Módulos e Objetos Externos

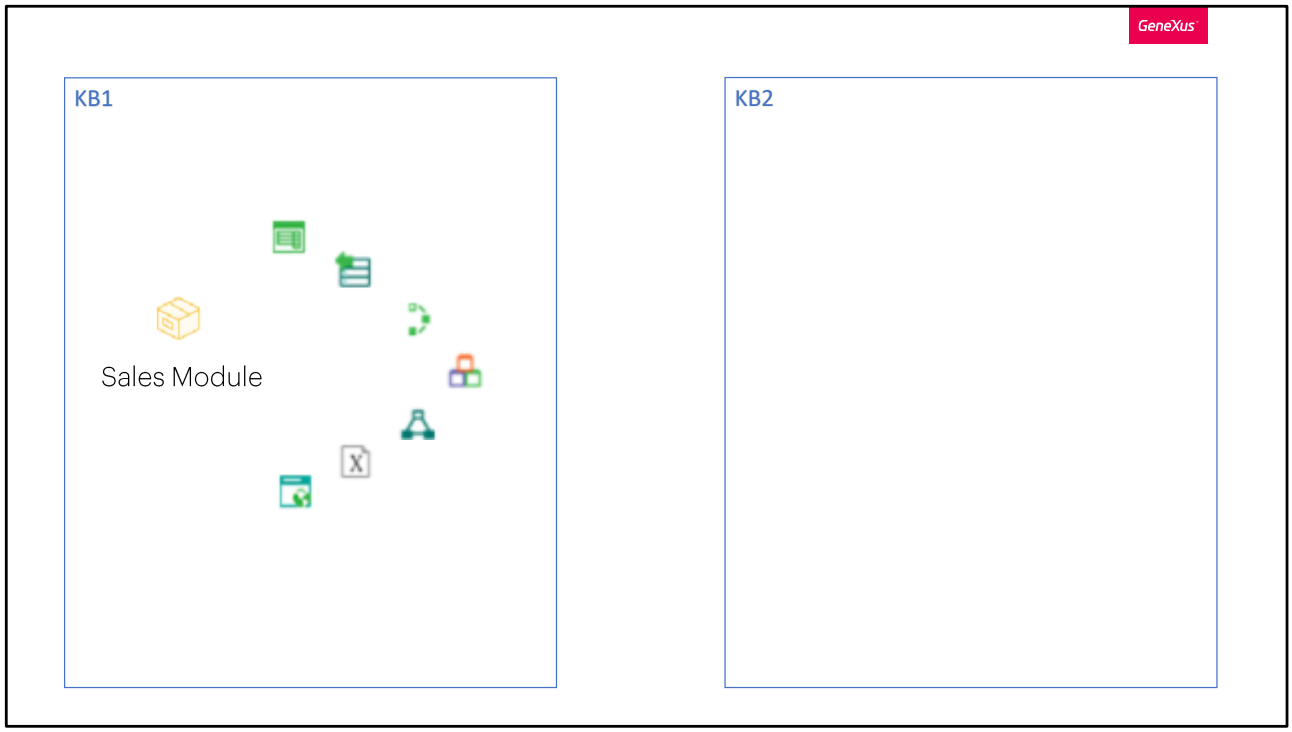
GeneXus™

A seguir veremos alguns objetos GeneXus que são úteis na hora de encapsular funcionalidades e organizar os objetos de nossa base de conhecimento.

GeneXus

Módulos

Vejamos em primeiro lugar os módulos.

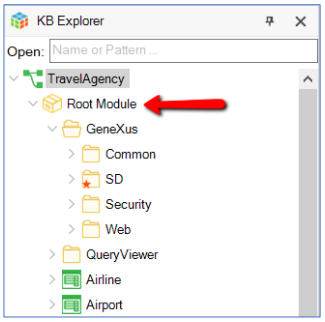


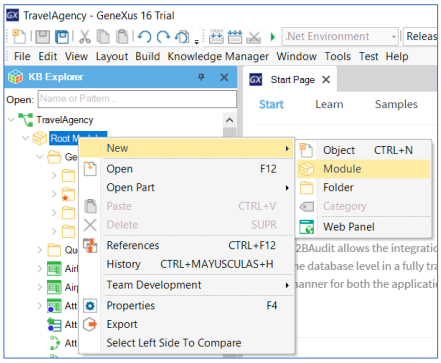
Os módulos são objetos containers GeneXus, que nos permitem agrupar objetos de nossa KB facilitando a compreensão da mesma, sua manutenção e a integração de objetos com outras KBs.

GeneXus

O que é um módulo?

- Objeto GeneXus que nos permite agrupar objetos da KB e encapsular suas funcionalidades
- Projetado para facilitar o entendimento, manutenção e integração de objetos entre KBs
- Módulos e pastas nos permitem criar hierarquias





Quando criamos uma base de conhecimento, é criado o módulo Root Module e, por padrão, todos os objetos que criamos ficam nesse módulo.

Tanto os módulos como as pastas nos ajudam a organizar objetos, no entanto, existem diferenças conceituais entre um módulo e uma pasta. Os módulos nos ajudam a encapsular e modularizar partes da KB, sendo capazes de estabelecer quais objetos são visíveis a partir de outros objetos e quais não são, como veremos mais adiante.

As pastas, por outro lado, são containers que apenas nos ajudam a organizar objetos, separando-os de acordo com algum critério.
Juntamente com os módulos, criam uma árvore hierárquica onde a raiz é sempre um módulo, como no caso do Root Module. Podemos ver isso no KB Explorer.

Os módulos podem ter pasta como filhos, mas as pastas não podem ter módulos como filhos.
Como regra geral, podemos dizer que se podem usar os módulos para encapsular e as pastas para organizar os objetos dentro do módulo.

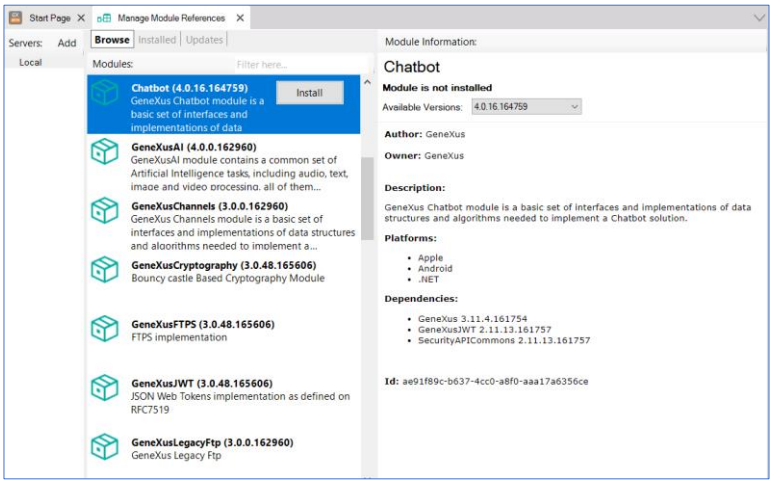
Para adicionar um objeto a um módulo, podemos arrastá-lo dentro do KB Explorer até o módulo, ou clicar com o botão direito do mouse sobre o

módulo e, em seguida New Object, ou alterar o valor da propriedade Module/Folder do objeto.

GeneXus

Adicionando módulos já criados para a KB

- Knowledge Manager / Manage Module References



Os módulos que são empacotados e que foram compartilhados conosco, os vemos através do menu Knowledge Manager / Manage Module References.

Para cada módulo disponível, podemos ver suas informações e decidir se o instalamos ou não em nossa KB. Se o instalamos, será salvo no nó References do KB Explorer, ao contrário dos objetos criados por nós, que por padrão são armazenados no Root Module.

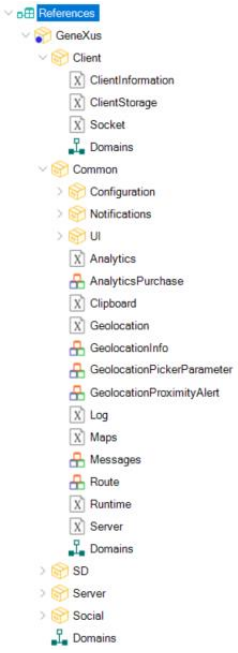
Os objetos destes módulos não poderemos modificar (os veremos como Read-Only) e já estão compilados, por isso quando damos F5 não será necessário especificá-los, gerá-los, etc.

No entanto, poderemos utilizá-los livremente a partir dos objetos da nossa aplicação, utilizando as funcionalidades que oferecem.

Qualquer membro da comunidade pode criar, compartilhar ou até mesmo vender seus módulos através do Marketplace.

Um destes módulos é o módulo GeneXus, também conhecido como GeneXus Core.

GeneXus Module



O módulo GeneXus é distribuído automaticamente e instalado em todas as KBs e, como qualquer módulo externo à nossa aplicação, o encontramos no nó References.

É composto por uma série de sub-módulos que contêm um conjunto de APIs com seus respectivos domínios e SDTs, que nos permitem interagir com diferentes tecnologias, dispositivos, sensores, aplicações, etc.

Estas API's são implementadas como Objetos Externos, que veremos a seguir.

GeneXus

Más información sobre módulos

https://wiki.genexus.com/commwiki/servlet/wiki?22411

Para mais informações sobre o objeto módulo, você pode seguir o seguinte link do Wiki:
https://wiki.genexus.com/commwiki/servlet/wiki?22411

GeneXus

Objetos Externos

Vejamos agora o que são os Objetos Externos.

Os objetos externos são objetos GeneXus que nos permitem acessar recursos externos à nossa KB, como se fossem mais um objeto da KB.

Por esse motivo, são cada vez mais frequentes e importantes em nossas aplicações, tanto web quanto para dispositivos móveis. Vamos ver como usá-los.

GeneXus

Quais recursos podemos utilizar como objetos externos em nossa KB?

- Native objects from programming languages:
  - .NET Assemblies (.dll)
  - Java Classes (.class)
- Resources from several external sources:
  - Enterprise Java Beans (EJB)
  - Stored Procedures in a DBMS
  - Web Services (WSDL=SOAP, OpenApi=REST) published in a Web Server
  - SAP BAPI modules
  - JSON files
  - XML schemas
- External Objects available in GeneXus
  - APIs for Smart Devices: Access to HW (camera, GPS, microphone, etc.) y SW (calendar, contacts, notifications, etc.)
  - APIs to Web: Access to events, communication with server, clipboard, maps, social networks, etc.
- External Objetos published in GeneXus Marketplace

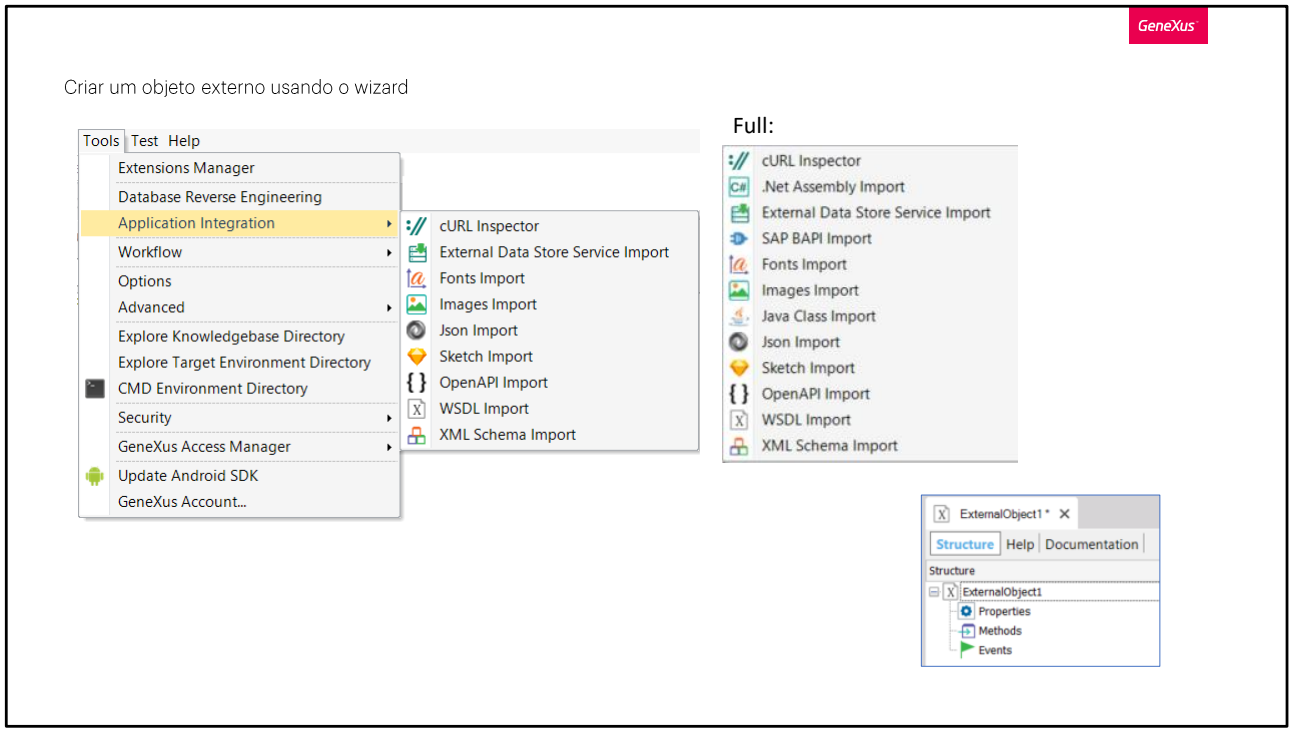
É possível importar diferentes tipos de recursos para nossa KB. Por exemplo, se temos algo programado em .NET, podemos gerar uma DLL e importá-la para a KB como um objeto externo. Então poderemos invocar a partir da nossa aplicação as funções incluídas na DLL, como se fossem procedimentos programados no GeneXus.
O mesmo acontece com as classes criadas em Java.

Também podemos importar recursos armazenados em outras fontes externas, como programas Java Beans, ou procedimentos armazenados em uma base de dados, webservices (tanto SOAP como REST), módulos SAP, arquivos JSON gerados por qualquer aplicação, ou esquemas XML.

GeneXus também fornece um conjunto de Objetos Externos que estão no RootModule ou no módulo GeneXus, que nos permitem o acesso a uma variedade de recursos, como por exemplo APIs para poder interagir com o hardware ou aplicações nativas de dispositivos móveis, ou APIs para acessar o servidor, eventos ou aplicações do Windows, ou sites externos, como o uso de mapas, redes sociais, etc.

Também dispomos de objetos externos publicados no Marketplace do GeneXus, que podemos incorporar em nossa aplicação.

También disponemos de objetos externos publicados en el Marketplace de GeneXus, que podemos incorporar a nuestra aplicación.

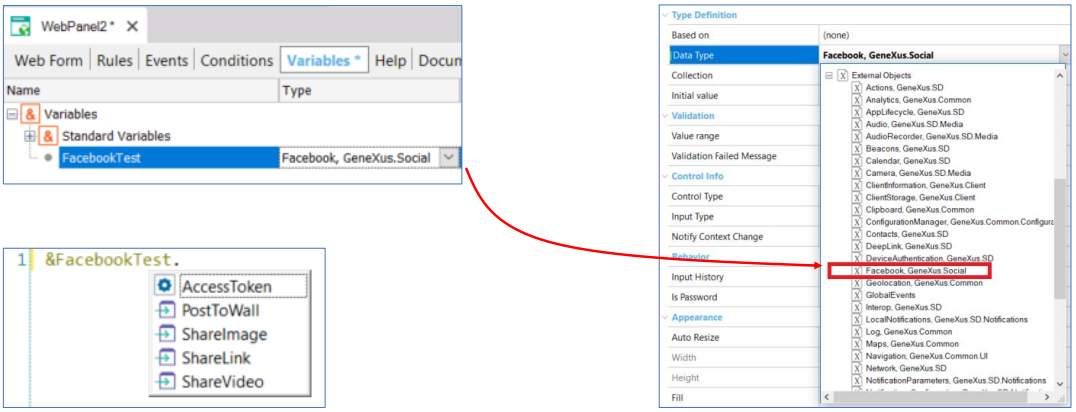


Para criar um objeto externo, a melhor maneira é fazê-lo através de um wizard. Se vamos ao menu Tools e escolhemos Application Integration, vemos os diferentes recursos a importar e será executado um wizard específico para esse recurso. Ao finalizar o wizard, será criado um objeto externo que ficará automaticamente associado ao recurso, todas as propriedades do objeto externo ficarão ajustadas conforme o tipo de recurso importado.

Podemos também criar um objeto externo com New Object como qualquer outro objeto GeneXus, mas nesse caso teremos que configurar suas propriedades, métodos e eventos manualmente.

Como usamos um objeto externo?

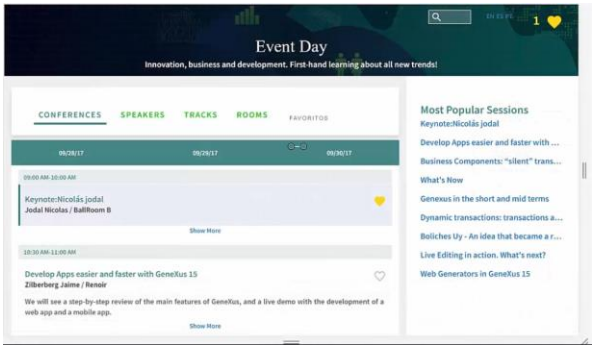
1) Criamos uma variável cujo tipo seja o Objeto Externo
2) Invocamos os métodos ou atribuímos as propriedades disponíveis

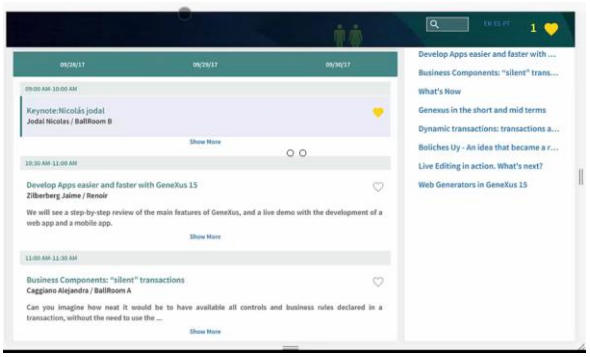


Uma vez criado o objeto externo com base nas propriedades correspondentes ao recurso externo que desejamos utilizar, o mesmo ficará disponível como qualquer outro tipo de dados na Base de conhecimento.

A maneira de usá-lo é a mesma que com qualquer tipo de dado estendido: definindo uma variável desse tipo e, em seguida, chamando os métodos e/ou configurando as propriedades que precisamos.

Exemplo de uso de um objeto externo com JavaScript



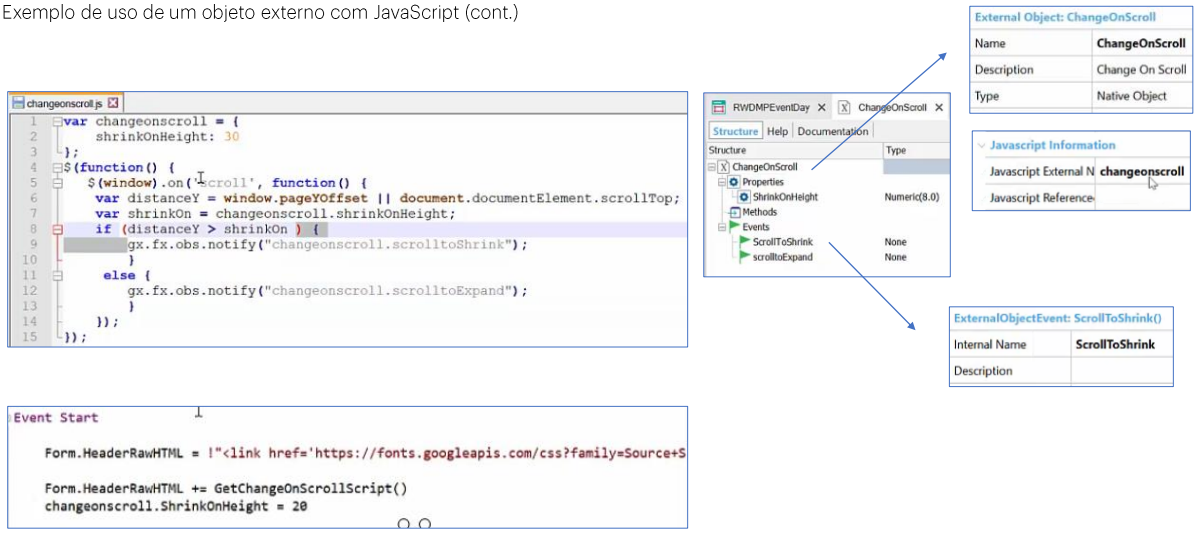


Outra coisa que podemos fazer com os External Objects é interagir com JavaScript, por exemplo para vincular eventos implementados em um JavaScript externo com um evento GeneXus.

Vejamos isto com uma aplicação de exemplo.

Vemos que quando fazemos scroll na aplicação, a barra superior fica menor.
Isto está sendo implementado com um evento JavaScriptChangeOnScroll programado de forma externa, vamos ver como podemos vincular esse JavaScript com GeneXus.

Exemplo de uso de um objeto externo com JavaScript (cont.)



O JavaScript que temos é muito simples, basicamente quando define que se chegou a uma certa altura de scroll dispara um evento Shrink e, caso contrário, dispara um evento Expand.

Como vamos fazer para vincular este JavaScript com GeneXus?

Fazemos isto com um objeto externo Changeonscroll, que basicamente está associado com um JavaScript
externo que se chama Changeonscroll e aqui estão os eventos que este JavaScript está disparando.
Neste caso, têm o mesmo nome que no JavaScript, mas poderíamos alterá-lo.

Estes eventos foram implementados no Web Master Panel (ou master page) onde está incluído o objeto externo Changeonscroll com o evento ScrolltoExpand e o evento ScrolltoShrink.
Em um caso, estamos ocultando o componente e, no outro caso, o estamos mostrando, pois basicamente queremos que ele seja visto ou se oculte dependendo de quão abaixo estivermos na barra de scroll.

Para incluir o JavaScript na aplicação, estamos fazendo da mesma forma que sempre fizemos,

que é adicionar o script no código HTML.

Para mais informações sobre os objetos externos, você pode seguir o seguinte link na tela.

GeneXus™

training.genexus.com
wiki.genexus.com