



Semantic Domains & Smart Devices APIs

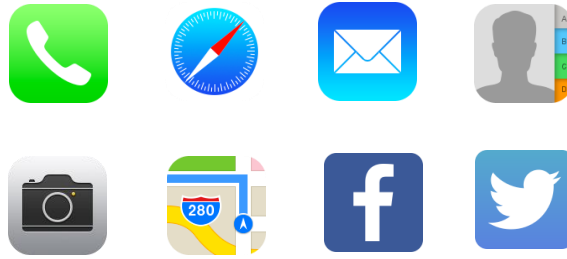
Integration Development

GeneXus™ 16

- 1 Semantic Domains
- 2 Smart Devices APIs

En este video vamos a estudiar dos formas de integrar nuestras aplicaciones con las características de los dispositivos.

Primero veremos los dominios semánticos que nos proveerán de algunas formas de integración básicas, y luego veremos algunas de las APIs mas comunes que nos Brinda GeneXus al momento de utilizar los dispositivos inteligentes.

Semantic DomainsSemantic
Domains

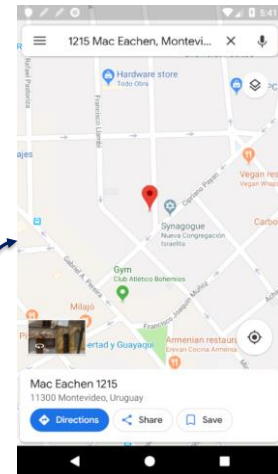
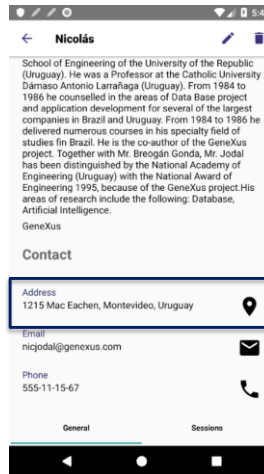
Las aplicaciones requieren integrarse con las características que nos ofrecen los dispositivos inteligentes, por ejemplo, deberán poder realizar llamadas telefónicas, abrir contenido web, enviar mensajes a través de los programas de chat o de mail instalados, acceder y usar la aplicación de contactos, la cámara, el GPS y las aplicaciones de mapas, así como interactuar con aplicaciones como Facebook o Twitter.

Algunas de estas funcionalidades se implementan automáticamente a través de los dominios semánticos o tipos de datos con semántica, a continuación veremos los que tenemos disponibles.

Semantic Domains: Address

Semantic
Domains

Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Name
SpeakerFullName	FullName
SpeakerImage	Image
SpeakerCVMini	Bio
CountryId	Id
CountryName	Name
CompanyId	Id
CompanyName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus
SpeakerIsKeynote	Boolean



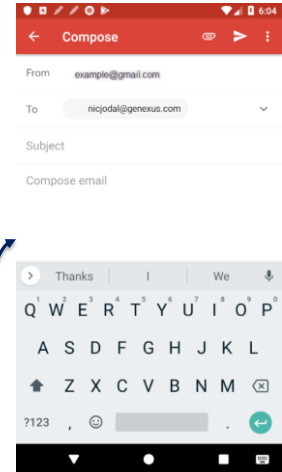
Por ejemplo el dominio Address se utiliza para almacenar direcciones, en tiempo de ejecución, si el atributo o variable definido sobre este dominio es Read-Only, entonces al hacer Tap nos lleva a la aplicación predeterminada de mapas que tenemos en nuestro dispositivo y se posiciona en esa dirección.

Aquí en el ejemplo vemos el View de los Oradores y al hacer Tap sobre la dirección, como el atributo está basado en el dominio Address nos presenta el mapa con Google Maps.

Semantic Domains: Email



Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Name
SpeakerFullName	FullName
SpeakerImage	Image
SpeakerCVMini	Bio
CountryId	Id
CountryName	Name
CompanyId	Id
CompanyName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus
SpeakerIsKeynote	Boolean



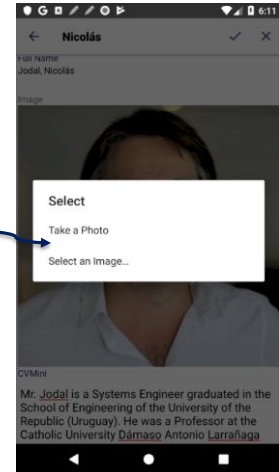
El dominio Email, es similar pero para almacenar direcciones de correo electrónico, y también cuando el atributo o variable es Read-Only, al hacer Tap se abrirá la aplicación predeterminada para enviar mensajes de correo electrónico.

En la misma pantalla de los Oradores, cuando hacemos Tap sobre el Email se abre la aplicación de correo, en este caso Gmail.

Semantic Domains: Image



Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Name
SpeakerFullName	FullName
SpeakerImage	Image
SpeakerCVMini	Bio
CountryId	Id
CountryName	Name
CompanyId	Id
CompanyName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus
SpeakerIsKeynote	Boolean



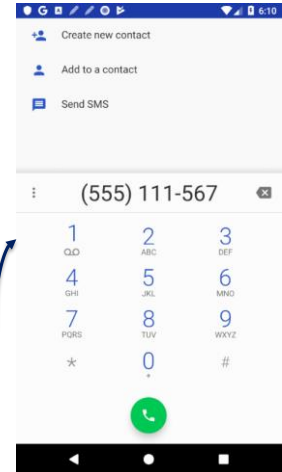
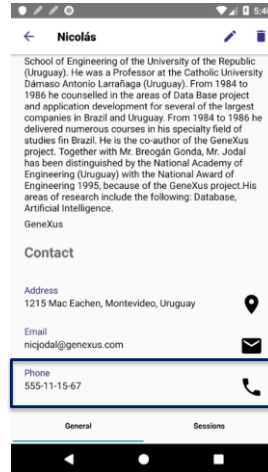
El dominio Image, se utiliza para almacenar imágenes, en modo Edición se preguntara al usuario si desea tomar una foto con la cámara del dispositivo o si desea seleccionar una imagen de la galería (se abre en forma automática la app de galería predeterminada). En modo Read-Only se muestra la imagen.

Acá vemos la edición de un Orador y cuando hacemos Tap sobre la imagen nos permite cambiarla usando la cámara o la galería.

Semantic Domains: Phone



Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Name
SpeakerFullName	FullName
SpeakerImage	Image
SpeakerCVMini	Bio
CountryId	Id
CountryName	Name
CompanyId	Id
CompanyName	Name
SpeakerPhone	Phone, GeneXus
SpeakerAddress	Address, GeneXus
SpeakerEmail	Email, GeneXus
SpeakerIsKeynote	Boolean



Semantic
Domains

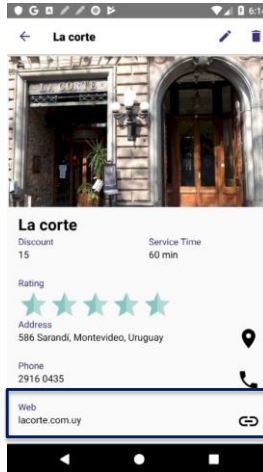
El dominio Phone se utiliza para almacenar números telefónicos. Cuando el atributo o variable es Read-Only al hacer Tap se abrirá la aplicación de llamada telefónica predeterminada.

Nuevamente, en el view de un Orador, si hacemos Tap sobre el teléfono se abrirá la aplicación de llamadas predeterminada.

Semantic Domains: Url



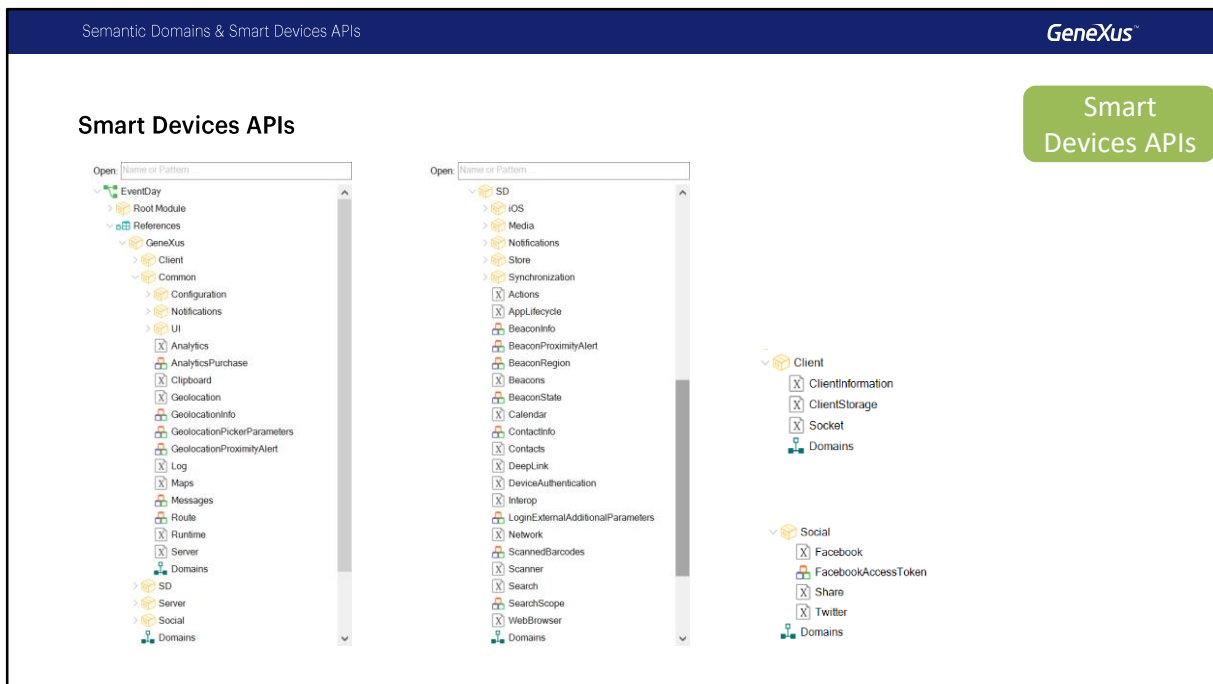
Name	Type
Restaurant	Restaurant
RestaurantId	Id
RestaurantName	Name
RestaurantImage	Image
RestaurantAddress	Address, GeneXus
RestaurantGeolocation	Geolocation, GeneXus
RestaurantPhone	Phone, GeneXus
RestaurantWeb	Url, GeneXus
RestaurantDiscount	Discount
RestaurantRating	Numeric(1.0)
RestaurantQuickestServiceTime	QuickestServiceTime



Semantic
Domains

Por ultimo tenemos el dominio Url que se utiliza para representar direcciones Web, igual que las anteriores, cuando es Read-Only, al hacer Tap se abrirá el navegador predeterminado.

En el ejemplo vemos el view de Restaurants y al hacer Tap sobre la dirección Web se abrirá el navegador predeterminado.



Por otro lado, GeneXus nos proporciona una serie de APIs, las cuales podremos utilizar para interactuar con distintas características del dispositivo o aplicaciones.

Dentro del módulo GeneXus bajo el nodo References encontraremos todas las APIs, SDTs, y Procesos que éstas requieren.

Todos estos módulos se importan cuando creamos Base de Conocimientos.

Si echamos un vistazo rápido, vemos que tenemos APIs especiales para SD, otras especiales para Web, y otras que valen para una y otra plataforma como las del modulo Common.

Vamos a ver algunas de ellas.

ClientInformation API

Structure	
Structure	Type
ClientInformation [Read-only]	
Properties	
Id	VarChar(128)
OSName	VarChar(40)
OSVersion	VarChar(40)
NetworkID	VarChar(128)
Language	Character(20)
DeviceType	SmartDeviceType, GeneX...
PlatformName	VarChar(128)
AppVersionCode	VarChar(40)
AppVersionName	VarChar(40)
Methods	
Events	

```

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 sub modulo Client tenemos la API ClientInformation, esta Interfaz es muy utilizada para obtener propiedades del dispositivo como el ID único, el Sistema Operativo y la versión o el tipo de dispositivo.

Nosotros la utilizamos por ejemplo para guardar las sesiones favoritas del usuario, como aun no tenemos registrado al usuario utilizamos el ID del dispositivo.

Interop API

The screenshot displays the GeneXus IDE interface for the Interop API. On the left, the 'Structure' pane shows the 'Interop [Read-only]' project with a tree view of its components. The 'Methods' section is expanded, listing various methods and their return types. On the right, the 'Events' pane shows three event definitions: 'SendMessage', 'Play Video', and 'Call'. Each event is associated with a specific icon (envelope, clapperboard, and phone respectively) and contains a sequence of actions using the Interop API methods.

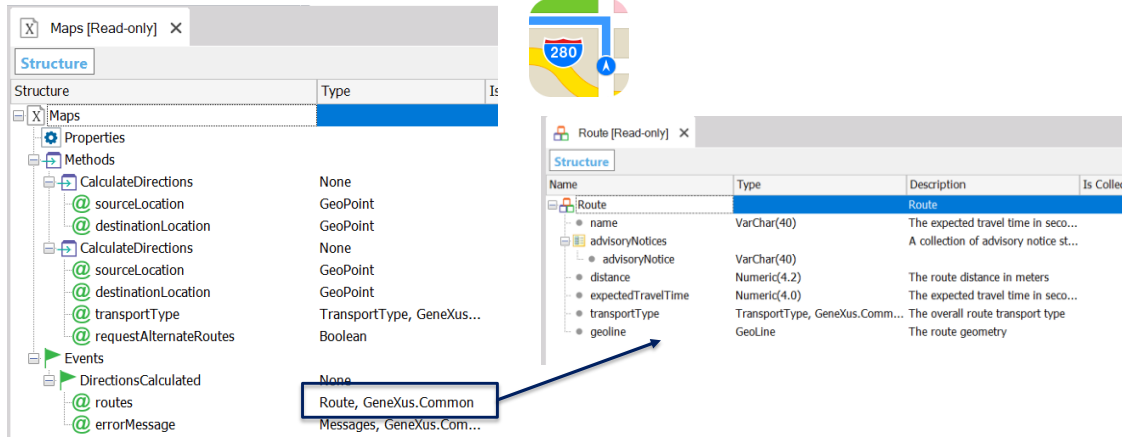
Structure	Type
Interop	
Properties	
ApplicationState	ApplicationState, GeneXus
Methods	
SendMessage	None
PlayVideo	None
PlayAudio	None
PlaceCall	None
SendEmail	None
SendEmailAdvanced	None
SendSMS	None
Msg	None
Confirm	Boolean
OpenInBrowser	None
CanOpen	Boolean
Open	None
ClearCache	None
IOSSetBadgeNumber	None
SetBadgeNumber	None
IOSSetBadgeTextToTabIndex	None
IOSSetSelectedTabIndex	None
@ tabIndex	Numeric(4.0)
Events	

```

1 Event 'SendMessage'
2   Composite
3     Interop.Confirm('Are you sure you want to Send a Message?')
4     Interop.SendMessage('This is a Message', '1234-5678')
5   Endcomposite
6 Endevent
7
8 Event 'Play Video'
9   Composite
10    Msg('Starting video playback')
11    Interop.PlayVideo('https://www.youtube.com/watch?v=UfvECtDetwY')
12  Endcomposite
13 Endevent
14
15 Event 'Call'
16   Interop.PlaceCall('1234-5678')
17 Endevent
  
```

Otra API muy utilizada es Interop, dentro del sub modulo SD, que nos ofrece toda una serie de Métodos para interactuar con funciones comunes como SendMessage para enviar un mensaje de texto, PlayVideo o PlayAudio para reproducir archivos multimedia, PlaceCall para establecer una llamada, SendEmail para enviar un mensaje de correo, Msg para mostrar un mensaje al usuario, Confirm para pedir confirmación al usuario antes de ejecutar una acción, etc.

Maps API



The screenshot displays the GeneXus IDE interface. On the left, the 'Maps [Read-only]' window shows the 'Structure' tab with a tree view of the Maps API. The tree includes 'Properties', 'Methods', and 'Events'. The 'Methods' section is expanded, showing 'CalculateDirections' and 'CalculateDirections' (both with '@' icons). The 'Events' section is also expanded, showing 'DirectionsCalculated' and 'errorMessage' (both with '@' icons). The 'Type' column for these methods and events is 'None'. The 'Is' column for these methods and events is 'Route, GeneXus.Common'. A blue box highlights the 'Route, GeneXus.Common' entry in the 'Is' column, with an arrow pointing to the 'Route' object structure on the right.

On the right, the 'Route [Read-only]' window shows the 'Structure' tab with a table view of the Route object structure. The table has columns: Name, Type, Description, and Is Collection. The 'Route' object is expanded, showing its properties and methods.

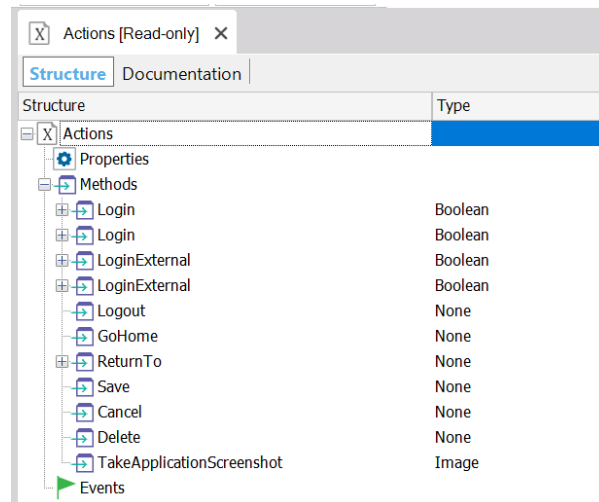
Name	Type	Description	Is Collection
Route		Route	
name	VarChar(40)	The expected travel time in seco...	
advisoryNotices		A collection of advisory notice st...	
advisoryNotice	VarChar(40)		
distance	Numeric(4,2)	The route distance in meters	
expectedTravelTime	Numeric(4,0)	The expected travel time in seco...	
transportType	TransportType, GeneXus.Comm...	The overall route transport type	
geoline	GeoLine	The route geometry	

En el sub modulo Common tenemos a Maps API, esta se puede utilizar en Web y en SD.

Nos permite calcular caminos y obtener las rutas entre dos puntos.

Muchas de estas APIs necesitan datos estructurados, por lo que van a ver que utilizan SDTs que ya están definidos, por ejemplo Route del modulo GeneXus.Common

SD Actions API



Structure	Type
Actions	
Properties	
Methods	
Login	Boolean
Login	Boolean
LoginExternal	Boolean
LoginExternal	Boolean
Logout	None
GoHome	None
ReturnTo	None
Save	None
Cancel	None
Delete	None
TakeApplicationScreenshot	Image
Events	

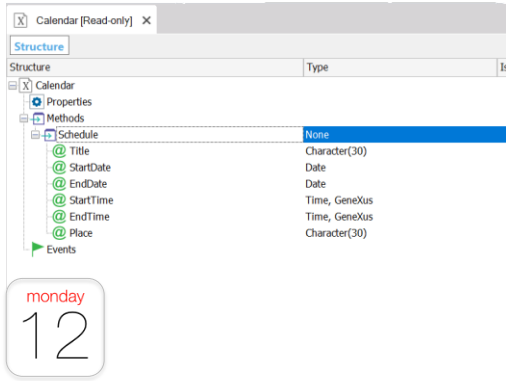
Smart
Devices APIs

La API Actions del modulo SD nos provee métodos estándar para:

- Llamar al Login y hacer el logout de la app. Esto lo veremos en detalle cuando estudiemos GAM.
- Ir a Home, que retorna al objeto inicial.
- ReturnTo que nos permite especificar un objeto distinto al anterior para hacer return.
- Save, Cancel, Delete: se usan en los objetos generados por el patrón WorkWith para encapsular las llamadas a los Business Components.
- TakeApplicationScreenShot para capturar una imagen de la pantalla del dispositivo.

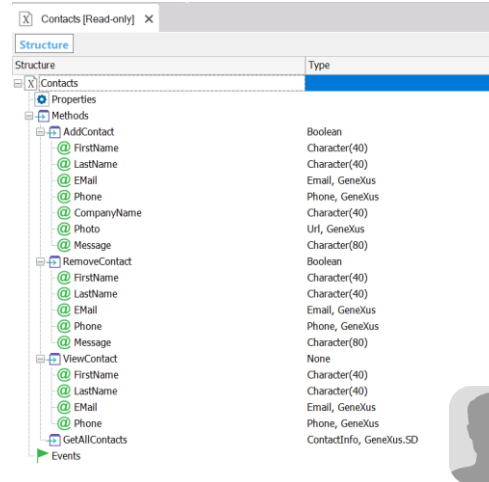
En este ejemplo vemos como se utiliza esta API cuando se realizan las operaciones de Save, Cancel desde el patrón WorkWith.

Calendar API



Smart
Devices APIs

Contacts API



La API Calendar nos permite generar un evento en la aplicación de calendario predeterminada del dispositivo.

La Api Contacts, nos permite interactuar con la aplicación de contactos del dispositivo, podemos agregar contactos, eliminarlos, ver la información de un contacto u obtener toda la lista.

Ambas están en el sub modulo SD.

Semantic Domains & Smart Devices APIs
GeneXus™

Network API

Network [Read-only]

Structure

- [-] Network
 - [-] Properties
 - [-] ApplicationServerURL
 - [-] Methods
 - [-] IsServerAvailable
 - [-] IsServerAvailable
 - [-] @ Url
 - [-] Type
 - [-] Type
 - [-] @ Url
 - [-] TrafficBasedCost
 - [-] TrafficBasedCost
 - [-] @ Url
 - [-] Events
 - [-] NetworkStatusChanged

ClientStorage API

ClientStorage [Read-only]

Structure

- [-] ClientStorage
 - [-] Properties
 - [-] Methods
 - [-] Set
 - [-] @ key
 - [-] @ value
 - [-] SecureSet
 - [-] @ key
 - [-] @ value
 - [-] Get
 - [-] @ key
 - [-] Remove
 - [-] @ key
 - [-] Clear
 - [-] Events
 - [-] Events

Smart
Devices APIs

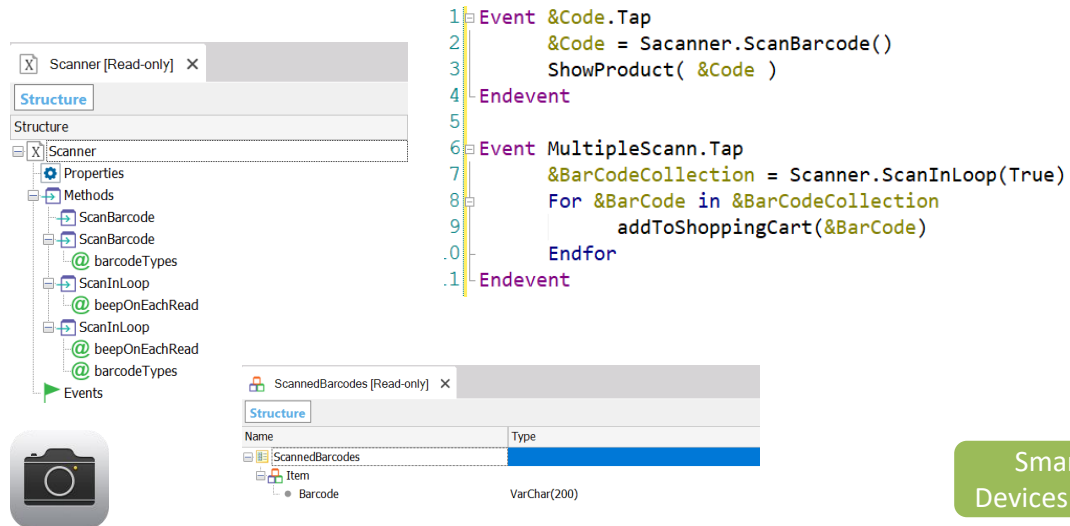


Tenemos una API Network, que nos permite saber si hay conexión con el servidor en un momento dado, y el tipo de conexión.

La API ClientStorage que nos permite guardar y recuperar valores, similares a las cookies, pero que se almacenan en el dispositivo.

Ambas son del sub modulo SD.

Scanner API



The screenshot displays the GeneXus IDE interface for the Scanner API. On the left, the 'Structure' pane shows the 'Scanner' object with properties, methods, and events. The 'Methods' section includes 'ScanBarcode', 'ScanBarcodeTypes', 'ScanInLoop', and 'ScanInLoopTypes'. The 'Events' section includes 'Tap', 'MultipleScann.Tap', and 'BeepOnEachRead'. A camera icon is visible below the structure pane. In the center, the code editor shows two event handlers: one for '&Code.Tap' that calls 'ScanBarcode()' and 'ShowProduct()', and another for 'MultipleScann.Tap' that calls 'ScanInLoop(True)' and iterates through '&BarcodeCollection' to call 'addToShoppingCart()'. On the right, the 'ScannedBarcodes' structure is shown, containing a 'ScannedBarcodes' collection of 'Item' objects, each of type 'Barcode' (VarChar(200)).

```

1 Event &Code.Tap
2   &Code = Sacanner.ScanBarcode()
3   ShowProduct( &Code )
4 Endevent
5
6 Event MultipleScann.Tap
7   &BarcodeCollection = Scanner.ScanInLoop(True)
8   For &Barcode in &BarcodeCollection
9     addToShoppingCart(&Barcode)
10  Endfor
11 Endevent

```

Name	Type
ScannedBarcodes	Item
Barcode	VarChar(200)

Smart
Devices APIs

La API Scanner, también del sub modulo SD, nos permite utilizar la cámara del dispositivo para escanear Códigos de barra o QR, tiene métodos para escanear un solo código, en ese caso retorna el valor en una variable de tipo Char o en un bucle y al finalizar nos retorna una lista de todos los códigos escaneados en una colección basada en el SDT ScannedBarcodes.

En el próximo video continuaremos viendo el uso de otras APIs y haremos un ejemplo en nuestra aplicación.



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