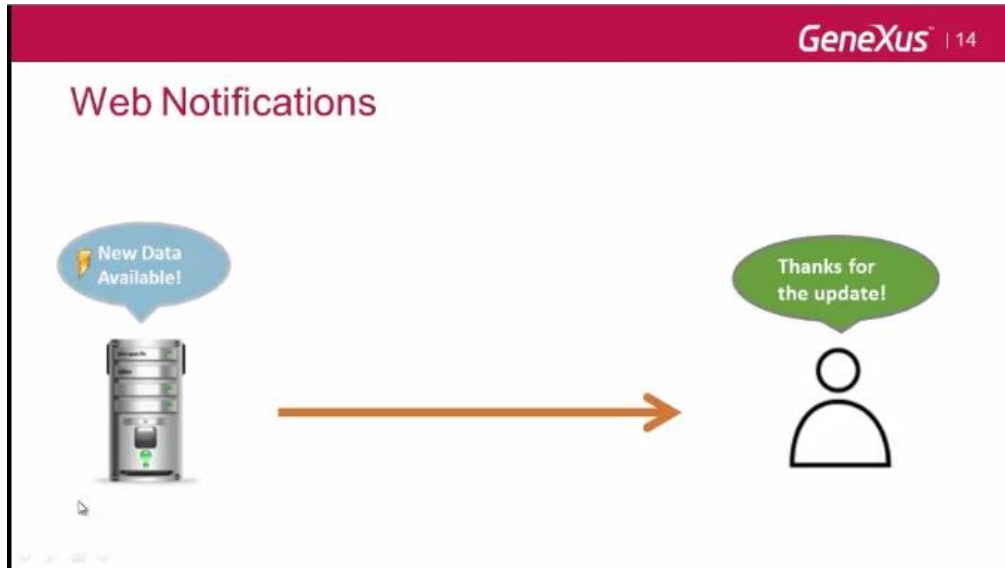


Notificaciones Web

Logramos aplicaciones que ayuden a los usuarios recibir feedback inmediato en tiempo real.



Consiste en que el servidor va a poder mandar información al cliente sin que éste lo solicite. Son notificaciones silenciosas que se reciben del lado del cliente. Estas notificaciones utilizan la tecnología web socket, que son parte del estándar HTML5.

Aquí tenemos la nueva API que es parte del envío de la notificación y es la que nos permite enviar una notificación a un cliente o a todos.

The screenshot shows the GeneXus IDE interface with the 'WebNotification' structure. The structure is organized into Properties and Methods. The 'Notify' method is highlighted in blue. The 'NotifyClient' method is also highlighted in blue. The 'Broadcast' method is highlighted in blue. The 'Notify' method has a parameter 'sdNotification' of type 'NotificationInfo'. The 'NotifyClient' method has parameters 'sdNotification' and 'clientId' of type 'NotificationInfo'. The 'Broadcast' method has a parameter 'sdNotification' of type 'NotificationInfo'.

Structure	Type	Is Collection	Description
WebNotification			Web Notification
ClientId	Character(100)	<input type="checkbox"/>	
ErrCode	Numeric(4,0)	<input type="checkbox"/>	
ErrDescription	Character(100)	<input type="checkbox"/>	
Notify	Numeric(4,0)	<input type="checkbox"/>	
NotifyClient	NotificationInfo	<input type="checkbox"/>	
sdNotification	Numeric(4,0)	<input type="checkbox"/>	
clientId	Character(100)	<input type="checkbox"/>	
sdNotification	NotificationInfo	<input type="checkbox"/>	
Broadcast	None	<input type="checkbox"/>	
sdNotification	NotificationInfo	<input type="checkbox"/>	

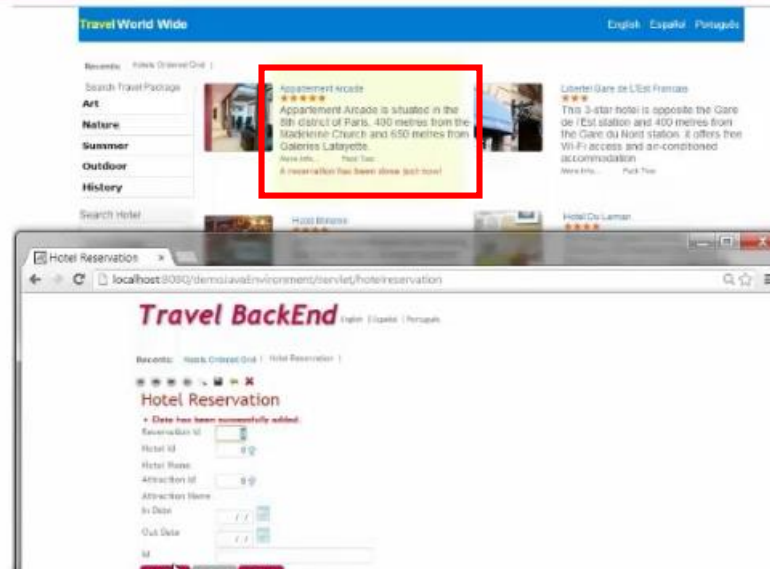
Del lado de la recepción de la notificación tenemos un nuevo evento: El evento OnMessage; que nos permite programar una acción al momento de la llegada de la notificación. Este evento recibe por parámetro el SDT que vemos en pantalla.

OnMessage Event

NotificationInfo		
Name	Type	Description
NotificationInfo		Notification Info
▪ Id	Character(100)	Id
▪ Object	Character(100)	Object
▪ Message	LongVarChar(2M)	Message

Pero veamos esto en un ejemplo muy sencillo. Lo que simulamos es que al momento de la agencia de viaje realiza una reserva todos los usuarios que en ese momento estén viendo el hotel, recibirán una notificación, la cual les avisa que en ese momento alguien realizó una reserva. En este ejemplo, la reserva se va a realizar sobre los hoteles que uds tienen arriba, a su izquierda; entonces podrán observar que al momento de realizar la reserva dicha notificación llega a todos los usuarios.

Pudieron observar como luego el mensaje a todos los usuarios sin la necesidad de refrescar la pantalla.



Aquí podrán observar cómo se arma dicha notificación y luego como utilizamos el método broadcast para enviar la notificación a todos los usuarios.

```
NewHotelReservation x
1 &HotelReservationSDT.AttractionId = &AttractionId
2 &HotelReservationSDT.HotelId = &HotelId
3 &HotelReservationSDT.RoomsAvailable = &HotelRoomsAvailable
4 &HotelReservationSDT.ReservationHotelId = &ReservationHotelId
5 &NotificationInfo.Id=&AttractionId.ToString()
6 &NotificationInfo.Message=&HotelReservationSDT.ToJson()
7 &webnotification.Broadcast(&NotificationInfo)
```

Desde el lado de la recepción de la notificación, como dijimos anteriormente, utilizamos el evento OnMessage gracias al for each line in grid1 podemos refrescar solamente la línea correspondiente a la notificación sin la necesidad de refrescar toda la pantalla.

```
event onmessage(&NotificationInfo)
  if &NotificationInfo.Id.Trim() = &AttractionId.ToString().Trim()
    &HotelReservationSDT.FromJson(&NotificationInfo.Message)
    for each line in grid1
      if &HotelReservationSDT.HotelId = &HotelId
        notification.Caption = "A reservation has been done just now!"
      endif
    endfor
  endif
endevent
```