|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 28/08/2025 |
| ELABORADO POR | EMERSON NASCIMENTO |

**INTEGRAÇÃO DE FATURAMENTO**

**PROTHEUS TELECONTROL**

|  |
| --- |
| **Configurações** |

Para que a integração do faturamento funcione, algumas configurações precisam ser efetuadas.

Essas configurações são feitas em arquivos json e preenchimento de parâmetros.

Os arquivos json indicam como o faturamento será informado ao Telecontrol e como o retorno será processado e gravado no Protheus.

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

Os arquivos de configuração seguem uma nomenclatura padrão:

- faturamento\_cadastra\_post.json – indica como deve ser informado ao Telecontrol que o pedido foi faturado

- faturamento\_cancela\_post.json – indica como deve ser informado ao Telecontrol que uma nota fiscal foi cancelada

- 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 fazem a leitura do pedido e que integram o pedido no Protheus.

Classes desenvolvidas para integração de pedidos:

- 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

- TTLCFaturamento – classe especializada. Serve para informar o Telecontrol sobre o faturamento de um pedido de venda (herdada TTLCAPICadastro)

Rotinas desenvolvidas para integração dos pedidos:

M460FIM() – Envia os dados de faturamento para o Telecontrol.

M521DFNS() – Envia a informação de cancelamento da nota fiscal para o Telecontrol.

|  |
| --- |
| **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'}"

        }

    ],

    "pedidovenda": {

        "importa": {

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

            "description": "Importação de pedido de venda",

            "endpoint":"{|| '/posvenda-pedido/pedidos/dataInicial'+ cConsulta +'/dataFinal/' + cConsulta2 }"

        },

        "put": {

            "description": "Marca o pedido de venda como importado",

            "endpoint": "{|| '/posvenda-pedido/pedidos/pedido/' + 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.

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 faturamento\_cadastra\_post.json** |

O arquivo *faturamento\_cadastra\_post.json* contém a configuração necessária para informar ao Telecontrol sobre o faturamento de um pedido.

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.

A configuração um campo o padrão é:

“campo”: “bloco de código ADVPL”

Exemplo:

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

Segue modelo com o envio do número do pedido de venda:

{

    "pedido": "{|| SC5->C5\_NUM}"

}

Conteúdo inicial do arquivo:

{

"posto": "{|| GetAdvFVal('SA1','A1\_CGC',xFilial('SA1')+SF2->(F2\_CLIENTE+F2\_LOJA),1)}",

"notaFiscal": "{|| SF2->F2\_DOC}",

"serie": "{|| SF2->F2\_SERIE}",

"emissao": "{|| left(FWTimeStamp(3, SF2->F2\_EMISSAO),10)}",

"totalNota": "{|| alltrim(cValToChar(SF2->F2\_VALBRUT))}",

"totalIpi": "{|| alltrim(cValToChar(SF2->F2\_VALIPI))}",

"totalIcms": "{|| alltrim(cValToChar(SF2->F2\_VALICM))}",

"cfop": "{|| ''}",

"transportadora": "{|| SF2->F2\_TRANSP}",

"valorFrete": "{|| alltrim(cValToChar(SF2->F2\_FRETE))}",

"envio": "{||''}",

"codigoRastreio": "{|| ''}",

"itens": {

"pedido": "{|| GetAdvFVal('SC5','C5\_XIDTLC',xFilial('SC5')+SD2->D2\_PEDIDO,1)}",

"pedidoItem": "{|| GetAdvFVal('SC6','C6\_XIDTLC',xFilial('SC6')+SD2->(D2\_PEDIDO+D2\_ITEMPV),1)}",

"peca": "{|| SD2->D2\_COD}",

"quantidade": "{|| alltrim(cValToChar(SD2->D2\_QUANT))}",

"preco": "{|| alltrim(cValToChar(SD2->D2\_PRCVEN))}",

"baseIcms": "{|| alltrim(cValToChar(SD2->D2\_BASEICM))}",

"valorIcms": "{|| alltrim(cValToChar(SD2->D2\_VALICM))}",

"aliquotaIcms": "{|| alltrim(cValToChar(SD2->D2\_PICM))}",

"baseIpi": "{|| alltrim(cValToChar(SD2->D2\_BASEIPI))}",

"valorIpi": "{|| alltrim(cValToChar(SD2->D2\_VALIPI))}",

"aliquotaIpi": "{|| alltrim(cValToChar(SD2->D2\_IPI))}",

"baseSubsTributaria": "{|| alltrim(cValToChar(if(SD2->D2\_ICMSRET>0,SD2->D2\_BASEICM,0)))}",

"valorSubsTributaria": "{|| alltrim(cValToChar(SD2->D2\_ICMSRET))}",

"pecaPedida": "{|| SD2->D2\_COD}"

},

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

}

Caso a informação de faturamento seja recebida corretamente, o código 201 será apresentado pelo Telecontrol.

Também será devolvido o json abaixo:

{

"faturamento": 6091596,

"pedidos": [

"12157278"

]

}

|  |
| --- |
| **Configurações – arquivo faturamento\_cancela\_post.json** |

O arquivo *faturamento\_cancela\_post.json* contém a configuração para informar ao Telecontrol que uma nota fiscal foi cancelada.

A configuração deve ser efetuada da seguinte forma:

“campo”: “bloco de código ADVPL”

Exemplo:

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

Conteúdo inicial do arquivo de configuração:

{

"posto": "{|| GetAdvFVal('SA1','A1\_CGC',xFilial('SA1')+SF2->(F2\_CLIENTE+F2\_LOJA),1)}",

"notaFiscal": "{|| SF2->F2\_DOC}",

"serie": "{|| SF2->F2\_SERIE}"

}

|  |
| --- |
| **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 TTLCFaturamento (TLC\_ClasseFaturamento.tlpp)** |

A classe TTLCFaturamento é responsável por enviar a informação de faturamento por enviar a informação sobre o cancelamento de notas fiscais.

Hierarquia

* TObject
  + TTLCAPICadastro
    - TTLCFaturamento

Construtores

* Construtor TTLCPedidoVenda():New()

Métodos

* Cadastra
* Cancela

Exemplo

User Function TFatPV()

Local oAPIFaturamento := TTLCFaturamento:New()

    oAPIFaturamento:Cadastra('123456789','001')

    FWFreeObj(@oAPIFaturamento)

Return

**TTLCFaturamento():Cadastra**

Sintaxe: Cadastra(cNotaFiscal, cSerie, cCodRet, cRetorno)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cNotaFiscal | caractere | Número da nota fiscal |  |
| cSerie | caractere | Série da nota fiscal |  |
| cCodRet | caractere | Variável que receberá o código de retorno do processamento |  |
| cRetorno | caractere | Texto com o conteúdo retornado pela API Telecontrol |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lRet | lógico | Indica se faturamento foi registrado pelo Telecontrol |

**TTLCFaturamento():Cancela**

Sintaxe: Cancela(cNotaFiscal, cSerie, cCodRet, cRetorno)

Parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Descrição | Obrigatório |
| cNotaFiscal | caractere | Número da nota fiscal |  |
| cSerie | caractere | Série da nota fiscal |  |
| cCodRet | caractere | Variável que receberá o código de retorno do processamento |  |
| cRetorno | caractere | Texto com o conteúdo retornado pela API Telecontrol |  |

Retorno

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrição |
| lRet | lógico | Indica se o aviso de cancelamento foi registrado pelo Telecontrol |

|  |
| --- |
| **Rotina M460FIM (M460FIM.prw)** |

A rotina M460FIM é um ponto de entrada executado após o faturamento. Ele será utilizado para enviar a informação de faturamento para o Telecontrol. A classe TTLCFaturamento será utilizada para comunicação entre os sistema.

Esta rotina é executada automaticamente ao gerar uma nota fiscal de saída.

|  |
| --- |
| **Rotina M521DFNS (M521DFNS.prw)** |

A rotina M521DFNS é um ponto de entrada executado após o cancelamento da nota fiscal. Ele será utilizado para enviar a informação do cancelamento de uma nota fiscal para o Telecontrol. A classe TTLCFaturamento será utilizada para a comunicação entre os sistemas.

Esta rotina deve ser agendada para que os pedidos sejam integrados no intervalo mais conveniente para a MDX.

|  |
| --- |
| **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\_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 |

|  |
| --- |
| **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 integração.

|  |  |
| --- | --- |
| **CAMPO** | **F2\_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 |

As tabelas envolvidas no processo de integração são:

|  |  |
| --- | --- |
| Nome | Descrição |
| SF2 | cabeçalho de notas fiscais de saída |

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

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

|  |  |
| --- | --- |
| Nome | Descrição |
| TLC\_DEFINE.ch | arquivo com as constantes utilizadas nas rotinas |
| TCL\_XFUN.tlpp | funções diversas |
| TLC\_ClasseFaturamento.tlpp | classe para transmitir os dados de faturamento para o Telecontrol |
| TLC\_Classes.tlpp | classes base para cadastros e manutenção do LOG |
| M460FIM.prw | ponto de entrada executado após o faturamento de um pedido de venda |
| M521DNFS.prw | ponto de entrada executado após o cancelamento de uma nota fiscal de saída. |