|  |  |  |  |
| --- | --- | --- | --- |
| CLIENTE | |  |  | | --- | --- | |  | **Rua Robert Bosch, 544**  **Parque Industrial Tomas Edson,**  **São Paulo - SP, 01144-020** | |
| PROJETO | INTEGRAÇÃO DE ENTIDADES TELECONTROL X PROTHEUS |
| CONTATO | JOANACIO LIMAjoalima@mdxtelecom.com.br |
| RESPONSÁVEL TWOIT | EMERSON NASCIMENTO |
| DATA | 06/05/2025 |
| ELABORADO POR | EMERSON NASCIMENTO |

**INTEGRAÇÃO DE CADASTROS**

**PROTHEUS TELECONTROL**

|  |
| --- |
| **Configurações** |

Para que as APIs de cadastros funcionem, algumas configurações precisam ser efetuadas.

Essas configurações são feitas em arquivos json e a partir de parâmetros.

Os arquivos json indicam os endereços das APIs, dos endpoints, como os dados devem ser montados para serem enviados e como o retorno deve ser tratado, entre outras configurações.

Por padrão os arquivos serão gravados dentro da pasta *system* do Protheus, na pasta *@telecontrol*.

Os arquivos de configuração de cadastros seguem uma nomenclatura padrão:

- xxxx\_get.json – indica como deve ser montado o arquivo para consulta do cadastro no Telecontrol (atualmente sem conteúdo)

- xxxx\_post.json – indica como deve ser montado o arquivo para inclusão de registro no Telecontrol

- xxxx\_put.json – indica como deve ser montado o arquivo para alteração de registro no Telecontrol

- xxxx\_show.json – indica como deve ser montada a visualização do registro consultado no Telecontrol

- endpoints.json – contém os endereços das APIs/endpoints do Telecontrol e as credenciais de acesso a essas APIs

Esses arquivos de configuração serão utilizados pelas rotinas/classes que montam as informações que serão enviadas às APIs do Telecontrol, acionados a partir de pontos de entrada do Protheus ou a partir de rotinas agendadas.

Classes desenvolvidas para integração dos cadastros:

- TTLCUtil – classe com métodos para criação de pastas

- TTLCLog – classe interna para manutenção do LOG de registros

- TTLCAPICadastro – classe interna com os métodos genéricos utilizados na integração dos cadastros

- TTLCPeca – classe especializada para cadastro de peças (herdada TTLCAPICadastro)

- TTLCPosto – classe especializada para cadastro de postos autorizados (herdada TTLCAPICadastro)

- TTLCPreco – classe especializada para cadastro de preço de peças (herdada TTLCAPICadastro)

- TTLCProduto – classe especializada para cadastro de produtos (herdada TTLCAPICadastro)

|  |
| --- |
| **Configurações – arquivo endpoints.json** |

O arquivo *endpoints.json* contém os endereços das APIs/endpoints do Telecontrol e as credenciais de acesso a essas APIs.

Exemplo de conteúdo do arquivo endpoints.json:

{

    "uri": "http://api2.telecontrol.com.br",

    "header": [

        {

            "key": "Access-Application-Key",

            "value": "{|| 'xxxxxxxxxxxxxx'}"

        },

        {

            "key": "Access-Env",

            "value": "{|| 'HOMOLOGATION'}"

        },

        {

            "key": "Content-Type",

            "value": "{|| 'application/json'}"

        }

    ],

    "produtos": {

        "post": {

            ["uri": "http://api2.telecontrol.com.br",]

            "description": "Cadastro de produto",

            "endpoint": "{|| '/posvenda-core/produtos'}"

        },

        "put": {

            "description": "Atualização de produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + cConsulta}"

        },

        "get": {

            "description": "Consulta produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + cConsulta}"

        }

    }

}

Campos obrigatórios:

uri – indica o endereço das APIs do Telecontrol.

    "uri": "http://api2.telecontrol.com.br"

header – indica como será montado o cabeçalho da requisição.

    "header": [

        {

            "key": "Access-Application-Key",

            "value": "{|| 'xxxxxxxxxxxxxx'}"

        },

        {

            "key": "Access-Env",

            "value": "{|| 'HOMOLOGATION'}"

        },

        {

            "key": "Content-Type",

            "value": "{|| 'application/json'}"

        }

    ]

    O header deve ser criado utilizando os conjuntos de valores key e value.

    Será necessário obter os conteúdos de key e value junto à Telecontrol.

Campos opcionais:

    "produtos": {

        "post": {

            "description": "Cadastro de produto",

            "endpoint": "{|| '/posvenda-core/produtos'}"

        },

        "put": {

            "description": "Atualização de produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + cConsulta}"

        },

        "get": {

            "description": "Consulta produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + cConsulta}"

        }

    }

Os opcionais são os dados dos endpoints utilizados para os cadastros.

Onde se lê “produtos”, pode ser:

“produtos” – para indicar os dados do endpoint do cadastro de produtos

“pecas” – para indicar os dados do endpoint do cadastro de peças

“postos” – para indicar os dados do endpoint do cadastro de postos autorizados

“precos” – para indicar os dados do endpoint do cadastro de preços

“faturamento” – para indicar os dados do endpoint que receberá as informações de faturamento

“ordens\_servico” – para indicar os dados do endpoint que disponibilizará as ordens de serviço

“pedidos” – para indicar os dados do endpoint que disponibilizará os pedidos

Para o correto funcionamento dos endpoints, é necessário informar cada ‘verbo’ disponibilizado, indicando a descrição e o endpoint em si.

        "post": {

            "description": "Cadastro de produto",

            "endpoint": "{|| '/posvenda-core/produtos'}"

        },

Acima temos a configuração para o endpoint de gravação de um registro, cujo verbo é o post.

- “description” é um campo informativo, com a descrição do endpoint/verbo.

- “endpoint” é onde deve ser informado o endpoint em si, que será utilizado em conjunto com o campo “uri” principal.

Caso o endpoint utilize uma uri específica, essa uri deve ser informada dentro da configuração do verbo em questão. Por exemplo:

        "put": {

            "uri": "http://apinova.telecontrol.com.br",

            "description": "Atualização de produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + cConsulta}"

        },

Da forma como apresentado acima, o endpoint do verbo put irá utilizar a uri "http://apinova.telecontrol.com.br", enquanto o endpoint do verbo post (um pouco acima) utilizará a uri geral "http://api2.telecontrol.com.br".

Há uma condição especial nos verbos de alteração e consulta, que podem necessitar do envio de parâmetros diretamente no endpoint.

Isto pode ser observado no verbo put, que precisa que seja informada a chave do registro que será alterado.

        "put": {

            "uri": "http://apinova.telecontrol.com.br",

            "description": "Atualização de produto",

            "endpoint": "{|| '/posvenda-core/produtos/referencia/' + **cConsulta**}"

        }

A montagem do endereço do endpoint pode utilizar até 2 parâmetros, de nomes cConsulta e cConsulta2. O conteúdo desses parâmetros será utilizado ao montar o endereço do endpoint.

|  |
| --- |
| **Configurações – arquivo xxxx\_get.json** |

O arquivo *xxxx\_get.json* precisa existir na pasta, mas atualmente não necessita de conteúdo.

Um conteúdo seria necessário se a API de consulta do Telecontrol necessitasse de dados na requisição da consulta, o que não é o caso; basta enviar o código da entidade diretamente no endereço do endpoint.

Caso a consulta seja efetuada com sucesso, será retornado o código 200 juntamente com o conteúdo do registro.

|  |
| --- |
| **Configurações – arquivo xxxx\_post.json** |

O arquivo *xxxx\_post.json* contém as informações para montagem do conteúdo do registro que deverá ser cadastrado no Telecontrol.

A montagem desse arquivo requer a indicação do campo do campo que o Telecontrol espera receber e do conteúdo que deverá ser gravado no campo. O nome do campo e o tipo do conteúdo são fornecidos pelo Telecontrol. Para configurar um campo, o padrão é: “campo”: “bloco de código ADVPL”

Exemplo:

"cnpj": "{|| ALLTRIM(SA2->A2\_CGC)}"

O exemplo acima vai gravar no campo cnpj o conteúdo obtido de {|| ALLTRIM(SA2->A2\_CGC)}, que no caso é o CNPJ do fornecedor que estiver posicionado.

Segue modelo com o cadastro de um preço de peça:

{

    "siglaTabela": "{|| ALLTRIM(DA1->DA1\_CODTAB)}",

    "pecaReferencia": "{|| ALLTRIM(DA1->DA1\_CODPRO)}",

    "preco": "{|| DA1->DA1\_PRCVEN}",

    "returnID": "{|| cValToChar(objRetorno['tabelaItem'])}"

}

No modelo acima podemos observar o campo returnID, que é um campo especial onde pode ser obtido o ID do registro gravado no Telecontrol. Esse ID pode ser gravado no Protheus para que seja criado o vínculo do cadastro do Protheus com o cadastro do Telecontrol. O conteúdo do campo returnID depende do endpoint que estiver sendo utilizado. No caso do endpoint de preços de peças, o conteúdo deve ser tabelaItem, que é devolvido pelo endpoint de cadastro de preços caso a gravação seja efetuada com sucesso.

Exemplo do resultado da gravação do preço, cujo código de retorno será 200 (sucesso ao gravar):

{

**"tabelaItem": 11121212**,

"tabela": {

"tabela": 850,

"siglaTabela": "PADRAO",

"descricao": "Tabela Padrão"

},

"peca": {

"peca": 2030434,

"descricao": "2600mAh Power Battery Micro USB -",

"referencia": "PW-M0190",

"ipi": "5",

"multiplo": 1,

"devolucaoObrigatoria": "false",

"acessorio": "false",

"itemAparencia": "false",

"origem": "IMP",

"ativo": "true"

},

"preco": "12.5"

}

No caso acima, o código 11121212 será atribuído ao registro cadastrado no Telecontrol e devolvido, para que seja tratado por quem solicitou o cadastro.

|  |
| --- |
| **Configurações – arquivo xxxx\_put.json** |

O arquivo *xxxx\_put.json* contém as informações para montagem do conteúdo do registro que deverá ser alterado no Telecontrol.

A montagem desse arquivo requer a indicação do campo do campo que o Telecontrol espera receber e do conteúdo que deverá ser gravado no campo. Segue exatamente as mesmas regras do arquivo *xxxx\_post.json*. O nome do campo e o tipo do conteúdo são fornecidos pelo Telecontrol.

Para configurar um campo o padrão é:

“campo”: “bloco de código ADVPL”

Exemplo:

"cnpj": "{|| ALLTRIM(SA2->A2\_CGC)}"

Segue modelo com a alteração de um preço de peça:

{

    "preco": "{|| DA1->DA1\_PRCVEN}"

}

Caso a alteração seja efetivada será obtido o código retorno 200, juntamente com o conteúdo do registro alterado.

Exemplo de retorno da gravação do preço:

{

**"tabelaItem": 11121212**,

"tabela": {

"tabela": 850,

"siglaTabela": "PADRAO",

"descricao": "Tabela Padrão"

},

"peca": {

"peca": 2030434,

"descricao": "2600mAh Power Battery Micro USB -",

"referencia": "PW-M0190",

"ipi": "5",

"multiplo": 1,

"devolucaoObrigatoria": "false",

"acessorio": "false",

"itemAparencia": "false",

"origem": "IMP",

"ativo": "true"

},

"preco": "12.5"

}

|  |
| --- |
| **Configurações – arquivo xxxx\_show.json** |

O arquivo *xxxx\_show.json* contém as informações para montagem da visualização de um registro cadastrado no Telecontrol (ao efetuar uma consulta).

A montagem desse arquivo requer a indicação do campo do campo que o Telecontrol devolve, juntamente com a configuração de como o conteúdo desse campo deve ser apresentado. O nome do campo e o tipo do conteúdo são fornecidos pelo Telecontrol.

Exemplo de configuração do registro do posto autorizado:

{

    "tipo\_posto": {

        "tag": "descricao",

        "titulo": "{|| 'Tipo Posto'}",

        "dica": "{|| 'Tipo do posto'}",

        "tipo": "{|| 'C'}",

        "tamanho": "{|| 20}",

        "decimal": "{|| 0}",

        "cbox": "{|| ''}"

    },

    "cnpj": {

        "referencia": "A2\_CGC"

    },

    "posto": {

        "titulo": "{|| 'ID Telecontrol'}",

        "dica": "{|| 'ID do posto no Telecontrol'}",

        "tipo": "{|| 'N'}",

        "tamanho": "{|| 15}",

        "decimal": "{|| 0}",

        "cbox": "{|| ''}"

    }

}

No arquivo acima podemos observar duas formas de configurar o conteúdo do retorno:

- detalhando o conteúdo ao passar cada informação de configuração

- indicando um campo referência do Protheus, de onde será obtida a configuração do campo

Caso deseje informar manualmente, a configuração segue o padrão “campo”: “bloco de código”

    "tipo\_posto": {

        "tag": "descricao",

        "titulo": "{|| 'Tipo Posto'}",

        "dica": "{|| 'Tipo do posto'}",

        "tipo": "{|| 'C'}",

        "tamanho": "{|| 20}",

        "decimal": "{|| 0}",

        "cbox": "{|| ''}"

    }

A configuração acima indica que o campo tipo\_posto, recebido do Telecontrol, será apresentado com o título ‘Tipo Posto’, o conteúdo será do tipo será caractere (os tipos são aqueles existentes no Protheus) com tamanho 20 e sem decimais; não terá lista de opções para montagem de combo box.

Para o caso de utilizar os dados de um campo existente no Protheus, utilize um campo como referência.

    "cnpj": {

        "referencia": "A2\_CGC"

    },

A configuração acima indica que o campo cnpj, recebido do Telecontrol, será apresentado utilizando as informações do campo A2\_CGC (CNPJ do Fornecedor) do Protheus. Desta forma o título, tipo, tamanho e demais informações utilizadas para apresentar o campo, serão obtidas do campo A2\_CGC.

|  |  |
| --- | --- |
| Ícone  O conteúdo gerado por IA pode estar incorreto. | Após a implantação serão criados os arquivos com conteúdo padrão, que poderão ser implementados em função de novas necessidades ou alterações nas APIs do Telecontrol. |

Exemplo de apresentação de um posto autorizado utilizando o arquivo *postos\_show.json* padrão:

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

|  |
| --- |
| **Classe TTLCUtil (TLC\_Classes.tlpp)** |

A classe TTLCUtil facilita a criação de pastas no disco. Ela cria toda a ‘ramificação’ das pastas, caso seja necessário.

Hierarquia

* TObject
  + TTLCUtil

Construtores

* Construtor TTLCUtil():New()

Métodos

* AddPathDelimiter
* RemovePathDelimiter
* CriaDir

Exemplos

User Function TUTIL()

Local oUtil := TTLCUtil:New()

Local cPath

cPath := oUtil:AddPathDelimiter(‘c:\temp’)

Alert(cPath + CRLF + oUtil:RemovePathDelimiter(cPath))

Return cPath

**TTLCUtil():AddPathDelimiter**

Sintaxe:

AddPathDelimiter( cPath )

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cPath | caractere | Caminho no qual deseja acrescentar o delimitador |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| cPath | caractere | Caminho com o delimitador adicionado |

**TTLCUtil():RemovePathDelimiter**

Sintaxe:

RemovePathDelimiter( cPath )

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cPath | caractere | Caminho no qual deseja remover o delimitador |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| cPath | caractere | Caminho com o delimitador removido |

**TTLCUtil():CriaDir**

Sintaxe:

CriaDir ( cPath )

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cPath | caractere | Caminho que deseja criar |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Retorno .T. indica que a pasta foi criada |

Exemplo

User Function TUTIL()

Local oUtil := TTLCUtil():New()

Local cPath := ‘c:\temp\exemplo1\pasta2’

If !oUtil:CriaDir(cPath)

Alert(‘O caminho ’ + cPath + ‘ não foi criado.’)

EndIf

Return cPath

|  |
| --- |
| **Classe TTLCPeca (TLC\_ClassePecas.tlpp)** |

A classe TTLCPeca efetua a integração de peças no Telecontrol.

Hierarquia

* TObject
  + TTLCAPICadastro
    - TTLCPeca

Construtores

* Construtor TTLCPeca():New()

Métodos

* Cadastra
* Consulta

Exemplos

User Function CadPeca()

Local oPeca := Telecontrol.Integracao.Peca.TTLCPeca():New()

SB1->(dbseek(xFilial() + ‘PRODUTO1’))

If !oPeca:Cadastra()

Alert(‘Erro ao integrar o produto ’ + SB1->B1\_COD)

EndIf

Return

**TTLCPeca():Cadastra**

Sintaxe:

Cadastra(xRegistro, cCodRet)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| xRegistro | caractere/numerico | Código do produto/peça ou número do registro da tabela SB1 que será enviado para cadastro no Telecontrol. Se nenhuma informação for passada, o registro posicionado será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se o cadastro foi efetuado |

**TTLCPeca():Consulta**

Sintaxe:

Consulta(cCodigo, cCodRet, cRetorno, lMostraMsg, lMostraReg)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cCodigo | caractere | Código da peça que será consultada. Se nenhuma informação for passada, o registro posicionado na tabela SB1 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |
| cRetorno | caractere | Json devolvido pela API |  |
| lMostraMsg | logico | Informa se as mensagens e alertas devem ser exibidos |  |
| lMostraReg | logico | Informa se o registro consultado deve ser apresentado na tela |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se a consulta foi efetuada com sucesso |

Exemplos

User Function ConPeca()

Local oPeca := Telecontrol.Integracao.Peca.TTLCPeca():New()

SB1->(dbseek(xFilial() + ‘PRODUTO1’))

If !oPeca:Consulta(,,,,.T.)

Alert(‘Erro ao consultar a peça ’ + SB1->B1\_COD)

EndIf

Return

|  |
| --- |
| **Classe TTLCPosto (TLC\_ClassePostos.tlpp)** |

A classe TTLCPosto efetua a integração dos postos autorizados no Telecontrol. O posto autorizado deve ser incluído no cadastro de fornecedores com o campo Tipo Posto (A2\_TPOSTO) preenchido.

Hierarquia

* TObject
  + TTLCAPICadastro
    - TTLCPosto

Construtores

* Construtor TTLCPosto():New()

Métodos

* Cadastra
* Consulta
* Altera

Exemplos

User Function CadPosto()

Local oPosto := Telecontrol.Integracao.Posto.TTLCPosto():New()

SA2->(dbseek(xFilial() + ‘POSTO1’))

If !oPosto:Cadastra()

Alert(‘Erro ao integrar o posto autorizado ’ + SA2->A2\_COD + ’/’ + SA2->A2\_LOJA)

EndIf

Return

**TTLCPosto():Cadastra**

Sintaxe:

Cadastra(xRegistro, cCodRet)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| xRegistro | caractere/numerico | Código do posto autorizado ou número do registro da tabela SA2 que será enviado para cadastro no Telecontrol. Se nenhuma informação for passada, o registro posicionado na tabela SA2 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se o cadastro foi efetuado |

**TTLCPosto():Consulta**

Sintaxe:

Consulta(cCodigo, cCodRet, cRetorno, lMostraMsg, lMostraReg)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cCodigo | caractere | Código do posto autorizado. Se nenhuma informação for passada, o registro posicionado na tabela SA2 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |
| cRetorno | caractere | Json devolvido pela API |  |
| lMostraMsg | logico | Informa se as mensagens e alertas devem ser exibidos |  |
| lMostraReg | logico | Informa se o registro consultado deve ser apresentado na tela |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se a consulta foi efetuada com sucesso |

Exemplos

User Function ConPosto()

Local oPosto := Telecontrol.Integracao.Posto.TTLCPosto():New()

SA2->(dbseek(xFilial() + ‘POSTO1’))

If !oPosto:Consulta(,,,,.T.)

Alert(‘Erro ao consultar o posto autorizado ’ + SA2->(A2\_COD + A2\_LOJA))

EndIf

Return

|  |
| --- |
| **Classe TTLCPreco (TLC\_ClassePrecos.tlpp)** |

A classe TTLCPreco efetua a integração dos preços de peças no Telecontrol.

Hierarquia

* TObject
  + TTLCAPICadastro
    - TTLCPreco

Construtores

* Construtor TTLCPreco():New()

Métodos

* Cadastra
* Consulta
* Altera

Exemplos

User Function CadPreco()

Local oPreco := Telecontrol.Integracao.Preco.TTLCPreco():New()

DA1->(dbseek(xFilial() + ‘TABPRC’ + ‘PECA1’))

If !oPreco:Cadastra()

Alert(‘Erro ao integrar o preço da peça ’ + DA1->DA1\_CODPRO + ‘ e tabela ’ + DA1->DA1\_CODTAB)

EndIf

Return

**TTLCPreco():Cadastra**

Sintaxe:

Cadastra(xRegistro, cCodRet)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| xRegistro | caractere/numerico | Código da tabela de preços e do produto ou número do registro da tabela DA1 que será enviado para cadastro no Telecontrol. Se nenhuma informação for passada, o registro posicionado na tabela DA1 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se o cadastro foi efetuado |

**TTLCPreco():Consulta**

Sintaxe:

Consulta(cCodigo, cCodRet, cRetorno, lMostraMsg, lMostraReg)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cCodigo | caractere | Código da tabela de preços e do produto. Se nenhuma informação for passada, o registro posicionado na tabela DA1 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |
| cRetorno | caractere | Json devolvido pela API |  |
| lMostraMsg | logico | Informa se as mensagens e alertas devem ser exibidos |  |
| lMostraReg | logico | Informa se o registro consultado deve ser apresentado na tela |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se a consulta foi efetuada com sucesso |

Exemplos

User Function ConPreco()

Local oPreco := Telecontrol.Integracao.Preco.TTLCPreco():New()

DA1->(dbseek(xFilial() + ‘TABPRC’ + ‘PECA1’))

If !oPreco:Consulta(,,,,.T.)

Alert(‘Erro ao consultar o preço da peça ’ + DA1->DA1\_CODPRO + ‘ e tabela ’ + DA1->DA1\_CODTAB)

EndIf

Return

|  |
| --- |
| **Classe TTLCProduto (TLC\_ClasseProdutos.tlpp)** |

A classe TTLCProduto efetua a integração dos produtos no Telecontrol.

Hierarquia

* TObject
  + TTLCAPICadastro
    - TTLCProduto

Construtores

* Construtor TTLCProduto():New()

Métodos

* Cadastra
* Consulta
* Altera

Exemplos

User Function CadProduto()

Local oProduto := Telecontrol.Integracao.Produto.TTLCProduto():New()

SB1->(dbseek(xFilial() + ‘PRODUTO1’))

If !oProduto:Cadastra()

Alert(‘Erro ao integrar o produto’ + SB1->SB1\_COD)

EndIf

Return

**TTLCProduto():Cadastra**

Sintaxe:

Cadastra(xRegistro, cCodRet)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| xRegistro | caractere/numerico | Código do produto ou número do registro da tabela SB1 que será enviado para cadastro no Telecontrol. Se nenhuma informação for passada, o registro posicionado na tabela SB1 será utilizado. |  |
| cCodRet | Caractere | Recebe o código de retorno da API |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se o cadastro foi efetuado |

**TTLCProduto():Consulta**

Sintaxe:

Consulta(cCodigo, cCodRet, cRetorno, lMostraMsg, lMostraReg)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cCodigo | caractere | Código do produto. Se nenhuma informação for passada, o registro posicionado na tabela SB1 será utilizado. |  |
| cCodRet | caractere | Recebe o código de retorno da API |  |
| cRetorno | caractere | Json devolvido pela API |  |
| lMostraMsg | logico | Informa se as mensagens e alertas devem ser exibidos |  |
| lMostraReg | logico | Informa se o registro consultado deve ser apresentado na tela |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lOk | logico | Indica se a consulta foi efetuada com sucesso |

Exemplos

User Function ConProduto()

Local oProduto := Telecontrol.Integracao.Produto.TTLCProduto():New()

SB1->(dbseek(xFilial() + ‘PRODUTO1’))

If !oProduto:Consulta(,,,,.T.)

Alert(‘Erro ao consultar o produto ’ + SB1->B1\_COD)

EndIf

Return

|  |
| --- |
| **PARÂMETROS** |

Alguns parâmetros são necessários para que o processo de integração dos registros no Telecontrol seja efetuado corretamente. Esses parâmetros fazem parte das configurações para funcionamento das rotinas e classes utilizadas no processo de integração.

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Padrão |
| TI\_TLCPROD | caractere | Regra para identificar um produto na tabela SB1. Essa regra precisa ser escrita em ADVPL. | SB1->B1\_TIPO = ‘PA’ |
| TI\_TLCPECA | caractere | Regra para identificar uma peça na tabela SB1. Essa regra precisa ser escrita em ADVPL. | SB1->B1\_TIPO = ‘PI’ |
| TI\_LOGTAB | caractere | Alias para a tabela de LOG de integração | SZ0 |
| TI\_FLDSEP | caractere | Separador/delimitador de pastas/diretórios | \ |
| TI\_URITLC | caractere | URI padrão das APIs/endpoints do Telecontrol.  Será utilizado caso o campo “uri” não seja informado no arquivo endpoints.json | http://api2.telecontrol.com.br |
| TI\_FLDCFG | caractere | Pasta onde serão gravados os arquivos de configuração das rotinas de integração Protheus x Telecontrol. | \system\@telecontrol |

|  |
| --- |
| **Tabela de LOG** |

Para que o processo de integração mantenha um LOG das movimentações de integração será necessário criar uma tabela para gravar esse LOG. Após criar a tabela será necessário informar o alias dessa tabela no parâmetro **TI\_LOGTAB.**

**Substitua XXX pelo alias escolhido para a tabela de LOG**.

|  |  |
| --- | --- |
| **PREFIXO** | **XXX (a critério da MDX)** |
| **DESCRIÇÃO** | LOG de integrações Telecontrol |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_FILIAL** |
| **TÍTULO** | Filial |
| **DESCRIÇÃO** | Filial do sistema |
| **TIPO** | C |
| **TAMANHO** | (conforme dicionário) |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_API** |
| **TÍTULO** | API |
| **DESCRIÇÃO** | API |
| **TIPO** | C |
| **TAMANHO** | 30 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | API Telecontrol |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_DESCRI** |
| **TÍTULO** | Descrição |
| **DESCRIÇÃO** | Descrição |
| **TIPO** | C |
| **TAMANHO** | 40 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | Descrição da API |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_URI** |
| **TÍTULO** | URI |
| **DESCRIÇÃO** | URI (webservice) |
| **TIPO** | C |
| **TAMANHO** | 60 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Endereço da API |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_ENDP** |
| **TÍTULO** | endpoint |
| **DESCRIÇÃO** | endpoint (webservice) |
| **TIPO** | C |
| **TAMANHO** | 60 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Endpoint da API Telecontrol |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_VERBO** |
| **TÍTULO** | Método |
| **DESCRIÇÃO** | Método do endpoint |
| **TIPO** | C |
| **TAMANHO** | 10 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Método do endpoint:  - get  - post  - put  - delete |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_DADOS** |
| **TÍTULO** | Dados |
| **DESCRIÇÃO** | Dados enviados |
| **TIPO** | M |
| **TAMANHO** | 10 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Dados enviados ao endpoint |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_DATA** |
| **TÍTULO** | Data Inc |
| **DESCRIÇÃO** | Data da Inclusão |
| **TIPO** | D |
| **TAMANHO** | 8 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | Data da inclusão do LOG |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_HORA** |
| **TÍTULO** | Hora Inc |
| **DESCRIÇÃO** | Hora da Inclusão |
| **TIPO** | C |
| **TAMANHO** | 8 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | Hora da inclusão do LOG |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_DTPROC** |
| **TÍTULO** | Data Proc |
| **DESCRIÇÃO** | Data do Processamento |
| **TIPO** | D |
| **TAMANHO** | 8 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Data do Processamento |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_HRPROC** |
| **TÍTULO** | Hora Proc |
| **DESCRIÇÃO** | Hora do Processamento |
| **TIPO** | C |
| **TAMANHO** | 8 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Hora do Processamento |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_STATUS** |
| **TÍTULO** | Situação |
| **DESCRIÇÃO** | Situação do Processamento |
| **TIPO** | C |
| **TAMANHO** | 1 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **OPÇÕES** | 1=Aguardando integração;2=Erro ao integrar;3=Intregrado com sucesso |
| **HELP** | Situação do processamento:  1=Aguardando integração;  2=Erro ao integrar;  3=Intregrado com sucesso |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_RET** |
| **TÍTULO** | Retorno |
| **DESCRIÇÃO** | Retorno da API |
| **TIPO** | M |
| **TAMANHO** | 10 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Dados recebidos da API |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_CODRET** |
| **TÍTULO** | Cód.Retorno |
| **DESCRIÇÃO** | Código de Retorno |
| **TIPO** | C |
| **TAMANHO** | 3 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | Código de retorno recebido do endpoint |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_TABELA** |
| **TÍTULO** | Tabela |
| **DESCRIÇÃO** | Tabela Utilizada |
| **TIPO** | C |
| **TAMANHO** | 3 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Tabela principal utilizada para montagem dos dados |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_RECNO** |
| **TÍTULO** | Registro |
| **DESCRIÇÃO** | Registro da Tabela |
| **TIPO** | N |
| **TAMANHO** | 15 |
| **DECIMAIS** | 0 |
| **PICTURE** | @E 999,999,999,999,999 |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Registro da tabela principal utilizado para montagem dos dados |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_MSG** |
| **TÍTULO** | Mensagem |
| **DESCRIÇÃO** | Mensagem |
| **TIPO** | C |
| **TAMANHO** | 200 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Mensagem para detalhar a situação do processamento |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_IDRET** |
| **TÍTULO** | ID Retorno |
| **DESCRIÇÃO** | ID Retorno |
| **TIPO** | C |
| **TAMANHO** | 15 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Sim |
| **HELP** | ID recebido da API |

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_CLASSE** |
| **TÍTULO** | Classe |
| **DESCRIÇÃO** | Classe |
| **TIPO** | C |
| **TAMANHO** | 50 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | Não |
| **HELP** | Classe utilizada para integração |

|  |
| --- |
| **Campo ID Telecontrol** |

Para criar o vínculo entre o registro cadastrado no Protheus e o registro cadastrado no Telecontrol, será necessário criar um campo para que o ID Telecontrol seja gravado no Protheus.

Esse campo deverá ser criado em todas as tabelas e será preenchido automaticamente pela rotina de cadastro ou alteração.

|  |  |
| --- | --- |
| **CAMPO** | **XXX\_XIDTLC** |
| **TÍTULO** | ID Telecontr |
| **DESCRIÇÃO** | ID Telecontrol |
| **TIPO** | C |
| **TAMANHO** | 20 |
| **PICTURE** |  |
| **CONTEXTO** | Real |
| **PROPRIEDADE** | Visualizar |
| **BROWSE** | (a critério da MDX) |
| **HELP** | ID do registro no Telecontrol |

**Substitua XXX pelo alias da tabela pertinente**.

As tabelas de cadastro envolvidas no processo de integração são:

|  |  |
| --- | --- |
| Nome | Descrição |
| SA2 | cadastro de fornecedores (postos autorizados) |
| SB1 | cadastro de produtos e peças |
| DA1 | cadastro de preços |

|  |
| --- |
| **Monitor de integração** |

Foi criado um monitor de integração para que seja possível visualizar os dados da integração (informações enviadas e recebidas) bem como a situação da integração.

O monitor apresentará as informações gravadas na tabela de LOG (TI\_LOGTAB), onde será possível identificar, por exemplo, porque uma integração falhou.

Para visualizar os registros de LOG será necessário criar um item de menu em quais - e quantos - módulos forem necessários.

|  |  |
| --- | --- |
| **Menu** | Atualizações |
| **Submenu** | Integr. Telecontrol |
| **Nome da rotina** | Monitor Integração |
| **Programa** | TLC\_Monitor.PRW |
| **Rotina** | TLCMONIT() |
| **Módulo** | (a critério da MDX) |
| **Tipo** | Função de usuário |
| **Tabelas** | XXX |

**Substitua XXX pelo alias escolhido para a tabela de LOG**.

Tela do monitor

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

|  |
| --- |
| **Arquivos** |

Segue listagem dos arquivos envolvidos no processo de integração dos cadastros:

|  |  |
| --- | --- |
| Nome | Descrição |
| TLC\_DEFINE.ch | arquivo com as constantes utilizadas nas rotinas |
| TCL\_XFUN.tlpp | funções diversas |
| TLC\_ClassePecas.tlpp | classe para manutenção de peças no Telecontrol (cadastro e consulta) |
| TLC\_ClassePostos.tlpp | classe para manutenção de postos autorizados (cadastro, alteração e consulta) |
| TLC\_ClassePrecos.tlpp | classe para manutenção de preços (cadastro, alteração e consulta) |
| TLC\_ClasseProdutos.tlpp | classe para manutenção de produtos (cadastro, alteração e consulta) |
| TLC\_Classes.tlpp | classes base para cadastros e manutenção do LOG |
| TLC\_Monitor.prw | tela para apresentar o browse e os detalhes dos registros de LOG |
| TLC\_SHRET.prw | tela para apresentar os dados do registro consultado no Telecontrol |

|  |
| --- |
| **Pontos de entrada** |

Para que os registros sejam incluídos ou alterados em tempo real, é necessário que sejam implementados os pontos de entrada para efetuar as ações necessárias no momento em que o evento de gravação é executado.

Também foram implementadas opções para consulta de registros no Telecontrol e para cadastro manual.

Segue listagem dos pontos de entrada implementados:

|  |  |  |
| --- | --- | --- |
| Nome | Fonte | Descrição |
| MA020ROT | MA020ROT.prw | ponto de entrada para adicionar as opções para consulta e cadastro manual do posto autorizado no Telecontrol a partir do browse de fornecedores. |
| CUSTOMERVENDOR | MATA020\_PE.prw | pontos de entrada para cadastrar ou alterar automaticamente – e em tempo real – o posto autorizado no Telecontrol.  Este ponto de entrada atua no cadastro de fornecedores. |
| MT010ALT | MT010ALT.prw | ponto de entrada para enviar a alteração do produto ou peça para o Telecontrol. |
| MT010INC | MT010INC.prw | ponto de entrada para cadastrar um produto ou peça no Telecontrol. |
| MTA010MNU | MTA010MNU.prw | ponto de entrada para adicionar as opções para consulta ou cadastro manual de produtos no Telecontrol a partir do browse de produtos. |
| OMSA010 | OMSA010\_PE.prw | pontos de entrada para cadastrar ou alterar automaticamente – e em tempo real – o preço das peças no Telecontrol.  Este ponto de entrada atua na tabela de preços. |

|  |  |
| --- | --- |
| Ícone  Descrição gerada automaticamente | Os pontos de entrada enviados são exemplos de uso. Será preciso avaliar se a MDX já implementou os pontos de entrada e efetuar as alterações necessárias para manter o funcionamento atual e implementar a integração com o Telecontrol. |