GeneXus by Globant

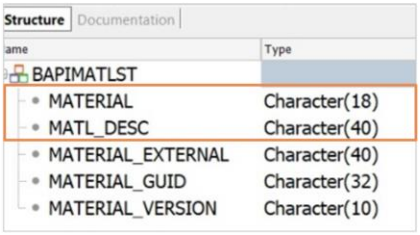
Integrando a KB ao SAP ERP

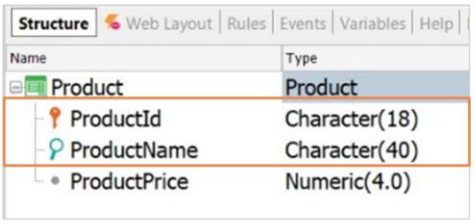
Executando o método GetList

Anteriormente, estabelecemos a conexão com o ERP de SAP e invocamos o método GetList da Bapi de Materiais para receber a lista de materiais.

Nosso objetivo agora é percorrer esta lista e despejá-la na tabela Produto de nossa aplicação.

Executando o método GetList





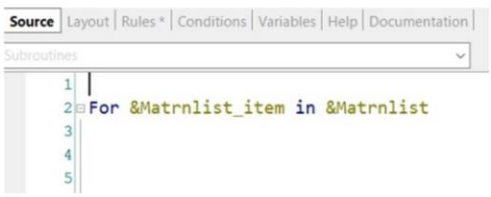
Em primeiro lugar observemos que na definição de nossa transação Produto o Id é autonumerado, e o nome é Character de comprimento 20.

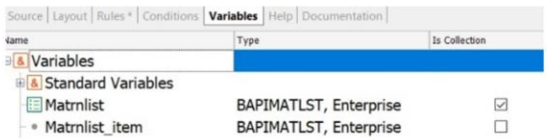
Mas se olharmos agora para a estrutura do SDT BAPIMATLIST veremos que o número associado ao material está definido como um Character de comprimento 18 e sua descrição como Character de 40.

Por isso modificamos a estrutura de nossa transação para que seja compatível com estas definições.

Bom. Devemos agora definir o processo que se encarregue de percorrer a lista de materiais recebida do ERP e despejá-la na tabela de produto.

Executando o método GetList





Criamos um procedimento chamado UserUpdateProducts.

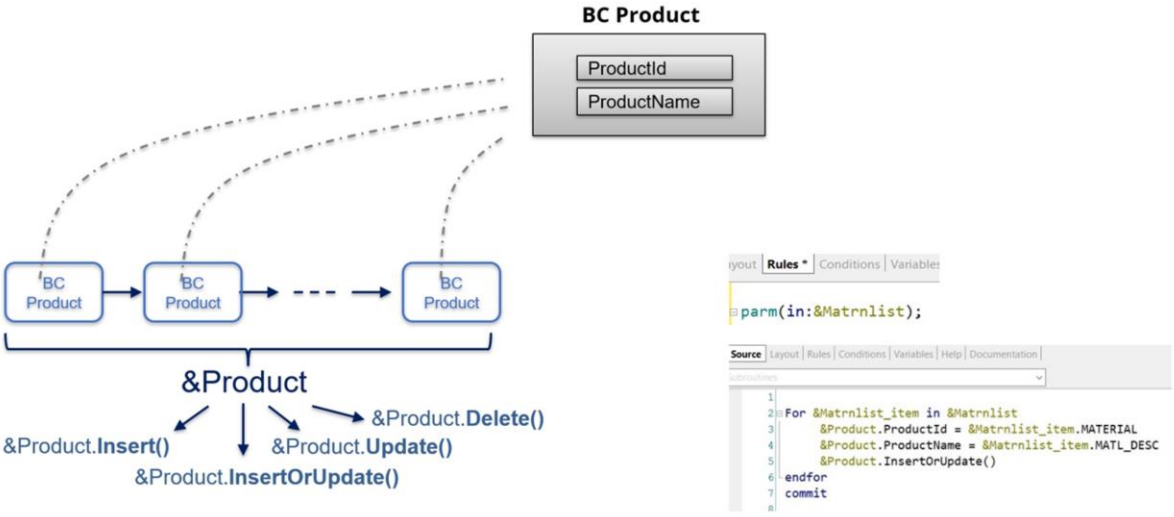
Este procedimento deve receber a lista de materiais retornada pela execução do método GetList, então definimos a variável &MatRnList do tipo de dado BapiMatList, e a marcamos como coleção.

Vamos agora para a aba Rules e definimos a correspondente regra Parm. Precisamos percorrer esta coleção de materiais, então precisamos definir outra variável do tipo BapiMatList mas dessa vez para representar um elemento da coleção, então a definimos simples e não como coleção. Bom. Vamos agora ao source para percorrer a lista de materiais:

E para cada um desses elementos, devemos inseri-lo como um produto novo em nossa tabela, se não existir anteriormente, ou atualizá-lo se já existir.

Portanto, declaramos nossa transação Produto como Business Component e utilizaremos o método InsertOrUpdate para processar os registros.

Business Component: Product



Lembremos que Business Component é uma propriedade das transações que permite disparar sua lógica fora da transação

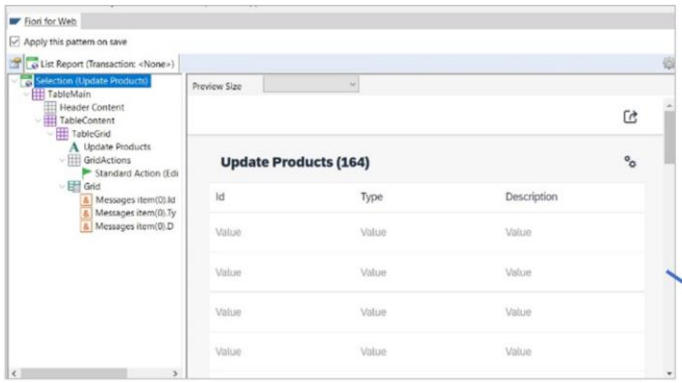
Os métodos que estamos vendo se aplicam tanto a variáveis simples como coleção de Business Components. Em particular, o método InsertOrUpdate tem um comportamento específico, pois primeiro tenta fazer o insert do registro e, se falhar devido a uma chave duplicada, então ele o recupera e o atualiza.

Este é o comportamento que nos ajuda a resolver nosso requisito, pois nosso objetivo é manter sempre atualizada nossa tabela de produtos, inserindo os novos ou atualizando os existentes.

Então, voltamos ao procedimento e definimos a variável &Product, Business Component. E no source completamos o código.

Uma vez atribuídos os valores para o Id e nome do produto, aplicamos o método InsertOrUpdate. Fechamos o for e declaramos o commit, necessário ao trabalhar com Business Components.

Executando o método GetList



Evento Start

```
SAPMaterialGetList(&Matrnlist, &Messages)

if &Messages.Count = 0
     UpdateMyProducts(&Matrnlist)
     msg("Products were updated")
else
     msg("Errors in the execution")
endif
```

Bom. Atingido este ponto, precisamos agora de um objeto que dispare esses processos. Portanto, criamos um web panel chamado UpdateUserProducts e aplicamos o pattern Fiori for Web a ele. Escolhemos o floorplan ListReport, e vamos indicar que carregamos seus dados a partir de um SDT.

Lembremos que o processo SAPMaterialGetList retorna a lista de materiais e a lista de possíveis erros ocorridos. A lista de materiais operamos internamente para atualizar nossos produtos, então tomamos o sdt Messages como origem de dados. Este web panel então exibirá a coleção de erros, se houver algum registrado, e processará a lista de materiais.

Removemos as ações sobre o grid e verificamos as propriedades correspondentes para incluí-lo no Fiori launchpad e no menu da master page.

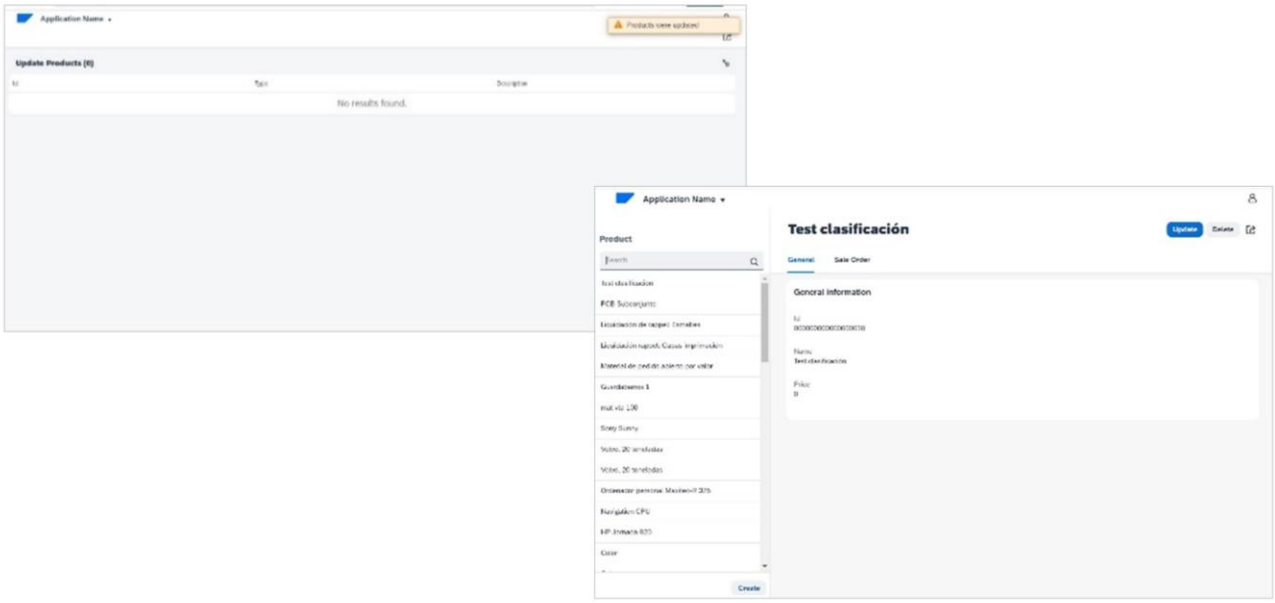
Gravamos. E vamos agora aos seus eventos.

Queremos que ao abrir este objeto seja chamado o processo que invoca o método GetList, sejam exibidos os erros, se houver, e seja processada a lista de materiais.

Assim, no evento Start definimos a chamada ao procedimento SAPMaterialGetList, definindo as variáveis necessárias.

Em seguida, consultamos se houve erros ou não e processamos os materiais mostrando mensagens que indicam se houve falhas ou se foi realizado corretamente.

Executando o método GetList



Antes de executar, nos certificamos de remover os dados de teste para carregar os produtos a partir dos materiais devolvidos pelo ERP.

Pressionamos F5

Primeiro verificamos que não temos nenhum produto carregado. E agora vamos ao web panel para carregar os materiais devolvidos pelo ERP.

Vemos que foi executado corretamente e não há retorno de erros ou avisos.

E finalmente voltamos aos produtos para verificar que foram carregados corretamente.

A seguir, trabalharemos com as ordens de venda.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com