

Behavior: Events Execution Order



25-Behavior.Events.execution.order_sp



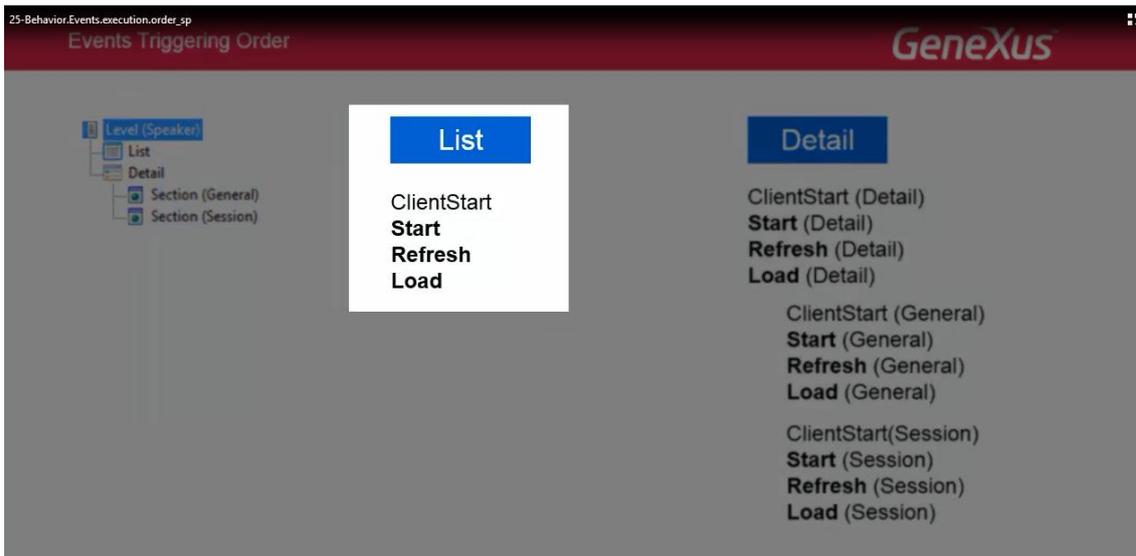
Developing the mobile application

Behavior: Events execution order

Cecilia Fernández | GeneXus Training

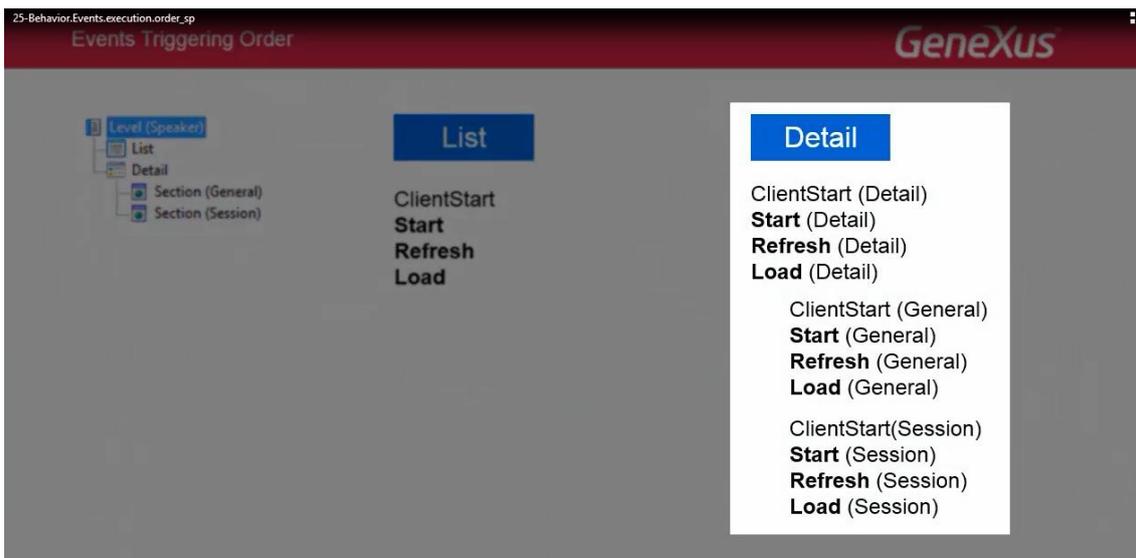
Veremos brevemente el orden de ejecución de los eventos del List y del Detail de un WorkWith, tanto cuando se abre el objeto por primera vez como cuando se dispara un evento del cliente.

Para los Panels for Smart Devices, valdrá lo mismo que para el List.



Si estamos abriendo un List, se van a ejecutar en este orden: el evento ClientStart en el cliente.. y luego los eventos Start, Refresh y Load, en el servidor si es que estamos dentro de una arquitectura online.

Si el objeto abierto es el Detail:



entonces se van a ejecutar en ese orden.

El ClientStart del Detail:



List

ClientStart
Start
Refresh
Load

Detail



ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

seguido del Start, Refresh y Load.. del propio Detail..

y luego por cada Section, empezando por la General



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

se van a ejecutar esos mismos eventos.



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

Luego de cargada la pantalla,



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

sea esta de List o Detail, el usuario está listo para interactuar.

¿Qué sucede cuando el usuario dispara una acción?



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

Client Events?

El código asociado es ejecutado del lado del cliente. Esto es, en el dispositivo.

Si se encuentra una llamada a un procedimiento Rest, Data Provider o Business Component, se realiza la llamada y se queda a la espera del resultado para seguir ejecutando el resto del código en el cliente.

Obsérvese que en cualquier caso, los eventos del sistema, Start, Refresh, Load.. no son ejecutados... a menos que se encuentre un comando Refresh dentro del propio código



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

Client Events?

Refresh Command?

¿Qué pasa en este caso?

Dependiendo de la configuración de caching, se decide si el dispositivo debe ir y obtener datos del server o no.

Luego de ir al server si no hay cambios en los datos, no se trae nada al cliente.

De lo contrario, los eventos Refresh y Load de todo grid no basado en un SDT, son ejecutados



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

Client Events?

Refresh Command?

Un caso particular es cuando el comando Refresh es ejecutado en un evento del cliente, de una sección del Detail



List

ClientStart
Start
Refresh
Load

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

Client Events?

Refresh Command?

En ese caso serán ejecutados los eventos



List

ClientStart
Start
 Refresh
 Load

Detail

ClientStart (Detail)
Start (Detail)
 Refresh (Detail)
 Load (Detail)



ClientStart (General)
Start (General)
 Refresh (General)
 Load (General)

ClientStart(Session)
Start (Session)
 Refresh (Session)
 Load (Session)

Client Events?

Refresh Command?

para el nodo Detail



List

ClientStart
Start
 Refresh
 Load

Detail

ClientStart (Detail)
Start (Detail)
 Refresh (Detail)
 Load (Detail)



ClientStart (General)
Start (General)
 Refresh (General)
 Load (General)

ClientStart(Session)
Start (Session)
 Refresh (Session)
 Load (Session)

Client Events?

Refresh Command?



List

ClientStart
Start
 Refresh
 Load

Detail

ClientStart (Detail)
Start (Detail)
 Refresh (Detail)
 Load (Detail)

ClientStart (General)
Start (General)
 Refresh (General)
 Load (General)

ClientStart(Session)
Start (Session)
 Refresh (Session)
 Load (Session)

Client Events?

Refresh Command?



y las secciones que hubieran sido previamente activadas

... excepto aquellas inline que serán activadas siempre.

Sobre todo esto, le sugerimos ver más en nuestro wiki:

Events Triggering Order GeneXus™



ClientStart
Start
Refresh
Load

Client Events?

Refresh Command?

List

Detail

ClientStart (Detail)
Start (Detail)
Refresh (Detail)
Load (Detail)

ClientStart (General)
Start (General)
Refresh (General)
Load (General)

ClientStart(Session)
Start (Session)
Refresh (Session)
Load (Session)

<http://wiki.genexus.com/commwiki/servlet/hwikibypageid?17614>

Retengamos todo lo visto para cuando estudiemos en instantes las aplicaciones offline.

Nos quedan por estudiar dentro de “comportamiento”, las invocaciones, esto es: las call options

25-Behavior.Events.execution.order_sp Behavior GeneXus™

Orders & Searches & Conditions

Events

Invocations

