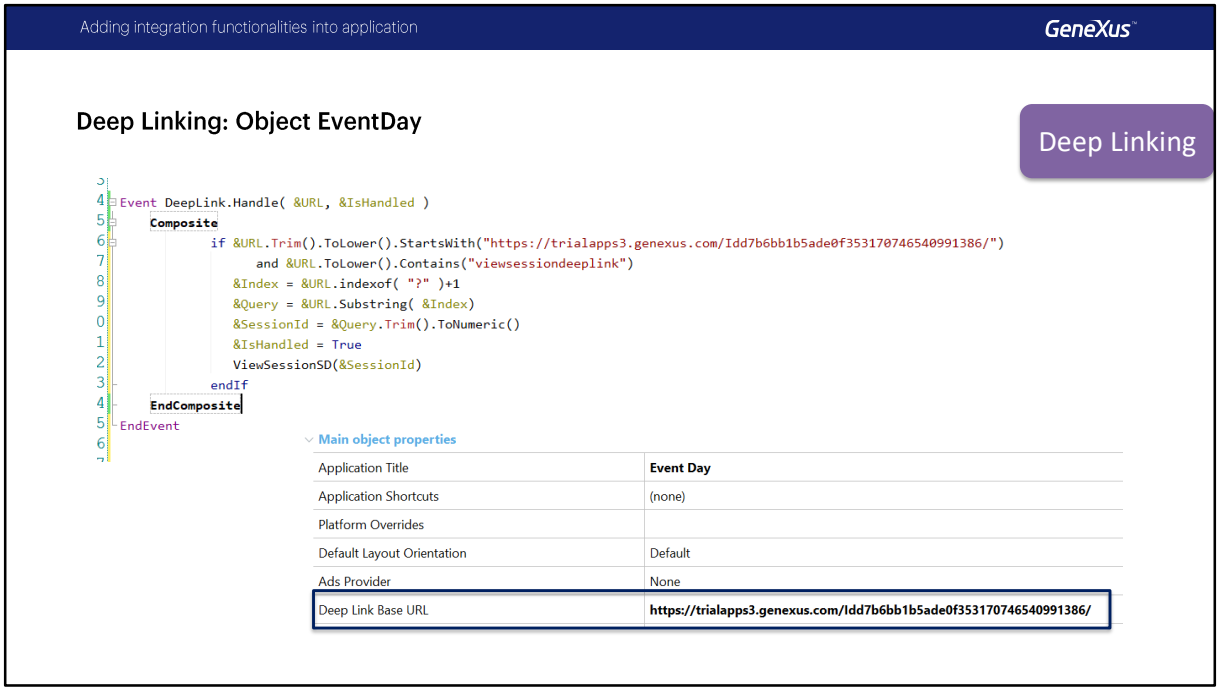
Yo tengo programado un web panel, ViewSessionDeepLink, que no tiene ninguna propiedad especial, en los parámetros recibe SessionId como habíamos dicho, y el Layout es igual al nodo Detail del WorkWithDeviceSsession.

Deep Linking: SD Panel ViewSessionSD

Deep Linking

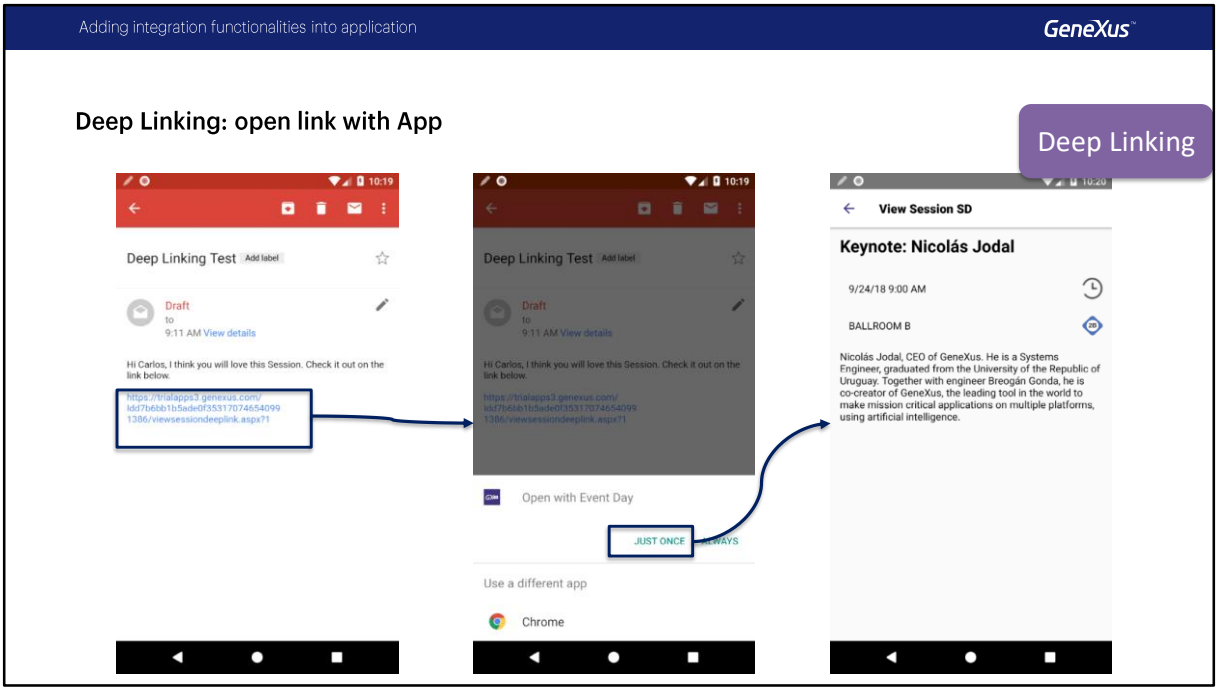


Por otro lado tengo un panel SD, un Panel for Smart Devices, que también recibe SessionId como parámetro, el Layout es exactamente igual al web panel, pero acá especificamos en una de las propiedades, Deep Link Name , el nombre del web panel, ViewSessionDeepLink. el prefijo se coloca automáticamente cuando seleccionamos el objeto, prefijo gx:



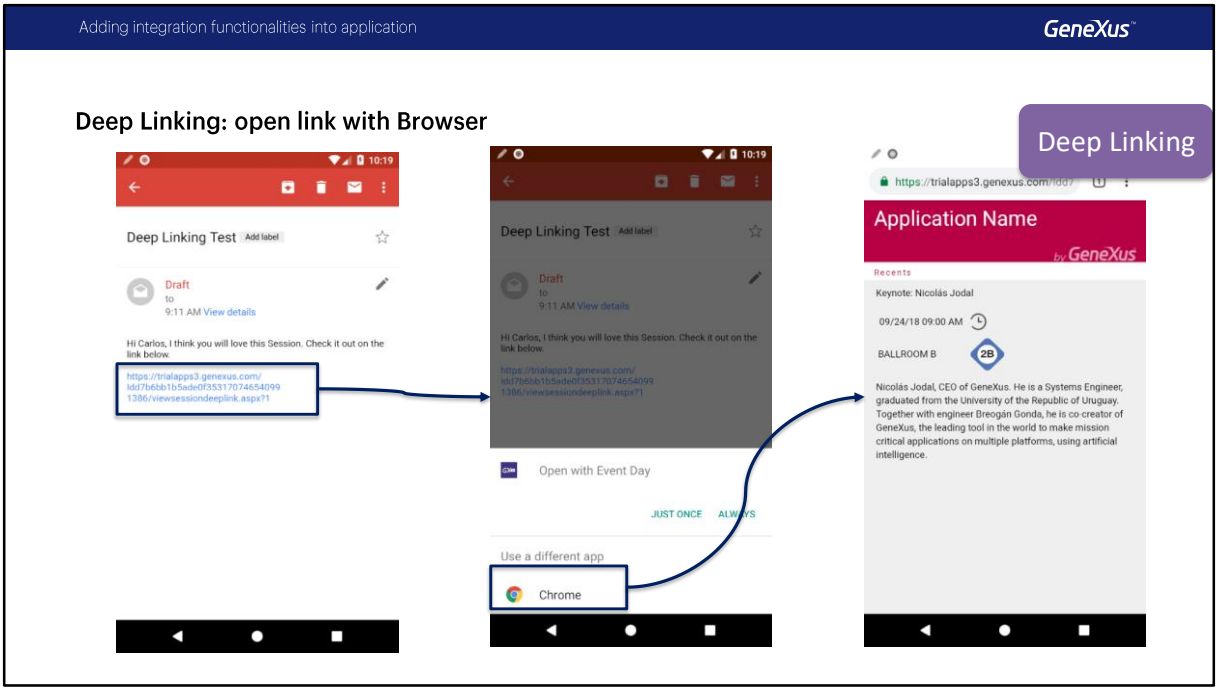
Por otro lado tenemos que programar en el objeto EventDay, el objeto main de la aplicación.
En las propiedades tenemos que especificar Deep Link Base URL con toda la URL de la aplicación web, esto lo podemos obtener de las propiedades del modelo si no lo conocemos.

Por otro lado, en los eventos tenemos que programar el manejo de el link para esto vamos a usar DeepLink, la API esta que mencionamos con el evento Handle.
vamos a ver esta API, fíjense que no tiene propiedades, no tiene métodos, solamente tiene un evento que recibe una URL y una bandera que indica luego si esa URL fue manejada o no.
Dentro del evento, como ejecuta del lado del cliente, hay que usar composite,
y acá primero vamos a fijarnos si la URL comienza con nuestra URL, con la URL de la aplicación, y además, si la URL contiene además en este caso "viewsessiondeeplink" que es el nombre del web panel.
Acá podríamos haber hecho un Case en el caso de que sean varios paneles SD.
Y luego vamos a parsear los parámetros para obtener el SessionId, tenemos que setear la variable &IsHandled en True sino nos llevaría al navegador, y luego vamos a ir con el SessionId a ViewSessionSD.



Vamos a verlo en ejecución, vamos al emulador.

Yo tengo un mail con el link a la sesión numero 1, la conferencia numero 1.
Vamos a hacer click en el link, fíjense que me ofrece abrir la aplicación EventDay, vamos a usar la opción Just Once así lo podemos usar otra vez, ahí se carga y se abre el panel ViewSessionSD y nos muestra la conferencia numero 1 que era la que estaba especificada en el parámetro.



Vamos a volver y vamos a ir al link de vuelta. Así vemos que pasa cuando lo abrimos con el navegador, elijamos chrome entonces, y aquí se abre el web panel que tiene la misma información, pero ahora nos muestra el contenido web.

Eso es todo por ahora y con esto terminamos el tema.

GeneXus™

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