

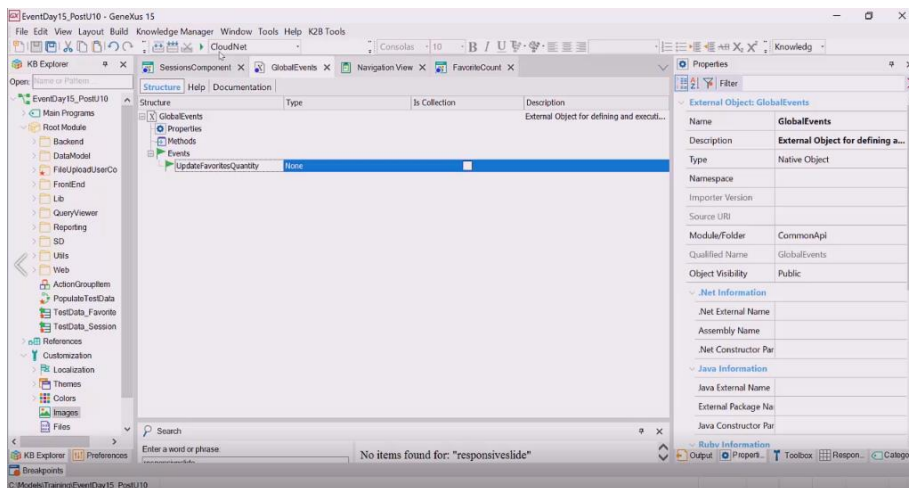
Una novedad muy importante para el desarrollo web en la versión GeneXus 15 son los Global Events.

Los Global Events básicamente nos sirven cuando queremos implementar una solución donde una acción en un componente en pantalla causa una reacción en otro componente.

Antes de la versión GeneXus 9.0 la comunicación entre web Component era súper limitada, la única comunicación posible era entre padres e hijos.

Pero los eventos globales nos permiten definir una comunicación entre componentes que no están relacionados directamente.

Por ejemplo, si vamos a la aplicación, este componente que despliega los eventos, al hacer clic en él se está refrescando un componente que se encuentra en el master page pero que no tiene una relación directa con el componente que lo está básicamente disparando.



Vamos a ver como logramos esto en GeneXus

Para ello tenemos un nuevo elemento, un nuevo objeto, GlobalEvents, podemos definir tantos

objetos Global Events como queramos, básicamente esto es un objeto externo, y podríamos definir GlobalEventsBackend, GlobalEventfrontend dependiendo cómo queramos agruparlos, es importante recordar que son globales y por lo tanto es importante utilizarlos con cierto criterio y ordenadamente para que no generen demasiado ruido.

Nosotros tenemos aquí nuestro GlobalEvents que básicamente lo que dice es que se actualice la cantidad de favoritos.

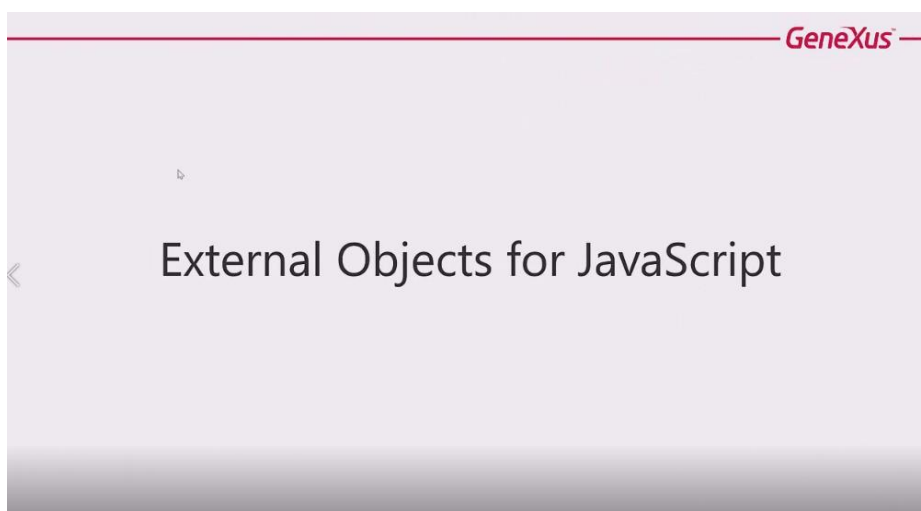
En nuestro caso, nuestro Evento no recibe parámetros, pero es posible implementar eventos que sí, si fuese necesario pasar parámetros seguirían definiendo los mismos, asociados al evento, con sus tipos de datos etc. y en el momento que se invoca es responsabilidad del objeto que dispara el evento definir cuáles son los parámetros y luego todos los componentes que implementan el evento o que escuchan por este evento van a recibir los parámetros con este formato.

¿Cómo disparamos este evento global? En nuestro caso lo estamos haciendo a partir del componente de sessions que básicamente cuando hace clic en la imagen de favoritos, se hace el toggle que muestra la imagen en gris o en amarillo y además se dispara el GlobalEvents.

¿Quién recibe/ejecuta este GlobalEvents? Eso es una decisión nuestra, y lo puede hacer cualquier componente que tenga sentido.

En nuestro caso, lo está haciendo el componente que muestra la cantidad de eventos favoritos, básicamente este componente implementa GlobalEvents.UpdateFavoritesQuantity y básicamente aquí va a hacer la llamada que necesite para actualizar la información, que podría ser un refresh etc.

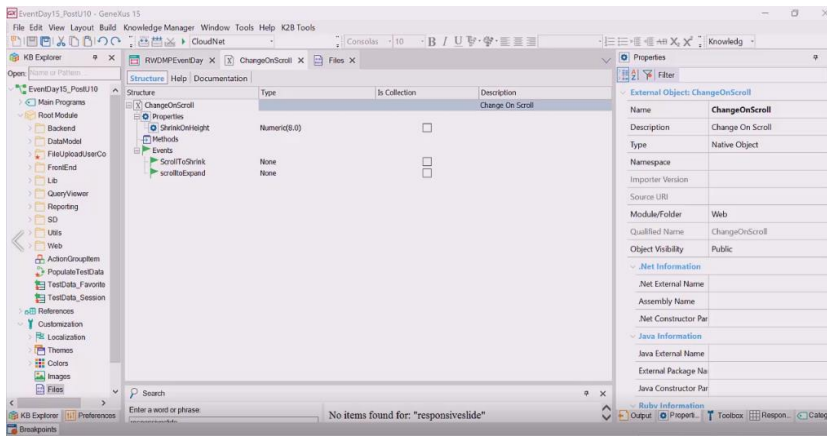
En definitiva si tuviéramos más de un componente que necesitar suscribirse a este evento, es posible, nosotros podríamos definir, o podríamos querer estar mostrando información de los favoritos en distintas partes de las pantallas y cuando alguien agrega un evento o una charla a sus favoritos, podríamos estar refrescando diferentes componentes cada uno de ellos implementando el Events.UpdateFavoritesQuantity.



Otra cosa que podemos hacer con los External Objects es interactuar con JavaScript, así que además del uso estándar de los external Objects y de los GlobalEvents también vamos a poder

utilizarlos para vincular eventos implementados en un JavaScript externo con evento GeneXus.

Vamos a verlo directamente en GeneXus...



En versiones anteriores, si queríamos realizar esto, podíamos hacerlo pero implementando un user control que es bastante más complejo que el objeto externo, esto nos permite acceder de forma más sencilla a funcionalidades implementadas a nivel de JavaScripts.

Vamos a ver un ejemplo en nuestra aplicación...

En este caso cuando hacemos scroll en la aplicación veíamos que en la barra superior se achicaba, esto se está implementando con un evento JavaScriptsChangeonscroll, vamos a ver como logramos vincular ese JavaScripts con GeneXus.

El JavaScripts que tenemos es muy sencillo, básicamente cuando define que se llegó a cierta altura de scroll dispara un evento Shrink y sino dispara un evento Expand.

¿Cómo vamos a hacer para interactuar o para vincular este JavaScripts con GeneXus? Lo hicimos a través de nuestro objeto externo, Changeonscroll básicamente está asociado con un JavaScripts externo que se llama Changeonscroll y aquí están los eventos que este JavaScripts está disparando, en este caso tienen el mismo nombre que lo que tiene el JavaScripts pero podríamos cambiarlo y mapeando el nombre externo a un nombre que de repente en GeneXus pueda tener más sentido.

¿Dónde implementamos estos eventos en nuestra aplicación? Bueno, lo estamos haciendo en la master page en donde tenemos el objeto externo Changeonscroll con el evento ScrolltoExpand y el evento ScrolltoShrink y en un caso estamos ocultando y en el otro caso estamos mostrando el componente que básicamente queremos que se vea o se oculte según qué tan abajo estamos en la barra de scroll.

Para incluir el JavaScripts en la aplicación lo estamos haciendo de la misma forma que lo hicimos siempre que es agregar el scripts en el rodo HTML.

Accessibility

Por último, vamos a ver que tenemos en la versión GeneXus 15 con respecto a las facilidades de crear aplicaciones accesibles.

Accesibilidad web, refiere a la práctica de diseñar y de desarrollar una aplicación de forma que todos los usuarios tengan acceso a su contenido y funcionalidades. Para cumplir con esta práctica inclusiva las aplicaciones web deben ser accesibles a todos sin importar el hardware o software desde el cual se accede, la ubicación en la que se encuentra, es decir su lenguaje o cultura y también sus habilidades. Por tanto es importante tener en cuenta a la hora de desarrollar que sea compatible con tecnología de apoyo como pueden ser lectores de pantalla, o magnificadores de pantalla y también que cumpla con especificaciones y recomendaciones de accesibilidad.

GeneXus 15 permite a los desarrolladores realizar aplicaciones con estas características y adicionalmente el código generado cumple con los estándares de HTML5 CSS3 que son la base para aplicaciones accesibles.

Pero una parte significativa del desarrollo de aplicaciones accesibles depende del desarrollador, es el desarrollador el que debe programar la aplicación siguiendo las guías de accesibilidad.

Por ejemplo, las imágenes siempre tienen que tener un valor en la propiedad Texto alternativo porque esto es lo que usan los lectores de pantalla que permiten acceso a usuarios ciegos. También permite el acceso a usuarios que tengan poco ancho de banda y que puedan apagar por ejemplo la visualización de imagen.

Como es nuestra responsabilidad desarrollar aplicaciones que cumplan con estos requisitos de accesibilidad es importante revisar y seguir los Guidelines publicados al respecto.

Como decíamos, en GeneXus lo que tenemos es acceso a propiedades que nos van a permitir definir o cumplir con todos los requisitos de accesibilidad.

En este caso lo que no tenemos es un botón derecho, propiedades generar aplicación como accesible, sino que somos nosotros los que tenemos la responsabilidad de definir a nivel de cada uno de los elementos de la aplicación y cada uno de los controles, las propiedades necesarias para que cumplan con estos requisitos de accesibilidad.

GeneXus lo que si hace es ser compatible con todos los estándares necesarios (HTML5 CSS3) para

generar aplicaciones accesibles, pero parte de la responsabilidad cae sobre nosotros.

