Manual de Integração Protheus

Resumo

Para a integração entre o Protheus e o Anymarket, foram desenvolvidas CLASSES em linguagem advpl, sendo estas responsáveis pela comunicação, inserção, consulta e atualização de dados através da API do Anymarket.

Estas CLASSES acessam o Anymarket através do padrão de comunicação Rest, sendo necessário um Token enviado para cliente para este acesso.Este Token deverá ser inserido na classe AnyAcesso na propriedade cToken. As outras classes são classes filhas da AnyAcesso, ou seja, herdam todas as propriedades e métodos da classe pai.

Ao receber as informações do Anymarket, as classes recebem um JSON, converte para um objeto e em seguida insere os dados em suas propriedades.

Ao enviar os dados para a Anymarket, as propriedades das classes são convertidas em uma string no padrão JSON e enviadas para o Anymarket. Se houver algum erro na string, é enviado um e-mail para o cliente com a URL, com o JSon e com o erro. É necessário criar o parâmetro MV_YANYEML no Protheus e cadastrar o e-mail do responsável.

As funções de envio de dados pelo Protheus possuem algumas limitações, elas não enviam descrições com caracteres especiais e por isso é feito um tratamento nos campos de descrições, alterando estes caracteres. Ex: O caractere “à” é alterado para “a”, o caractere “º” é alterado para “.”.

Classes

Classe Acesso

Esta classe tem o objetivo de fazer a comunicação entre o Protheus e o Anymarket. É nesta classe que o cliente irá inserir o Token e a URL base do Anymarket.

O cliente deverá alterar as propriedades cToken e cURLBase enviadas pelo Anymarket. Hoje a propriedade cURLBase está com a Url da versão 2 da API no Anymarket. Os métodos e a propriedade aHeadStr não devem ser alterados. A propriedade aHeadStr já está configurada e os métodos são usados pelos outras classes. Todas as outras classes desenvolvidas herdam desta classe.

Arquivo: AnyAcesso.prw

Propriedades da classe.

1. cToken : Propriedade para a inserção do Token enviado pela Anymarket
2. cURLBase: Url Base para comunicação com o Anymarket.
3. aHeadStr: Array com as chaves de comunicação para autenticação.

Métodos.

1. New: Método construtor da classe
2. Conectado: Método que analisa se há comunicação entre o servidor Protheus e o Anymarket.

3. EmailErro: Método de envio de e-mails, caso haja uma falha de comunicação e a requisição enviada para o Anymarket não seja entregue.

Classe Produto.

O objetivo desta classe é cadastrar produtos e skus, alterar, consultar, atualizar preço e estoque e obter a lista com todos os produtos cadastrados no Anymarket.

Arquivo: AnyProduto.prw

Propriedades da Classe:

1. cCodigo: Código do produto no Protheus. Este código não poderá ter espaços.
2. cIdWeb: Código no formato string do produto no Anymarket.
3. cIdSku: Código no formado string do sku no Anymarket.
4. cDesc: Descrição do produto. Está descrição não poderá ter caracteres especiais por limitação do Protheus.
5. nPeso: Peso do sku
6. nLargura: Largura do sku
7. nComprimento: Comprimento do sku
8. nAltura: Altura do sku
9. nTempGarantia: Tempo de garantia em meses.
10. cDesGarantia: Descrição para a garantia;
11. nMarkup: Markup do produto.
12. nTipoPreco: Tipo do preço (1=Manual, 2=Automático)
13. cInformacoes: Informações do produto no formato html. O sistema alterará automaticamente os enters do texto para o padrão html.
14. cOrigem: origem do produto (0 = Nacional, 1 - Importado)
15. aSkus: Array com os Skus do produto. Os dados deste array são dados da classe AnySku.
16. aCaracte: Array com as características do produto. Os dados deste array são dados da classe AnyCaracte.
17. cModelo: Modelo do produto
18. cNBM: Nbm do produto.
19. cIdCateg: Código da categoria cadastrada no Anymarket.
20. cDescCate: Descrição da categoria. A descrição da categoria não poderá ter caracteres especiais por limitações do Protheus.
21. cIdMarca: Código da marca cadastrada no Anymarket.
22. cDescMarca: Descrição da marca. A descrição da marca não poderá ter caracteres especiais por limitações do Protheus.
23. aEstoques: Propriedade utilizada para atualizar o estoque dos produtos no Anymarket. A atualização do estoque é feita por lote, ou seja, é possível enviar vários produtos de uma vez na atualização, diminuindo o tempo de atualização. Este é um array bidimensional contendo o código do sku e a quantidade a ser atualizada.

24. aPrecos: Propriedade utilizada para atualizar os preços dos produtos no Anymarket. A atualização dos preços é feita por lote, ou seja, é possível enviar vários produtos de uma vez na atualização, diminuindo o tempo de atualização. Este é um array bidimensional contendo o código do sku e o preço a ser atualizado.
25. aListaProd: Array com a lista de produtos já cadastradas no Anymarket. Este array é alimentado através do método AllProdutos e tem as seguintes propriedades: cIdWeb, cIdSku, cCodigo, cIdCateg e cIdMarca

Métodos

1. New: Método construtor da classe.
2. CriaProduto:Cria um novo produto no anymarket, caso o código do Sku não exista.
3. AddSku:Usada para criar um novo Sku no produto. É necessário criar a classe AnySku, informar suas propriedade e adicionar na propriedade.
4. AddCaracte:Cria uma nova característica do produto. O parâmetro recebido é a classe AnyCaracte.
5. GetProd:Localiza um produto cadastrado no AnyMarket através de uma das propriedades cIdWeb, cIdSku, cCodigo.
6. AtuaProduto:Atualiza o produto no AnyMarket através das propriedades informadas.
7. AllProdutos:Localiza todos os produtos cadastrados no Anymaket e adiciona-os na propriedade aListaProd.
8. AllAtuEstoques:Atualiza o estoque de todos os produtos cadastrados no Anymarket.
9. AllAtuPreco: Atualiza o preço de todos os produtos cadastrados no Anymarket.

Classe Sku.

O Objetivo esta classe é auxiliar a classe produtos na criação e atualização dos valores dos skus.

Arquivo: AnyProduto.prw

Propriedades:

1. cIdWeb: Código do Sku no Anymarket.
2. cDesc: Descrição do Sku.
3. cCodProd: Código do Sku cadastrado no Protheus.
4. cCodBarra: Código de barra.
5. nQde: Quantidade em estoque do Sku.
6. nPreco: Preço do Sku

Métodos:

1. New: Método construtor da classe.
2. Adiciona: Método para adicionar um novo Sku no produto.

Classe características.

O objetivo desta classe é cadastrar e alterar as características de um produto.

Arquivo: AnyProduto.prw

Propriedades:

1. cTitulo: Título da característica.
2. cValor: Valor da característica.

Ex: cTitulo := Potência

cValor := 10

Métodos:

1. New: Método construtor da classe.
2. Adiciona: Adiciona uma nova característica ao produto através das propriedades preenchidas pelo cliente.

Exemplos:

1. Criar um novo produto: Ver arquivo ESTCA001.prw
2. Atualizar o estoque de todos os produtos: Ver arquivo ESTCA002.prw
3. Atualizar o preço de todos os produtos: Ver aquivo ESTCA003.prw
4. Atualizar os produtos: Ver arquivo ESTCA004.Prw

Classe Categoria

Classe responsável por adicionar as categorias e subcategorias dos produtos.

Arquivo: AnyCategoria.prw

Propriedades:

1. cCodigo: Código da categoria no Protheus.
2. cCodPai: Código do categoria PAI no Protheus.
3. cIdWeb: Código da categoria no Anymarket.
4. cNome: Descrição da categoria
5. aAllCateg: Array com todas as categorias cadastradas no Anymarket. Este array possui as seguintes propriedades: cNome, cIdWeb, Classe com as subcategorias.

Métodos:

1. New: Método construtor da classe
2. GetCateg: Localiza uma categoria através do Id da categoria cadastrada no Anymarket. O Id é passado como parâmetro no método.
3. CriaCateg: Cria uma nova categoria no Anymarket de acordo com as propriedades informadas.

4. AtuaCateg: Atualiza os dados de uma categoria através das propriedades informadas.
5. AllCateg: Localiza todas as categorias e adiciona-as na propriedade aAllCateg.

Exemplos:

a) GetCateg:

```
oCateg      :=AnyCategoria():New()
lAchou      :=oCateg:GetCateg(aSubCat[nI,2])
If lAchou
      cId         :=oCateg:cIdWeb
EndIf
```

b) CriaCateg:

```
oCateg:cCodigo:=cCodCategoria
oCateg:cIdWeb:=cCodAny
oCateg:cNome:=cDescricao
oCateg:cCodPai:=cIdPai
oCateg:CriaCateg()
cId:=oCateg:cIdWeb
```

c) AtuaCateg:

```
oCateg      :=AnyCategoria():New()
lAchou      :=oCateg:GetCateg(aSubCat[nI,2])
If lAchou
      oCateg:cCodigo:=oCateg:cIdWeb
      oCateg:cNome:=cDescricao
      oCateg:cCodPai:=cCategPai
      oCateg:AtuaCateg()
EndIf
```

d) AllCateg:

```
oCateg      :=AnyCategoria():New()
oCateg:AllCateg
If nI := 1 to Len(oCateg:aAllCateg)
      cIdWeb  :=oCateg:aAllCateg[nI]:cIdWeb
      cCodigo:=oCateg:aAllCateg[nI]:cCodigo
      cNome:=oCateg:aAllCateg[nI]:cNome
      cCodPai:=oCateg:aAllCateg[nI]:cCodPai
EndIf
```

Classe Imagens

Classe responsável por adicionar uma nova imagem ao produto no Anymarket. Esta classe cadastra a URL da imagem. Ao cadastrar uma nova imagem, o sistema analisa se a URL já está cadastrada para o produto informado para não inserir em duplicidade.

Arquivo: AnyImagem.prw

Propriedades:

1. cIdWeb: Código do registro no Anymarket.
2. cIdProduto: Código do produto no Protheus.

3. cUrl: Url da imagem.

Métodos:

1. New: Método construtor da classe:
2. CriaImagem: Vincula um nova imagem ao produto de acordo com as propriedades informadas.

Exemplo:

```
AnyImagem():New()
oImagem:cIdProduto      :=Produto
oImagem:cURL            :=URL
oImagem:CriaImagem()
cIdWeb:=oImagem:cIdWeb
```

O exemplo também está no arquivo ESTCA001.prw

Classe Marca

Classe responsável por adicionar uma nova marca no Anymarket. Ao cadastrar uma nova marca, o sistema analisa se a marca já existe para não inserir em duplicidade.

Arquivo: AnyMarca.prw

Propriedades:

1. cCodMarca: Código do registro no Protheus.
2. cNome: Descrição da marca.
3. cIdWeb: Código da marca cadastrado no Anymarket
4. aListaMarcas: Array com todas as marcas cadastradas no Anymarket. Ao instanciar a classe, o array é preenchido. Este array possui as seguintes propriedades: IdWeb, cNome e cCodMarca.

Métodos:

1. New: Método construtor da classe.
2. CriaMarca: Criar uma nova marca de acordo com as propriedades informadas.
3. AtuaMarca: Atualiza uma marca de acordo com os parâmetros informados.
4. AllMarcas: Lista todas as marcas a adiciona-as na propriedade aListaMarcas.

Exemplo:

a) CriaMarca:

```
oMarca              :=AnyMarca():New()
oMarca:cCodigo      :=cCodigo
oMarca:cNome        :=cNome
oMarca:CriaMarca()
cId                 :=oMarca:cIdWeb
```

Obs: O exemplo também está no arquivo ESTCA001.prw

b) AtuaMarca:

```
oMarca              :=AnyMarca():New()
oMarca:cCodigo      :=cNome
oMarca:cNome        :=cCodigo
```

```
oMarca:AtuaMarca()
cId                :=oMarca:cIdWeb
```

ou

```
oMarca             :=AnyMarca():New()
oMarca:cIdWeb      :=cIdWeb
oMarca:cNome       :=cCodigo
oMarca:AtuaMarca()
```

c) AllMarcas:

```
oMarca      :=AnyMarca():New()
oMarca:AllMarcas
IfnI := 1 to Len(oMarca:aListaMarcas)
      cIdWeb  :=oMarca:aListaMarcas[nI]:cIdWeb
      cCodigo:=oMarca:aListaMarcas[nI]:cCodigo
      cNome:=oMarca:aListaMarcas[nI]:cNome
EndIf
```

Classe Pedido

Classe responsável por manipular os pedidos no Anymarket. Através desta classe é possível obter os pedidos através do Feed, fazer a notificação dos pedidos, enviar informações de Tracking, alterar o status do pedido, receber notificações de alterações de pedidos (necessário configurar uma URL e enviar para o Anymrket), etc:

Arquivo: AnyPedido.prw

Propriedades:

1. aPedidos: Lista dos pedidos obtidos do feed. Por padrão o Anymarket envia até 10 pedidos.

Métodos:

1. New: Método construtor da classe.
2. GetAllPedidos: Obtém os pedidos do feed.

Exemplo:

```
oAnyPedido  :=AnyPedido():New()
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)

Next
```

Obs.: Esta classe é usada em conjunto com a classe AnyPedId e se encontra no mesmo arquivo de código:

Classe AnyPedId

O objetivo desta classe é obter os dados e um pedido. Ela trabalha em conjunto com a classe AnyPedidos.

Arquivo: AnyPedido.prw

Propriedades:

1. cMarketPlace: Nome do MarketPlace.
2. cIdWeb: Código do pedido no Anymarket.
3. dDataEmissao: Data da criação do pedido no Anymarket.
4. cStatus: Status do pedido.
5. nVlrFrete: Valor do frete.
6. cMPlaceId: Id no MarketPlace.
7. cNome: Nome do cliente.
8. cCliMP:Código do cliente no Marketplace.
9. cCliAny: Código do cliente no Anymarket.
10. cDocumento: CPF ou CNPJ do cliente.
11. cTipoDoc: Tipo de documento “CPF” ou “CNPJ”.
12. cMunicipio: Município do cliente.
13. cUf: Unidade Federativa do cliente.
14. cPais: País do cliente.
15. cEndereco: Endereço do cliente.
16. cCep: Cep do endereço do cliente.
17. cTelefone: Telefone para contato do cliente.
18. cEmail: E-mail do cliente.
19. aFormPtgo: Array com as formas de pagamento do cliente.
20. aFormaEnvio: Array com as formas de envio da mercadoria para o cliente.
21. aItems: Itens do pedido de vendas.
22. nDesconto: Desconto
23. cComplemento: Complemento do endereço.
24. cBairro: Bairro do cliente.
25. cNumero: Número do endereço do cliente.
26. cUrlRastro: Url para rastro da mercadoria
27. cRastreio: Número de rastreio
28. cFormaEnv: Forma de envio do pedido.
29. cNota: Número da nota fiscal.
30. cSerie: Serie da nota fiscal.
31. cChaveNfe: Chave da nota fiscal eletrônica.
32. dData: Data de faturamento.
33. cHora: Hora do faturamento.
34. dDataEnvio: Data de envio da mercadoria.
35. choraEnvio: Hora do envio da mercadoria.
36. dDataEntrega: Data da entrega da mercadoria.
37. cHoraEntrega: Hora da entrega da mercadoria.
38. dDataPrevisao: Data da previsão de entrega da mercadoria.
39. cHoraPrevisao: Hora da previsão de entrega da mercadoria.
40. dDataTransp: Data de postagem
41. cHoraTransp: Hora da postagem

Métodos:

1. New: Método construtor da classe.
2. GetPedido: Localiza um pedido no Anymarket através do Id do pedido cadastrado no Anymarket.
3. Notificar: Método responsável por notificar um pedido de vendas para não ser apresentado novamente no feed. Obs: Caso seja alterado o status do pedido, o mesmo voltará a ser relacionado no feeds.
4. Concluir: Altera o status do pedido para concluído.
5. Cancelar: Altera o status do pedido para cancelado.
6. Enviar: Altera o status do pedido para enviado.
7. Faturar: Alterar o status do pedido para faturado.
8. Tracking: Envia as informações da nota e da forma de envio da mercadoria.
9. GetPedCall: Obtém o código do pedido no Anymarket enviado pelo CallBack de pedidos (necessário configurar uma URL e enviar para o Anymrket).

Exemplos:

a) Localizar um pedido:

```
cOrderID    :=2713

If oPedido:GetPedido(AllTrim(cOrderID))

EndIf
```

b) Notificar:

```
oAnyPedido  :=AnyPedido():New()
oPedido     :=AnyPedId():New()
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)
      cOrderID := oAnyPedido:aPedidos[nI,1]
      cToken:=oAnyPedido:aPedidos[nI,2]
      ...
      oAnyPedido:Notificar(cToken)
Next
```

c) Concluir:

```
oAnyPedido  :=AnyPedido():New()
oPedido     :=AnyPedId():New()
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)
      cOrderID := oAnyPedido:aPedidos[nI,1]
      cToken:=oAnyPedido:aPedidos[nI,2]
      ...
      oPedido:Concluir()
      oAnyPedido:Notificar(cToken)
Next
```

d) Cancelar:

```
oAnyPedido  :=AnyPedido():New()
oPedido     :=AnyPedId():New()
```

```
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)
      cOrderID := oAnyPedido:aPedidos[nI,1]
      cToken:=oAnyPedido:aPedidos[nI,2]
      ...
      oPedido:Cancelar()
      oAnyPedido:Notificar(cToken)
Next
```

e) Enviar:

```
oAnyPedido  :=AnyPedido():New()
oPedido     :=AnyPedId():New()
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)
      cOrderID := oAnyPedido:aPedidos[nI,1]
      cToken:=oAnyPedido:aPedidos[nI,2]
      ...
      oPedido:Concluir()
      oAnyPedido:Enviar(cToken)
Next
```

f) Faturar:

```
oAnyPedido  :=AnyPedido():New()
oPedido     :=AnyPedId():New()
oAnyPedido:GetAllPedidos()
For nI:=1 to Len(oAnyPedido:aPedidos)
      cOrderID := oAnyPedido:aPedidos[nI,1]
      cToken:=oAnyPedido:aPedidos[nI,2]
      ...
      oPedido:Faturar()
      oAnyPedido:Notificar(cToken)
Next
```

g) Tracking:

```
oAnyPedido:GetPedido(cOrderID)
oAnyPedido:cRastreio    := cCodRastro
oAnyPedido:cFormaEnv    := cFormaEnio
oAnyPedido:cNota        := AllTrim(cvaltochar(val(SF2-
>F2_DOC))+iif(i>1,"-"+cvaltochar(i-1),""))
oAnyPedido:cSerie       := AllTrim(SF2->F2_SERIE)
oAnyPedido:dData        := SF2->F2_EMISSAO
oAnyPedido:cHora        := SF2->F2_HORA
oAnyPedido:cChaveNFe    := SF2->F2_CHVNFE
If AllTrim(SF2->F2_TRANSP) != ''
      oAnyPedido:cFormaEnv    := Posicione('SA4', 1,
Filial('SA4')+SF2->F2_TRANSP, 'SA4->A4_NOME')
EndIf
If oAnyPedido:Tracking()
      oAnyPedido:Fatura()
Else
      ...
EndIf
```

h) GetPedCall:

```
oPedido           :=AnyPedId():New()
If oPedido:GetPedCall(cJson)
      ...
EndIf
```