GeneXus by Globant

Invocations Between Mobile Objects

Diego Marranghello

Veremos neste vídeo como invocar diferentes objetos e quais opções de invocação temos. Para este último, usaremos as CallOptions, que nos permitirão modificar em runtime certas propriedades que têm a ver com a experiência do usuário.

Mas antes, vamos revisar a sintaxe das invocações.

Por exemplo, o caso das invocações aos objetos WorkWith:

Para invocar o List de um Work With, é feito indicando o nome do Work With, ponto, o nome do nível e a seguir o método List sem parâmetros. E dessa maneira acessaremos a lista de registros.

Se quisermos acessar o detalhe de um determinado registro, em modo de visualização, é utilizada a sintaxe que vemos em tela, onde precisamos passar a primary key para identificar de quem queremos mostrar aquele detalhe.

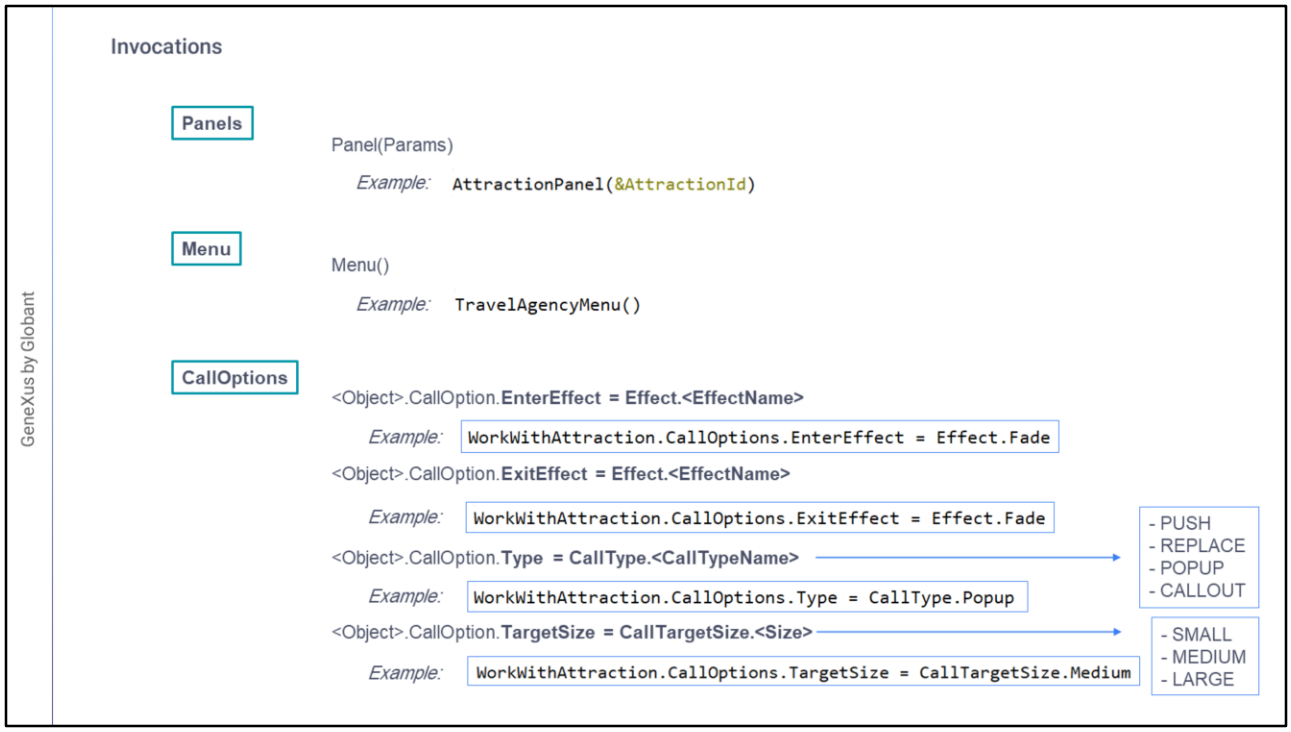
Para o caso do Detail em modo Edit, devemos indicar o modo:

Se será Insert, Update ou Delete. Ou seja, se será um novo registro, uma atualização ou uma exclusão de um já existente.

No caso do Insert temos 2 opções, não passar nenhum parâmetro, para que o usuário possa inserir todos os dados do zero. Ou se quisermos inicializar alguns valores nesse Insert, devemos passá-los em um Business Component. Neste caso, devemos

previamente inicializar no Business Component os valores que temos interesse em passar e então esses valores aparecerão inicializados na tela.

E para os casos de update e delete, devemos passar por parâmetro a chave primária do registro que queremos excluir ou atualizar.



Por outro lado, temos a invocación dos Panels, que é exatamente igual à invocación de qualquer outro objeto GeneXus, indicando o nome do panel, e passando os parâmetros que forem necessários de acordo com os parâmetros de entrada que tenha esse objeto Panel.

Também podemos chamar um objeto Menu.

Depois temos as CallOptions que, como dissemos, nos permitirão modificar em runtime certas propriedades que têm a ver com a experiência do usuário, estas configurações deverão ser realizadas um instante antes de invocar o objeto.

As propriedades que podemos configurar são:

• Os efeitos de transição, que podem ser o de Entrada ou o de Saída. Os valores possíveis são os do domínio Effect.

Este é um domínio enumerado, que oferece todas estas opções.

• O tipo de chamada, utilizando o domínio enumerado CallType, com os tipos Push, Replace, Popup e Callout (este último disponível apenas no iOS)

Esta configuração permite-nos modificar o comportamento da chamada, no que diz respeito ao tipo de call, que terá a ver com a stack de invocações e com o funcionamento do objeto chamado.

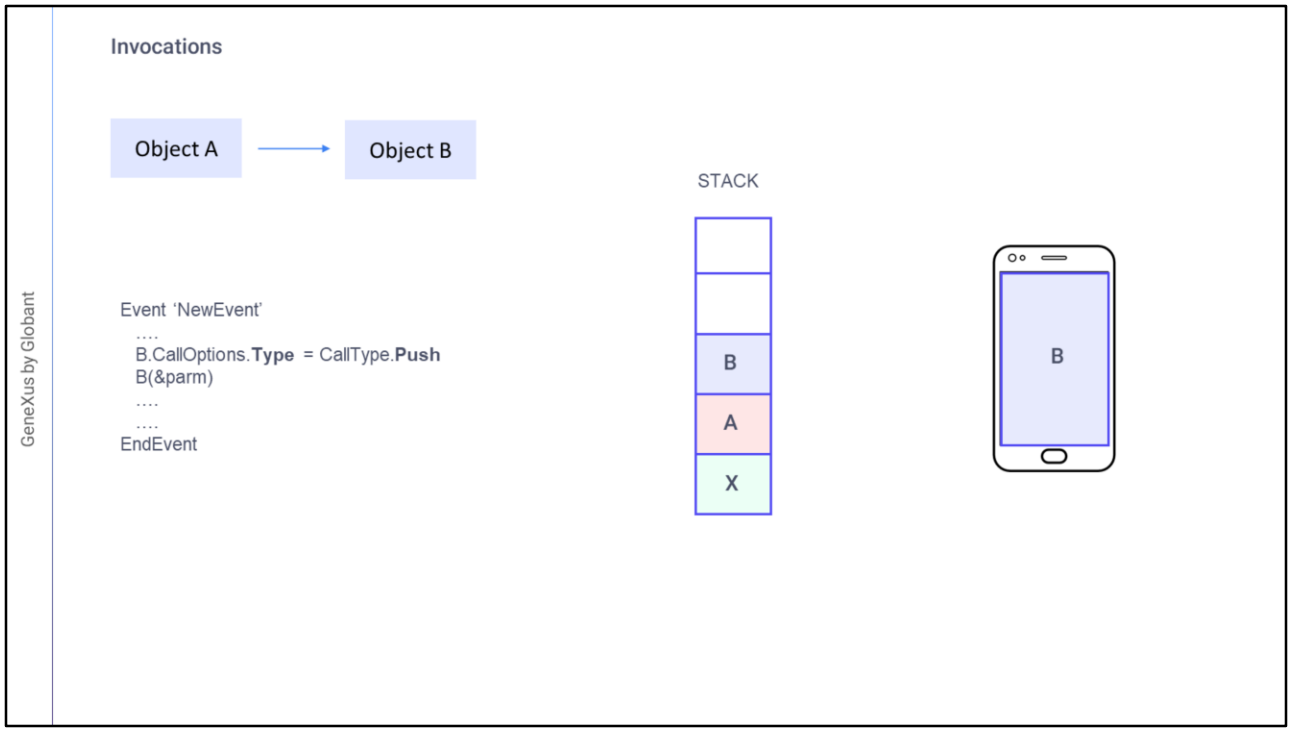
Os tipos Push e Replace definem o que acontecerá com a stack de invocações quando for feita a chamada, o que terá a ver com, para qual objeto será retornado ao finalizar a execução da chamada ou ao clicar em back.

Os tipos Popup e Callout serão para que a tela a ser invocada seja do estilo popup ou callout.

Para Popup, a tela poderá ser modal ou não, dependendo se há parâmetros retornados ou não. As janelas modais bloqueiam todas as funções e concentram o foco em uma ação específica. O usuário só pode fazer essa ação ou fechar a janela, caso não seja modal, tocando fora da área retornará para o chamador.

Para o caso de Callout, não será modal.

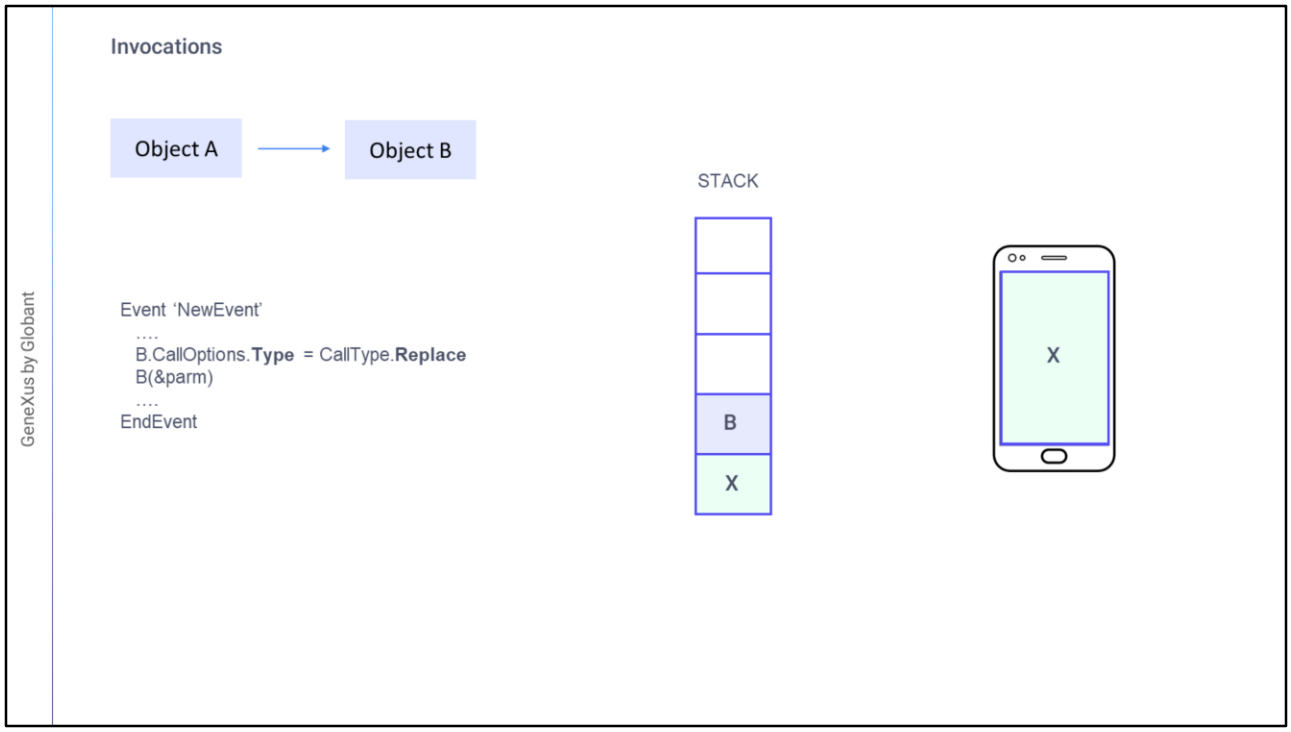
Quando escolhemos uma dessas duas opções, aparece a outra CallOption: CallTargetSize, para indicar o tamanho da tela Popup ou Callout, com as opções Small, Medium e Large



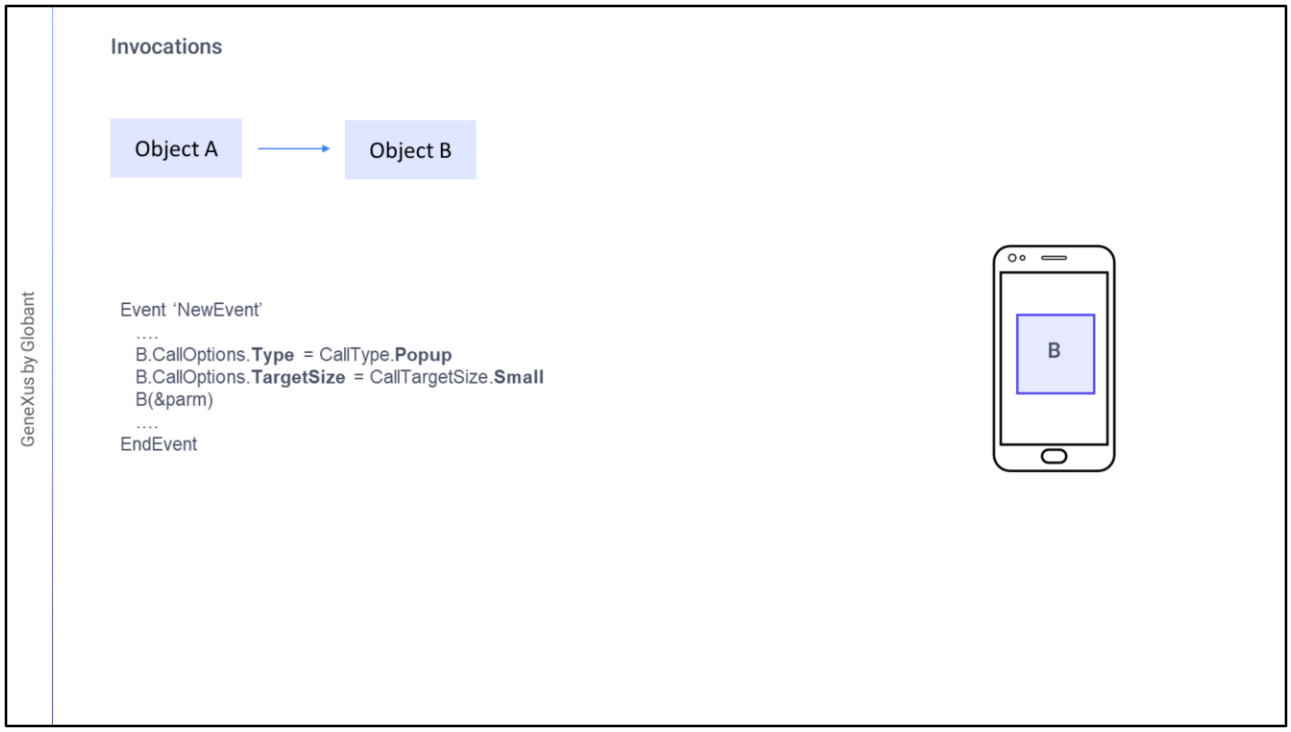
Em relação aos tipos Push e Replace, para entender o funcionamento de ambos, vejamos um pequeno exemplo:

Suponhamos que um objeto X, chamou um objeto A.
Se agora a partir de A em um evento chamamos um objeto B, com o tipo de call Push, o objeto chamado é colocado acima na stack.

Sua tela é aberta sobre a tela do chamador, ocupando exatamente o mesmo lugar. E o chamador espera para continuar sua execução, para que termine a execução do objeto chamado: B, que é assim retirado da stack.



Se, em vez disso, a partir do objeto A chamarmos B com o tipo de call Replace, o objeto chamado também será aberto ocupando exatamente a mesma área de tela que o chamador, mas substituirá na stack o objeto chamador. Portanto, quando terminar sua execução, não continuará a execução do evento de A, mas retornará ao objeto que estava antes na stack; neste caso, o objeto: X

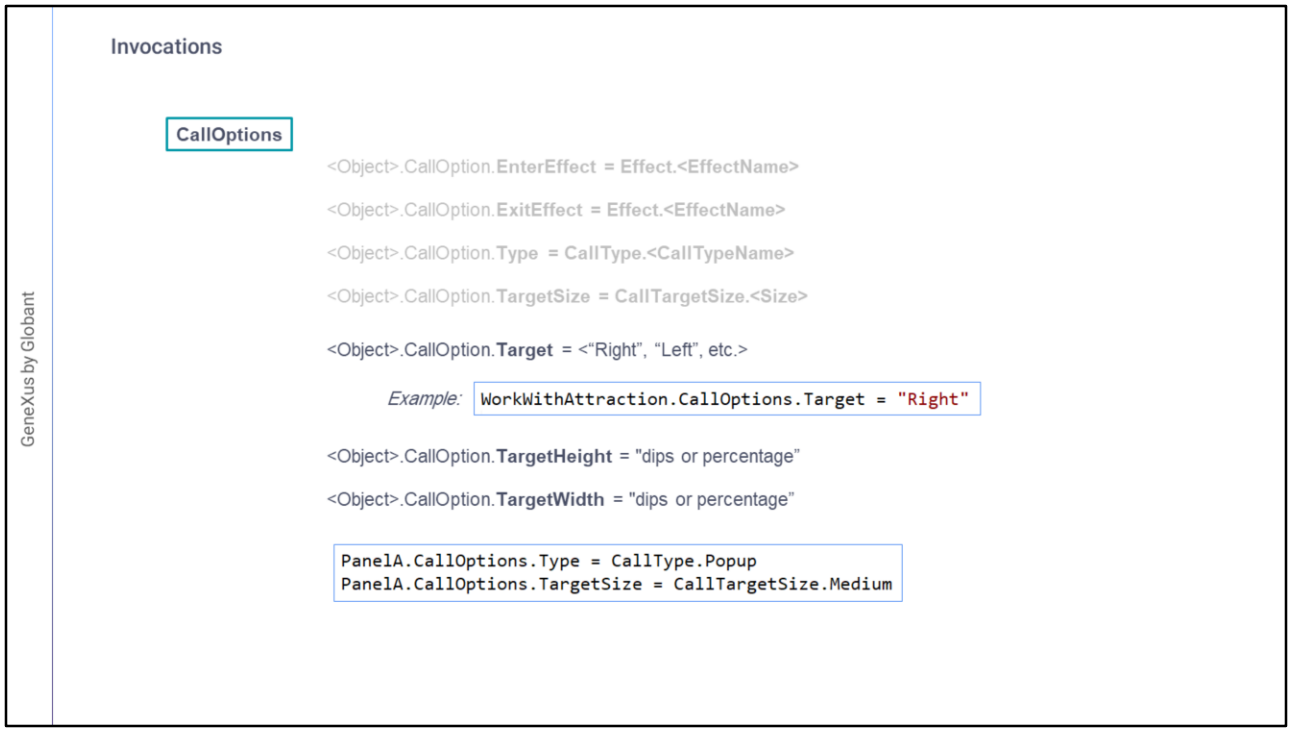


Agora, com relação ao Popup e Callout, vejamos o tipo Popup.

Se não for modificado o TargetSize, ocupará a mesma área de tela que o chamador. Caso contrário, ocupará a área que tenhamos especificado.

Se entre os parâmetros de invocação algum for output, então a caixa de diálogo será modal, ou seja, o chamador aguardará o retorno da execução de B para continuar

Se nenhum dos parâmetros for de output, então a caixa de diálogo não será modal.



Voltando às calloptions, também temos:

- O target, onde será exibido o objeto chamado, para indicar em qual dos targets possíveis você deseja carregar a tela chamada

Isso faz sentido quando se trata de estilos de navegação onde existem múltiplos targets para uma mesma tela, ou seja, tela dividida, mais utilizado em tablets do que em telefones. Não pode ser utilizado com Callout ou Popup.

- E de forma personalizada podemos definir o tamanho da janela chamada, especificando a Largura e Altura, em dips ou porcentagens, usando as propriedades TargetWidth e TargetHeight

Por exemplo, se quiséssemos chamar um panel como Popup com um tamanho Medium, poderíamos configurar as propriedades Type e TargetSize como vemos em tela e então fazer a chamada ao objeto normalmente.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com