

CPAPI012 - Requisição de Materiais para Produção

 Tempo aproximado para leitura: 10 minutos

Documento de API

Produto:	Datasul
Ocorrência:	Documentação de API
Nome Físico	cpp/cpapi012.p

Objetivo

Efetuar requisição de materiais para a produção, sendo responsável pela baixa na matéria-prima utilizada para a produção, gerando o movimento de material, atualizando as reservas da ordem e eventuais alocações.

Funcionamento

A API pode ser chamada de duas formas: A padrão (batch) e a persistente.

Na forma padrão, realiza-se a chamada através de um comando RUN normal, passando-se as temp-tables carregadas com as informações necessárias ao processamento de uma ou mais requisições. A sintaxe para a chamada é:

```
run cpp/cpapi012.p (input table tt-requis,  
                    input-output table tt-erro,  
                    input l-deleta-erros).
```

A execução na forma persistente, permite apenas o “carregamento” na memória do código da API, sem a execução de nenhuma requisição, devendo a temp-tables ser passada sem nenhum registro. Uma vez carregada, a API permite a execução de diversos “métodos”, que poderão ser chamados de forma interativa, englobando todo o processo de requisição de materiais. Esses “métodos” são procedures que são acessadas através de um “handle” que é retornado quando o programa é executado de forma persistente. A sintaxe para executar a API em modo persistente é;

```
run cpp/cpapi012.p persistent set h-handle (input      table tt-requis,  
                                           input-output table tt-erro,  
                                           input      l-deleta-erros).
```

A variável denominada h-handle irá receber o “handle” do programa, permitindo o acesso aos “métodos” internos da API.

Parâmetros

Os parâmetros utilizados pela API são:

- tt-requis – Temp-table contendo as informações principais da requisição, como a quantidade correspondente do acabado que será requisitada, a data da transação e o número da ordem de produção sendo requisitada.
- tt-erro – Temp-table que retorna os erros e “avisos” retornados pela API durante o processamento dos movimentos.
- l-deleta-erros – Informa à API que o conteúdo anterior, eventualmente existente na temp-table de erros será eliminado logo após o início do processamento da API. Normalmente utiliza-se YES, sendo parametrizado como NO apenas se a chamada para a API fizer parte de uma transação maior, que eventualmente já tenha gerado registros na temp-table de erros e que devam permanecer após a chamada da API.

Além das temp-tables acima, se a API for executada em modo persistente, são utilizadas outras temp-tables a fim de passar os parâmetros para os “métodos” da API. Essas temp-tables são:

- tt-aloca – Contém as informações relativas à alocação das reservas da ordem de produção.
- tt-reservas – Contém as informações relativas às reservas da ordem, as quais irão gerar a movimentação de requisição para a ordem.

Retorno

A API irá setar o return-value de forma a informar ao programa “chamador” se a execução foi realizada com sucesso ou não. Os valores possíveis para o return-value são: OK (execução correta) e NOK (execução com problemas). Os motivos para a rejeição do movimento de requisição são especificados na temp-table tt-erro.

Principais Campos

Os principais campos, utilizados pela API são:

- TT-REQUIS:

Tipo-trans – Define o tipo de transação que será executada. Pode valer 1 (requisição) ou 2 (devolução).

Nr-ord-produ – Deve conter o número da ordem de produção sendo requisitada.

Quantidade – Contém a quantidade correspondente ao produto acabado que será utilizada para calcular a quantidade de cada requisição, conforme a quantidade original da reserva.

Data – Data da transação da requisição/devolução.

Item-ini, Item-fim, Deposito-ini, Deposito-fim, Op-codigo-ini, Op-codigo-fim – Contém a faixa de seleção utilizada para gerar as requisições.

Carrega-reservas – Informa se a API irá carregar automaticamente uma temp-table interna com as requisições a serem processadas, ou se irá recebê-la através do método pi-recebe-tt-reservas. Esse parâmetro deve ser YES para execução batch (default) e NO para execução persistente.

Procura-saldos – Informa se a API irá realizar a busca dos saldos (FIFO) automaticamente no depósito das reservas (o default para execução batch é YES).

Prog-seg – Recebe nome do programa de origem, ou seja, do programa que está “chamando” a API de requisição.

Execução Persistente

Quando a API for executada em modo persistente, os seguintes métodos estarão disponíveis para execução:

f-adiciona-lista (input c-lista as char

input c-item as char)

Retorna uma lista separada por vírgulas, contendo a lista original passada como parâmetro mais o item informado (c-item).

pi-recebe-tt-requis (input table tt-requis)

pi-recebe-tt-reservas (input table tt-reservas)

pi-recebe-tt-aloca (input table tt-aloca)

Recebem um input parameter table contendo a temp-table correspondente. Tem por objetivo enviar para a API uma temp-table, que vai ser utilizada na execução de outros métodos.

pi-retorna-tt-requis (output table tt-requis)

pi-retorna-tt-reservas (output table tt-reservas)

pi-retorna-tt-aloca (output table tt-aloca)

Retornam (através de um output parameter table) da API a temp-table correspondente.

pi-processa-requis (input table tt-requis,

input-output table tt-erro,

input l-deleta-erros)

Procedure principal da API de requisição, responsável pelo processamento e geração de todos os movimentos de estoque. Recebe os mesmos parâmetros que a API de requisição quando executada em modo batch. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, o movimento não foi realizado.

pi-valida-ordem (input l-imediato,

input i-nr-ord-prod,

input c-origem,

output c-erro,

output c-texto)

Realiza validações relativas à ordem de produção a ser reportada. O parâmetro l-imediato informa à procedure se ela deve retornar imediatamente após detectar o primeiro erro, ou se deve realizar todas as validações antes de retornar. O parâmetro i-nr-ord-prod informa o número da ordem de produção que será validada. O campo c-origem determina a origem do programa chamador da API, podendo ser CP (produção) ou MI (Manutenção Industrial). O retorno da procedure ocorre através dos output parameters c-erro (que contém uma lista dos códigos de erros encontrados) e c-texto (que contém uma lista de textos eventualmente associados a cada erro). Também é utilizado o return-value (“OK” ou “NOK”).

pi-valida-requis (input l-imediato,

input i-nr-ord-prod,

input c-origem,

output c-erro,

output c-texto)

Realiza todas as validações relativas ao movimento de requisição. Os parâmetros e o return-value funcionam da mesma forma que a procedure anterior. Para executar essa procedure, é necessário que a API tenha recebido a temp-table de requisição (pi-recebe-tt-requis). Obs.: Essa procedure executa internamente a pi-valida-ordem (se o parâmetro l-imediato for NO).

pi-valida-reserva (input i-nr-ord-prod,

input i-seq,

input i-tipo-trans,

input l-imediato,

output c-erro,

output c-texto)

Valida um registro da temp-table de reservas, localizado através dos parâmetros i-nr-ord-prod e i-seq, que contém respectivamente o número da ordem e a sequência da reserva da ordem, sendo que esses campos constituem a chave primária e única da temp-table de reservas. O parâmetro i-tipo-trans indica o tipo de transação, podendo conter o valor 1 (requisição) ou 2 (devolução). Os parâmetros l-imediato, c-erro, c-texto e também o return-value funcionam como nas procedures anteriores. É necessário que a temp-table de reservas já esteja carregada na instância da API.

pi-valida-alocacao (input i-nr-ord-prod,

input i-seq,

input l-imediato,

output c-erro,

output c-texto)

Realiza a validação de um registro da temp-table de alocações. Os demais parâmetros e o return-value seguem o funcionamento das procedures anteriores.

pi-finalizar

Finaliza a execução persistente da API, eliminando o código da memória e tornando o handle inválido. Deve ser executada após o término de todos os movimentos, de preferência na saída do programa “chamador”.

pi-verifica-saldo (input c-cod-estabel)

Verifica se as requisições existentes na temp-table tt-reservas possuem ou não saldo disponível para serem atendidas. O campo c-cod-estabel indica o estabelecimento onde será realizada a busca de saldos. Para que essa procedure seja executada, é necessário que a temp-table tt-reservas tenha sido previamente carregada.

pi-carrega-tt-reservas

Cria internamente as temp-tables de reservas (tt-reservas) e de alocações (tt-aloca), podendo as mesmas serem obtidas por meio da execução dos métodos pi-retorna-tt-reservas e pi-retorna-tt-aloca. Para que a execução dessa procedure tenha êxito, a temp-table de requisição (tt-requis) deve ter sido previamente carregada.

[base_de_conhecimento](#) [mcp](#) [cpapi012](#) [datasul](#) [manufatura](#)
[requisicao_de_materiais_para_producao](#) [cpapi012_p](#)



[Política de
privacidade](#)

[Termos
de uso](#)