

ENAPI301 - Consistir e gravar informações referentes a estruturas de itens

 Tempo aproximado para leitura: 6 minutos

Documento de API

Produto:	Datasul
Ocorrência:	Documentação de API
Nome Físico	docapi/enapi301.p

Objetivo: Consistir e gravar informações referentes a estruturas de itens.

Considerações Gerais:

- Devem ser passadas 6 parâmetros para a criação de estruturas.
- A temp-table TT-VERSAO-INTEGR contém a versão de integração da API. Pode ser passada sem registros.
- A temp-table TT-ERROS-GERAL contém os erros ocorridos durante o processo de cadastramento da estrutura.
- A temp-table TT-ESTRUTURA é passado como parâmetro através do comando **INPUT-OUTPUT**, pois além de passar parâmetros, ela retorna informações para o programa chamador.
- A temp-table TT-REF-ESTRUT deve ser alimentada caso se queira trabalhar com itens pais ou componentes que possuam controle de estoque por referência.
- A variável lógica l-obsoleto deve ser passada como "yes" caso seja necessário que um item obsoleto seja incluído como item pai ou mesmo como componente de uma estrutura.
- A temp-table TT-COMPONENTE retorna o número de componentes gerados para cada item pai na estrutura.

A sintaxe para a chamada da API é:

```
run enp/enapi301.p (input      table tt-versao-integr,
                    output     table tt-erros-geral,
                    input-output table tt-estrutura,
                    input      table tt-ref-estrut,
                    input      no,
                    output     table tt-componente).
```

Parâmetros de Entrada

Temp-table tt-componente:

```
def temp-table tt-componente
  field it-codigo    like estrutura.it-codigo
  field contador     as integer format ">9".
```

TEMP-TABLE TT-COMPONENTE			
Atributo	Tipo	Formato	Valor Inicial
It-codigo	Char	X(16)	
Contador	Integer	>9	

Temp-table tt-estrutura : Temp-table que vai possuir as informações referentes ao cadastro das estruturas de itens.

Definição do Include: enp/enapi201.i:

def temp-table tt-estrutura no-undo like estrutura

```
field cod-refer-f      like estrutura.cod-refer
field cod-maq-origem   as integer format "999"      initial 0
field num-processo     as integer format ">>>>>>>9" initial 0
field num-sequencia    as integer format ">>>>>9"   initial 0
field ind-tipo-movto   as integer format "99"       initial 1
field cod-erro         as integer format "99999"    initial ?
field des-erro         as char   format "x(60)"
INDEX ch-codigo IS PRIMARY cod-maq-origem
                        num-processo
                        num-sequencia.
```

TEMP-TABLE tt-estrutura			
Atributo	Tipo	Formato	Valor Inicial
It-codigo	Char	X(16)	
Sequencia	Inte	>>>>9	
Es-codigo	Char	X(16)	
Fantasma	Logi	Sim/Não	Não
Revisão	Char	X(8)	
Serie-inicia	Char	X(12)	
Serie-final	Char	X(12)	
Op-codigo	Inte	>>>>>9	
Local-montag	Char	X(55)	
Fator-perda	Deci-2	>9.99	
Proporção	Deci-2	>>9.99	100
Quant-usada	Deci-10	->>, >>9.999999999	1
Data-inicio	Date	99/99/9999	Hoje
Data-termino	Date	99/99/9999	31/12/9999
Observação	Char	X(40)	
Tempo-reserv	Inte	>>>9	
Var-propor	Deci-2	>>9.99	
Cod-roteiro	Char	X(16)	
Conc-max	Deci-4	>>>9,9999	
Conc-min	Deci-4	>>>9,9999	
Concentração	Deci-4	>>>9,9999	
Rendimento	Deci-4	>>>9,9999	
Tipo-sobra	Inte	>9	1
Cod-refer	Char	X(8)	
Eng-resp	Inte	>9	
Cod-depos	Char	X(3)	
Reap-desmont	Logi	Sim/Não	Sim

Posição	Char	X(16)	
Variação-qt	Inte	9	3
Log-todas-ref-item	Logi	Sim/Não	Sim
Log-ref-estrut-dispon	Logi	Sim/Não	Não
Log-atualiza-via-mmp	Logi	Sim/Não	Não
Quant-liquid	Deci-10	->>, >>9.9999999999	1
Qtd-item	Deci-2	->>>>, >>9.9999	1
Ind-tipo-movto	Inte	99	1
Cod-erro	Inte	99999	
Desc-erro	Char	X(60)	

Obs.: O campo ind-tipo-movto determina a operação a ser realizada pela API:

- 1 – Inclusão
- 2 – Modificação
- 3 – Eliminação

Temp-table tt-ref-estrut: nesta temp-table são armazenadas as informações referentes aos itens controlados por referência fazendo a devida associação de um item com a sua respectiva referência.

Definição do Include: enp/enapi201.i:

```
def temp-table tt-ref-estrut no-undo like ref-estrut

    field cod-maq-origem    as integer format "999"    initial 0

    field num-processo      as integer format ">>>>>>>9" initial 0

    field num-sequencia     as integer format ">>>>>9"  initial 0

    field ind-tipo-movto    as integer format "99"      initial 1

    field cod-erro          as integer format "99999"   initial ?

    field des-erro          as char    format "x(60)"

    INDEX ch-codigo IS PRIMARY cod-maq-origem
                                num-processo
                                num-sequencia.
```

TEMP-TABLE TT-REF-ESTRUT			
Atributo	Tipo	Formato	Valor Inicial
It-codigo	Char	X(16)	
Sequencia	Inte	>>>>>9	
Es-codigo	Char	X(16)	
Cod-ref-it	Char	X(8)	
Cod-ref-es	Char	X(8)	

Execução

Execução: O programa **enapi301.p** irá executar estas validações básicas:

1ª Validação: Versão de Integração

O programa irá verificar se o programa chamador está íntegro com a **API**, e isto ocorre através da verificação da versão de integração passada como parâmetro. Caso a versão esteja incompatível, a **API** abortará a execução retornando o código de erro 3941.

Outras Validações

- Valida o item pai e o componente da estrutura
- Nas inclusões verifica se a estrutura já está cadastrada

Parâmetros de Saída

No retorno da execução do enapi301.p, será retornado um dos valores abaixo (via **RETURN-VALUE**):

NOK: Houveram erros durante o processamento.

OK: Execução com sucesso

 Macro desconhecida: 'rate'

[base_de_conhecimento](#) [enapi301](#) [men](#)



[Política de
privacidade](#)

[Termos
de uso](#)