GeneXus by Globant

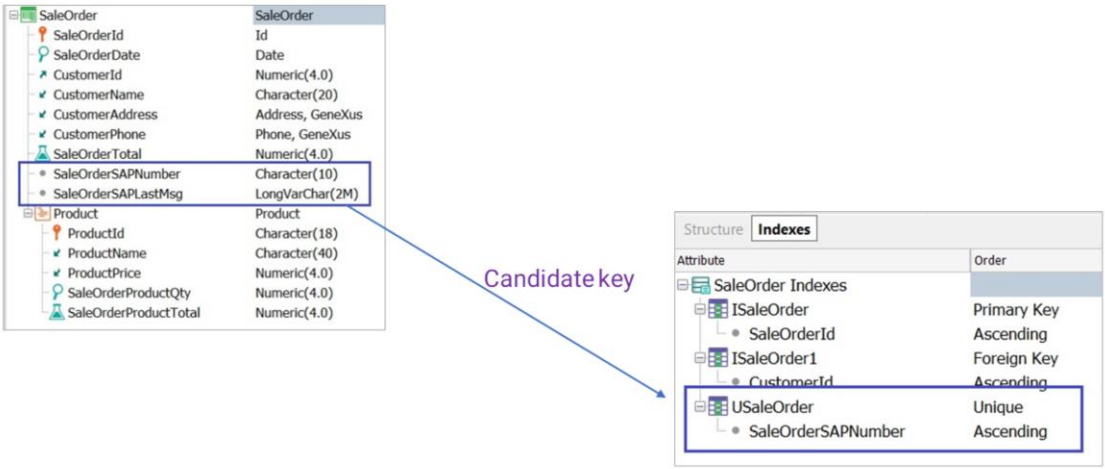
Integrando a KB ao SAP ERP

Trabalhando com Ordens de Venda

Avançando em nosso desenvolvimento, vamos agora trabalhar com as ordens de venda. Nosso objetivo é poder gerar um pedido em nossa aplicação e criá-lo no ERP de SAP.

Se a operação for bem-sucedida, será devolvido para nós um número de ordem de venda criado por SAP. E caso haja algum problema com a conexão ou criação do pedido, devolverá o erro gerado.

Trabalhando com Ordens de Venda

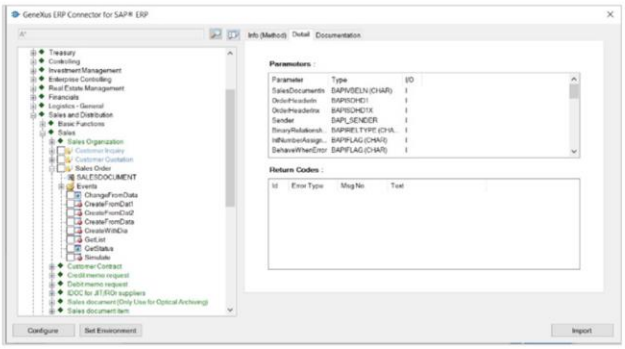


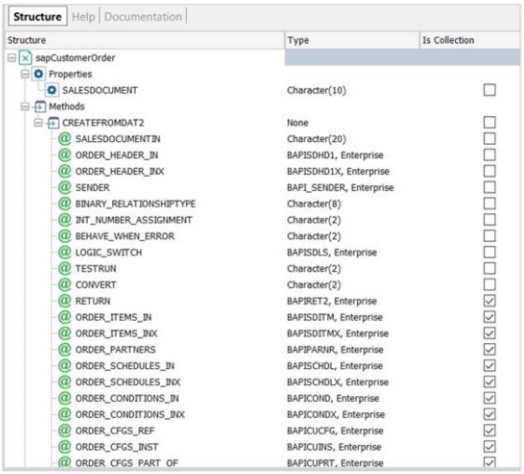
Portanto, precisamos adicionar um novo atributo à nossa transação SaleOrder para armazenar o número retornado pelo SAP. Nós o chamamos de SaleOrderSAPNumber. E definimos também outro atributo que armazenará a mensagem de erro em caso de ocorrer alguma falha. Nós o chamamos de SaleOrderSAPLastMsg, do tipo LongVarChar.

Este atributo será uma chave candidata nesta transação. Isto significa que será um valor único que identificará cada ordem de venda, embora não seja a chave primária definida na transação

Portanto, criamos o correspondente índice unique na tabela SALEORDER, sobre o novo atributo SaleOrderSAPNumber.

Trabalhando com Ordens de Venda





Bom. Agora precisamos importar a BAPI correspondente para poder criar ordens de venda no ERP.

Então, vamos para Tools / Application integration / SAP BAPI Import e pressionamos Ok.

Vamos para Sales and Distribution, Sales, Sales Orders e selecionamos este Create from

Vemos sua informação e a lista de parâmetros na aba Detail:

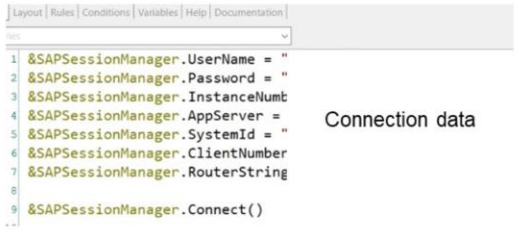
Anteriormente, analisamos detalhadamente os parâmetros e objetos a serem importados pela bapi, portanto não entraremos em detalhes agora.

Importamos. E, como já sabemos, os objetos ficarão no módulo Enterprise.

Finalizado o processo, se abrirmos o módulo Enterprise veremos o conjunto de novos SDTs e um novo objeto externo sapCustomerOrder.

Este objeto externo tem a propriedade SalesDocument e o método de criação com seu conjunto de parâmetros.

Trabalhando com Ordens de Venda



Como antes, vamos criar um procedimento que receberá o número de pedido criado por nossa aplicação, e irá tentar criá-lo no ERP de SAP. Caso a criação seja bem-sucedida, o procedimento devolverá o número de SaleOrder retornado pelo SAP. E em caso de existir alguma falha, devolverá o erro.

O número retornado pelo SAP será registrado no novo atributo SaleOrderSAPNumber, e a mensagem obtida do SAP no momento da criação do pedido será registrada no atributo SaleOrderSAPLastMsg.

Bom, criamos então o procedimento CreateSaleOrderSAP, e na aba Rules definimos a regra Parm para receber o identificador da ordem de venda.

Definimos em primeiro lugar os dados de conexão e nos conectamos com o ERP de SAP.

Se não houver erros na conexão, é gerado o cabeçalho da ordem

Estes dados dependem da implantação do SAP em uma empresa. É criado um Id para o tipo de pedido, neste exemplo "TA", se indica a organização de venda, o canal de distribuição, etc. O mesmo ocorre com o Cliente:'AG' é o código que indica que este é um Cliente.

Então neste For each, é percorrida cada linha da ordem de venda, e para cada linha é criada uma para o produto e outra para a entrega

Trabalhando com Ordens de Venda

```
&SAPSessionManager.TransactionBegin()

&sapCustomerOrder.CREATEFROMDAT2(&SALESDOCUMENTIN,&BAPISDHD1, &BAPISDHD1X, &BAPI_SENDE

&SAPSessionManager.TransactionCommit()


&SaleOrderSAPNumber.SetEmpty()

for &BAPIRET2Row in &BAPIRET2
     if &BAPIRET2Row.TYPE <> 'S' and &BAPIRET2Row.NUMBER = 311
          // Save the document number created
          &SalesOrderSAPNumber = &BAPIRET2Row.MESSAGE_V2
     endif
endfor

if not &SaleOrderSAPNumber.IsEmpty()
     for each
          where SaleOrderId = &SaleOrderId
          SaleOrderSAPNumber = &SaleOrderSAPNumber
     endfor
else
     for each
          where SaleOrderId = &SaleOrderId
          SaleOrderSAPLastMsg = &BAPIRET2.ToXml()
     endfor
endif
```

Bom, então para criar ou modificar um dado no ERP de SAP através de uma BAPI, deve ser utilizado o método TransactionBegin e TransactionCommit. Algumas BAPIs não realizam commit para a base de dados, portanto, deve ser disparado de forma explícita.

No meio, é realizada a criação do pedido, e para isso devemos utilizar a variável &sapCustomerOrder, definida com base na BAPI, pois trata-se de um método de instância.

Então é analisada a resposta da BAPI para obter o número do SAP, e são armazenados os valores correspondentes nos atributos SaleOrderSAPNumber e SaleOrderSAPLastMsg

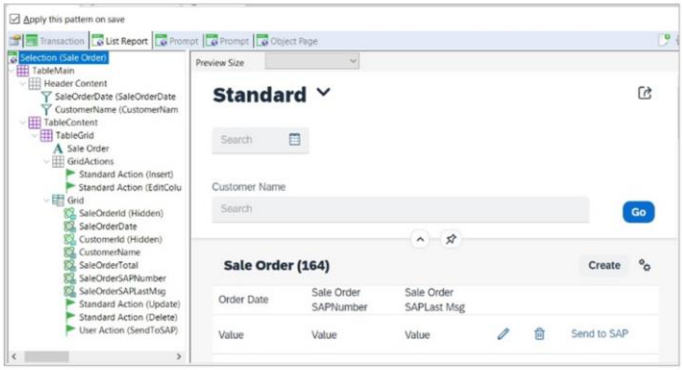
Finalmente, são carregados os erros caso tenham sido gerados.

Trabalhando com Ordens de Venda

```
CreateSaleOrderSAP(SaleOrderId) if Insert on AfterComplete;

noaccept(SaleOrderSAPNumber);

noaccept(SaleOrderSAPLastMsg);
```



Bem, uma vez finalizado este procedimento, devemos chamá-lo a partir das regras da transação SaleOrder de forma que para cada nova ordem que seja gerada, e depois de ser registrada nas tabelas de nossa aplicação, a mesma seja criada no ERP de SAP.

Para isso condicionamos a chamada ao modo de execução insert da transação e o condicionamos ao momento on AfterComplete, ou seja, após ser realizado o commit da transação.

Mas caso seja gerado aqui algum erro, queremos ter a possibilidade de tentar o envio novamente, por isso vamos personalizar nossa tela de trabalhar com ordens de vendas.

Lembremos que havíamos aplicado o floorplan ListReport ao aplicar o pattern Fiori for web a esta transação. Então vamos para a aba Pattern, e na tab ListReport, adicionamos os novos atributos no nível do grid.

E adicionamos também uma user action que chamamos de SendToSAP

Queremos definir esta ação como uma imagem, e para isso marcamos a propriedade Control Type como Image. Atribuímos o nome e também o código de um ícone

Trabalhando com Ordens de Venda

```
Event &SendToSAP.Click
    if SaleOrderSAPNumber.IsEmpty()
        CreateSaleOrderSAP(SaleOrderId)
    endif

    Grid.Refresh()

EndEvent
```

Event Grid.Load

```
if SaleOrderSAPNumber.IsEmpty()
    &SendToSAP.TooltipText = 'Send SO to SAP'
else
    &SendToSAP.TooltipText = 'SO alreday in SAP'
Endif
```

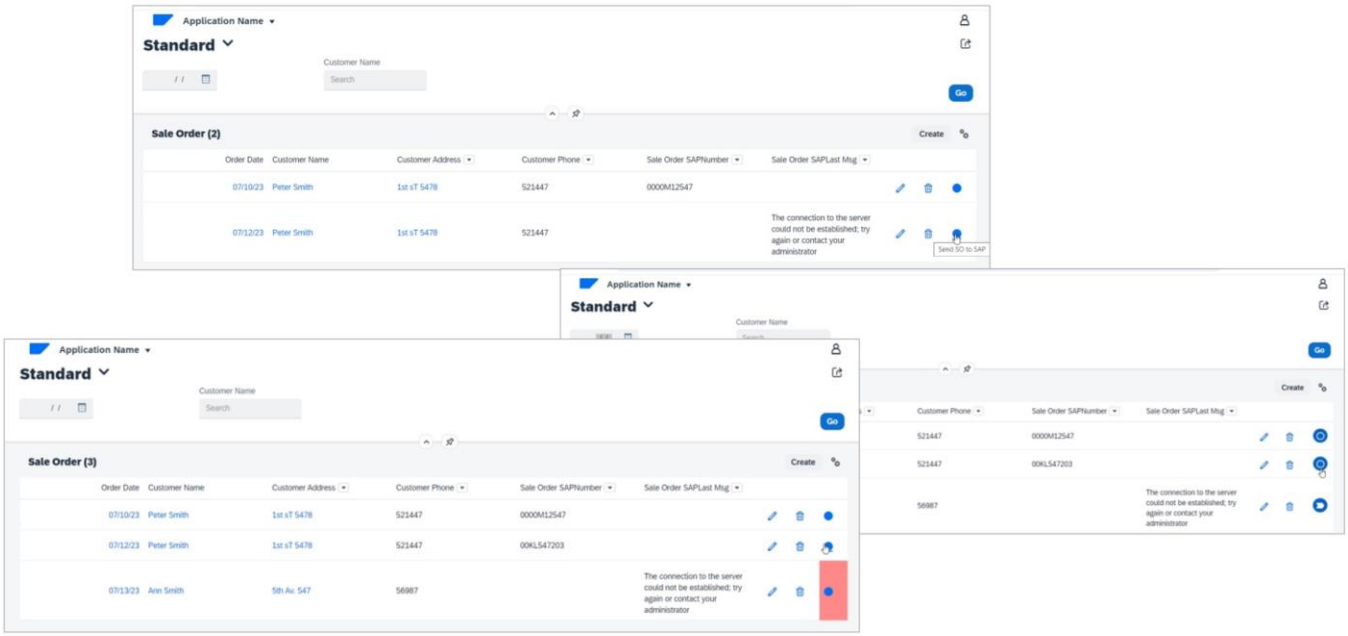
Abrimos o objeto e vamos para os eventos. No evento associado, e no caso em que o atributo SaleOrderSAPNumber esteja vazio, chamamos o procedimento de criação de pedido no SAP.

Como dissemos anteriormente, nosso objetivo é poder tentar novamente o envio a qualquer momento.

Por tratar-se de um evento de usuário, definido em um web panel, que afeta uma linha específica do grid, devemos provocar o Refresh depois dele.

Bom. Se formos agora ao evento Load, podemos alterar o tooltip text, de acordo com o status do pedido, ou seja, se a ordem já está criada no SAP ou está pendente de envio.

Trabalhando com Ordens de Venda



Para ver em execução, pressionamos F5.

Vemos as ordens de venda, com seu status atual. Enviamos uma das pendentes e vemos o número devolvido por SAP.

Também poderíamos, por exemplo, alterar a cor de fundo de acordo com o status de cada ordem, alterar os ícones, etc.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com