



Lojas virtuais que vendem

Manual API

Versão 2.0

Sumário

1	<i>Apresentação</i>	1
2	<i>Métodos</i>	2
	2.1 Permissões de acesso aos métodos	4
	2.2 Controle de volume de acessos à API.	4
3	<i>XML de produtos</i>	5
	3.1 Alteração de produtos	6
	3.2 Exclusão de produtos.	8
	3.3 Inclusão de produtos.	10
4	<i>XML de pedidos</i>	13
	4.1 Alteração de pedidos.	13
	4.2 Chaves do XML	14
5	<i>Realização de testes</i>	16
	5.1 Testes do método "ReportView".	17
	5.2 Testes do método "UtilityExecute".	20
	5.3 Testes do método "OrderUpdate".	21
	5.4 Testes do método "ProductManagement".	22
	5.5 Exemplo de VBS	23
6	<i>Dicionário</i>	24
7	<i>Melhores práticas</i>	31

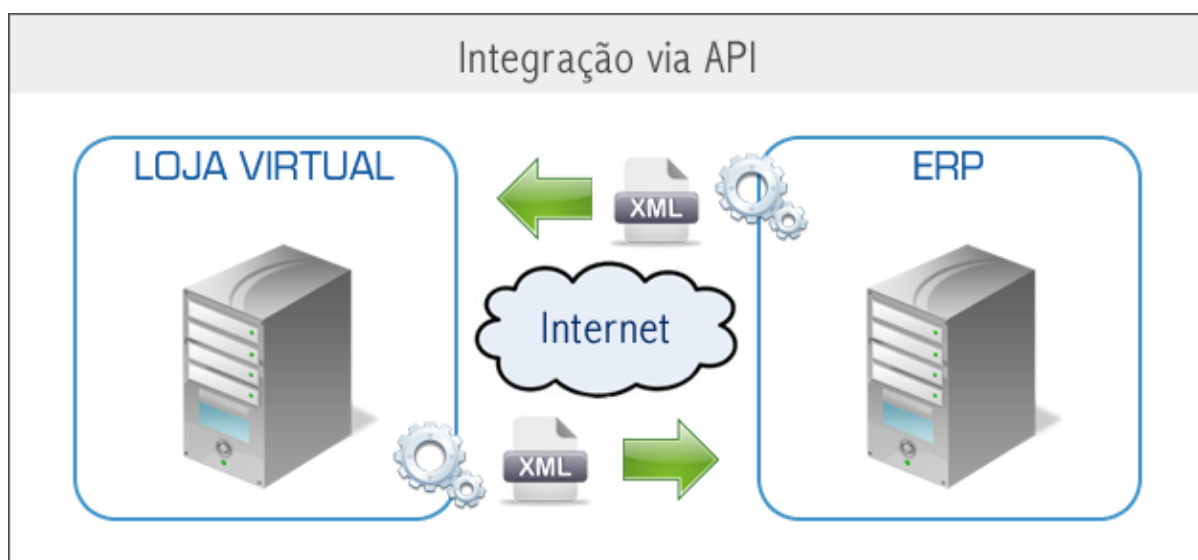
Controle de versão

Versão	Data	Seção	Descrição
2.0	13/11/12	2	Inserção: - Par5, Par6, Par7, Par8 - Page (Página que será retornada) e QtRecords (Quantidade de registros retornados por página) - FromRecord (Registro inicial solicitado do relatório) e ToRecord (Registro final solicitado do relatório)
2.0	13/11/12	5.1	Inserção: Campos retornados pela API na chave <Report>, Retorno fracionado e Seleção de campos retornados.
1.9	08/10/12	6	Inserção: Novos campos em Produto, Clientes, Pedidos, Detalhes e Acessos.
1.8	24/05/12	2.2 e 7	Alteração: Limite de 60 para 300 acessos por hora.
1.8	24/05/12	6	Inserção: Novo campo ChangeFlagProdAPI na tabela de campos de Produto.
1.7	14/03/12	4.2	Inserção: - Nova chave ChangeFlagAPI nos XMLs de exemplo de alteração de pedidos: <Field Name="ChangeFlagAPI" Value="99" />
1.6	11/10/11	6	Alteração: Novas bandeiras no campo cartão em Pedidos.
1.5	19/04/10	-	Criação do documento

1 Apresentação

A API (Application Programming Interface) do FastCommerce permite a sistematização remota de diversas tarefas realizadas regularmente pelos usuários através do site administrativo da loja virtual. Com esta API é possível integrar aplicativos externos tais como ERPs, CRMs, sistemas de BI, gateways de pagamento etc, e através destes aplicativos executar métodos da API para gerenciamento de produtos e pedidos, obter dados de relatórios e executar utilitários do FastCommerce.

O acesso à API ocorre sempre através do protocolo HTTPS (porta 443), com certificação EV SSL. A execução de cada método inicia-se através de uma solicitação enviada via FORM POST e termina em seguida com uma única resposta em XML. Cada execução gera somente um pageview, mesmo que seu resultado contenha ou altere milhares de registros. Da mesma forma, uma chamada que não retorne nem altere registro algum também contabilizará como um pageview.



A resposta à execução de cada método é síncrona e pode ocorrer de forma quase imediata ou pode demorar até alguns minutos, dependendo do método e da quantidade de registros lidos ou alterados.

ATENÇÃO:

Acessos à API não devem estar diretamente relacionados aos acessos a qualquer página da loja virtual, ou seja, cada acesso de cliente da loja não deve corresponder a um acesso à API. Para evitar que isto ocorra, existe uma limitação no volume de acessos por hora aos métodos da API (ver **2.2 Controle de volume de acessos à API**).

2 Métodos

O método informa qual a ação desejada na chamada à API. Cada chamada deve obrigatoriamente conter um dos quatro métodos abaixo:

- ◆ **ReportView** – Exportação de dados dos relatórios do FastCommerce em formato XML
- ◆ **UtilityExecute** – Alterações realizadas pelos utilitários administrativos
- ◆ **OrderUpdate** – Alterações dos status, observações curtas e objetos dos Correios dos pedidos.
- ◆ **ProductManagement** – Gerenciamento de produtos (alteração, exclusão e inclusão)

A comunicação se inicia através de um FORM POST para o seguinte endereço:

<https://www.rumo.com.br/sistema/adm/APILogon.asp>

Seguem abaixo os campos **obrigatórios** deste **POST**:

StoreName	Nome da loja
StoreID	ID da loja
Username	Nome do usuário administrativo
Password	Senha do usuário administrativo
Method	Método que será utilizado

Caso o método seja ReportView ou UtilityExecute, os campos são:

ObjectID	ID do relatório (opcional de ObjectName for informado)
ObjectName	Nome do relatório (opcional de ObjectID for informado)
Par1	Parâmetro 1 do relatório ou utilitário (ex.: data inicial)
Par2	Parâmetro 2 do relatório ou utilitário (ex.: data final)
Par3	Parâmetro 3 do relatório ou utilitário (ex.: nome do cliente)
Par4	Parâmetro 4 do relatório ou utilitário (ex.: e-mail do cliente)
Par5	Parâmetro 5 do relatório ou utilitário
Par6	Parâmetro 6 do relatório ou utilitário
Par7	Parâmetro 7 do relatório ou utilitário
Par8	Parâmetro 8 do relatório ou utilitário
Page	Página que será retornada
QtRecords	Quantidade de registros retornados por página
FromRecord	Registro inicial solicitado do relatório

ToRecord	Registro final solicitado do relatório
Fields	Campos que serão retornados (Só método ReportView)
OutputFormat	Formato do retorno: 0 HTML 1 XML 2 AJAX 3 Excel 4 Word 5 XML API (formato default, se o campo não for informado)

Se passar os 4 parâmetros de paginação, os preferenciais são Page e QtRecords. Neste caso, os dois parâmetros devem ser passados para que sejam considerados.

As opções de formato de retorno são as mesmas oferecidas na funcionalidade de relatórios do site administrativo. Na integração de aplicativos, sugerimos que utilize a opção 5 (XML API), que é o padrão.

Exemplo de form:

```
<form name=Logon action=https://www.rumo.com.br/sistema/adm/APILogon.asp method=POST>
  <input type=text name=StoreName value="Nome da loja">
  <input type=text name=StoreID value="99999">
  <input type=text name=Username value="Admin">
  <input type=password name=Password value="*****">
  <input type=text name=method value="ReportView ou UtilityExecute">
  <input type=text name=ObjectID value="425"> <!-- *** -->
  <input type=text name=ObjectName value="Lista de produtos para alterações via API"> <!-- *** -->
  <input type=text name=Par1 value="Nome da categoria"> <!-- opcional -->
  <input type=text name=Par2 value="Ref/Nome/Descr"> <!-- opcional -->
  <input type=text name=Par3 value="false"> <!-- opcional -->
  <input type=text name=Par4 value="false"> <!-- opcional -->
</form>

<!-- *** --> - Se o ObjectID for informado, não é necessário informar o ObjectName
```

Obs: Para listar o **ObjectID** e os parâmetros dos relatórios e utilitários, utilize o seguinte relatório:

◆ (FC) Lista de relatórios e utilitários

Qualquer relatório ou utilitário pode ser executado, mas existem nove relatórios que foram criados especificamente para utilização com a API:

- ◆ Lista de produtos para alterações via API
- ◆ Lista de pedidos para alterações via API
- ◆ Lista de pedidos e dados de cartões de crédito
- ◆ Lista de clientes para consultas via API
- ◆ Lista de pedidos e detalhes para consultas via API
- ◆ Lista de pedidos alterados para consultas via API
- ◆ Lista de classes de produtos para cadastro via API
- ◆ Categorias e IDs para cadastro de produtos via API
- ◆ Log de acessos por API

XMLRecords	Conteúdo do XML com todos os seus registros
------------	---

Caso o método seja **OrderUpdate** ou **ProductManagement**, o campo obrigatório que contém todos os registros é o seguinte:

Exemplo de form:

```
<form name=Logon action=https://www.rumo.com.br/sistema/adm/APILogon.asp method=POST>
  <input type=text name=StoreName value="Nome da loja">
  <input type=text name=StoreID value="99999">
  <input type=text name=Username value="Admin">
  <input type=password name=Password value="*****">
  <input type=text name=method value="OrderUpdate ou ProductManagement">
  <input type=text name=XMLRecords value="Conteúdo em XML com todos os registros">
</form>
```

2.1 Permissões de acesso aos métodos

Para o acesso a API, é necessário que o usuário tenha permissões de acordo com cada método:

Método	Permissão do usuário
OrderUpdate	Acesso aos utilitários de pedido
ProductManagement	Acesso aos utilitários de produto
ReportView	acesso ao tipo do relatório selecionado
UtilityExecute	acesso ao tipo do utilitário selecionado

Exemplo: Se o [ObjectID](#) escolhido for um relatório estatístico, o usuário precisa ter acesso aos relatórios estatísticos.

2.2 Controle de volume de acessos à API

Cada loja poderá fazer até 300 acessos à API por hora (média de cinco acessos por minuto).

Se este limite for excedido, novas consultas serão bloqueadas até a próxima virada da hora. Por exemplo, se forem feitas mais de 300 consultas entre 13h e 14h, somente após as 14h será possível fazer novos acessos à API, e assim por diante.

Quando o limite for excedido, o código 20 será retornado pela API, com a seguinte mensagem:

Reached limit of access per hour

3 XML de produtos

Estrutura do XML com todos os campos para inclusão, alteração ou exclusão de produtos:

Obs: É possível enviar múltiplos produtos por vez.

```
<Records>
  <Record>
    <Field Name="Comando" Value="I, A ou E" />
    <Field Name="NomeCat" Value="Acessórios" />
    <Field Name="IDCategoria" Value="999999" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="CodProd" Value="ABC0001" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="Peso" Value="999" />
    <Field Name="Descricao" Value="A emoção do povo brasileiro ao ver a camisa amarela entrando..." />
    <Field Name="DescrLonga" Value="Os mais de 190 milhões de corações apaixonados vibram junto..." />
    <Field Name="DescrHTM" Value="PoloBrazil.htm" />
    <Field Name="DescrURL" Value="http://www.cbf.com.br/php/home.php?e=29" />
    <Field Name="MetaKeywordsProd" Value="camiseta polo,brazil,copa do mundo" />
    <Field Name="URLTarget" Value="False" />
    <Field Name="Estoque" Value="999" />
    <Field Name="EstoqueMinimo" Value="99" />
    <Field Name="Disponivel" Value="True" />
    <Field Name="ICMS" Value="99" />
    <Field Name="Custo" Value="99.90" />
    <Field Name="Preco" Value="99.90" />
    <Field Name="PrecoProm" Value="99.90" />
    <Field Name="PrecoB2B" Value="99.90" />
    <Field Name="PrecoB2BProm" Value="99.90" />
    <Field Name="DataPromInicio" Value="dd/mm/aaaa" />
    <Field Name="DataPromFim" Value="dd/mm/aaaa" />
    <Field Name="IDParceiroProd" Value="999999" />
    <Field Name="MaxParcelasProd" Value="99" />
    <Field Name="XMLParcelasProd" Value="99" />
    <Field Name="Lancamento" Value="False" />
    <Field Name="EmDestaque" Value="False" />
    <Field Name="ImagemProd" Value="PoloBrazilP.jpg" />
    <Field Name="ImagemDet" Value="PoloBrazilD.jpg" />
    <Field Name="ImagemAmp" Value="PoloBrazilA.jpg" />
    <Field Name="Adicional1" Value="999999" />
    <Field Name="Adicional2" Value="999999" />
    <Field Name="Adicional3" Value="999999" />
    <Field Name="AdicionalD1" Value="XXXXXXXXXX" />
    <Field Name="AdicionalD2" Value="XXXXXXXXXX" />
    <Field Name="AdicionalD3" Value="XXXXXXXXXX" />
    <Field Name="Cores" Value="99" />
    <Field Name="OrdemProd" Value="999" />
    <Field Name="IDProdutoPai" Value="999999" />
    <Field Name="RamoProd" Value="99" />
    <Field Name="MaisProd" Value="bermuda,camiseta,relógio" />
    <Field Name="DiasReposicao" Value="9" />
    <Field Name="DataVencimento" Value="dd/mm/aaaa" />
    <Field Name="DiasAvisoVencimento" Value="99" />
    <Field Name="Embalavel" Value="True" />
  </Record>
  <Record>
    <Field Name="Comando" Value="I, A ou E" />
    <Field Name="NomeCat" Value="Acessórios" />
    <Field Name="IDCategoria" Value="888888" />
    <Field.....
  </Record>
  <Record>
    <Field Name="Comando" Value="I, A ou E" />
    <Field Name="NomeCat" Value="Acessórios" />
    <Field Name="IDCategoria" Value="777777" />
    <Field.....
  </Record>
</Records>
```

3.1 Alteração de produtos

Estrutura do XML para alteração do cadastro de produtos:

(para alteração o comando deve ser **A** e devem ser enviados somente os campos que forem alterados)

```
<Records>
  <Record>
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record>
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
</Records>
```

Neste exemplo, será realizada somente a alteração do nome do produto.

Obs: Para alterar produtos, os campos **Comando** e **IDProduto** são obrigatórios. É possível alterar múltiplos produtos por vez.

O resultado desta operação será retornado em XML. Segue o modelo:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <ErrCode>0</ErrCode>
  <ErrDescr>OK</ErrDescr>
  <Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" Deleted-Sub="0" ElapsedSeconds="0,000" />
</API>
```

ChangeFlagProdAPI: esta chave é utilizada para marcar os produtos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos.

Pode conter valores de 1 a 255. Este campo será automaticamente zerado sempre que ocorrerem alterações no produto através da ficha do produto no site administrativo ou via CSV.

Produtos novos são criados com ChangeFlagAPI=0.

Obs: Alterações no estoque do produto quando este é comprado ou quando o pedido contendo o produto for cancelado não alteram este campo.

Os seguintes relatórios de produtos podem ser utilizados com o método ReportView para trazer a lista de produtos da loja:

- Lista de produtos para alterações via API
- Lista de estoque de produtos para alterações via API

Nestes relatórios, é possível utilizar o filtro ChangeFlagAPI=0 para trazer somente os produtos novos ou alterados pela ficha do produto no site administrativo ou via CSV. Após a importação, utilize o método ProductManagement para alterar o valor de ChangeFlagProdAPI e evitar novas importações redundantes.

Para alterar o estoque existente de um produto, coloque um sinal de + (mais) ou de - (menos) após a quantidade, no campo Estoque.

Exemplos:

Utilize o seguinte XML para adicionar dois itens ao estoque existente do produto 999999. Se o produto estava com 50 itens no estoque, após executar o XML abaixo passará a ficar com 52 itens:

```
<Records>
  <Record>
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="Estoque" Value="2+" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
</Records>
```

Utilize o seguinte XML para retirar três itens do estoque existente do produto 999999. Se o produto estava com 50 itens no estoque, após executar o XML abaixo passará a ficar com 47 itens:

```
<Records>
  <Record>
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="Estoque" Value="3-" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
</Records>
```

Veja a lista com os possíveis códigos de erros (<ErrCode>) e suas descrições (<ErrDescr>):

0	OK	11	Report/Utility error
1	Store not found	12	Invalid StoreID for this logon
2	Invalid user	13	StoreID not informed
3	No Password	14	Store suspended
4	Logon Error	15	Demonstration period is over
5	Blocked IP address	16	XMLRecords not informed
6	Too many retries. User suspended for 3 minutes	17	Access denied
7	Logon Error. Next invalid logon will suspend user for 3 minutes.	18	Invalid Product(s) in XMLRecords
8	Invalid method	19	Invalid Order(s) in XMLRecords
9	Report/Utility not found or access denied	20	Reached limit of access per hour
10	API Access denied		

Se houver **erros**, o resultado desta operação será retornado em XML da seguinte forma:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <ErrCode>18</ErrCode>
  <ErrDescr>Invalid Product(s) in XMLRecords</ErrDescr>
  <IntCode>111</IntCode>
  <IntDescr>Invalid field:NOMEDOCAMPO Record:2</IntDescr>
  <Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
  ElapsedSeconds="0,000"/>
</API>
```

Sempre será retornado os registros anteriores que obtiveram êxito e por último o registro no qual ocorreu o erro.

Segue a lista com os possíveis códigos de erros (<IntCode>) e suas descrições (<IntDescr>):

111	Campo inválido	121	IDProdutoPai não deve ser informado pois a loja não tem sub-produtos
112	Campo duplicado	122	IDProdutoPai não encontrado
113	Excedeu máximo de registros	123	IDProduto não encontrado
114	Sem nome da categoria nem IDCategoria no registro para inclusão	124	IDProdutoPai não pode ser alterado
115	Comando inválido ou não informado	125	Problema na exclusão
116	Atingiu máximo de produtos do plano	126	IDProduto não encontrado em UPDATE
117	NomeCat do sub-produto não existe	127	IDCategoria não encontrado
118	Tipo de campo inválido, nunca deveria ocorrer	128	Não permite mudar o sub-produto de pai
119	Informado campo IDProduto na inclusão	129	Máximo de sub-produtos no produto atingido
120	IDProduto não informado para alteração e exclusão		

Na tag <Stats> são retornados os seguintes atributos:

Sent	Número de registros
Valid	Número de registros válidos
Included	Número de produtos incluídos
IncludedSub	Número de sub-produtos incluídos
Changed	Número de produtos alterados
ChangedSub	Número de sub-produtos alterados
Deleted	Número de produtos excluídos
DeletedSub	Número de sub-produtos excluídos
ElapsedSeconds	Tempo decorrido da operação

3.2 Exclusão de produtos

Estrutura do XML para exclusão do cadastro de produtos:

(para exclusão o comando deve ser **E** e devem ser enviados somente os campos **Comando** e **IDProduto**)

```
<Records>
  <Record>
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="999999" />
  </Record>
  <Record>
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="888888" />
  </Record>
</Records>
```

Obs: Para realizar a exclusão, os campos **Comando** e **IDProduto** são obrigatórios. É possível excluir múltiplos produtos por vez.

O resultado desta operação será retornado em XML. Segue o modelo:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="999999" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="888888" />
  </Record>
  <ErrCode>0</ErrCode>
  <ErrDescr>OK</ErrDescr>
  <Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
  ElapsedSeconds="0,000"/>
</API>
```

Veja a lista com os possíveis códigos de erros (<ErrCode>) e suas descrições (<ErrDescr>):

0	OK	11	Report/Utility error
1	Store not found	12	Invalid StoreID for this logon
2	Invalid user	13	StoreID not informed
3	No Password	14	Store suspended
4	Logon Error	15	Demonstration period is over
5	Blocked IP address	16	XMLRecords not informed
6	Too many retries. User suspended for 3 minutes	17	Access denied
7	Logon Error. Next invalid logon will suspend user for 3 minutes.	18	Invalid Product(s) in XMLRecords
8	Invalid method	19	Invalid Order(s) in XMLRecords
9	Report/Utility not found or access denied	20	Reached limit of access per hour
10	API Access denied		

Se houver **erros**, o resultado desta operação será retornado em XML da seguinte forma:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="E" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
  </Record>
  <ErrCode>18</ErrCode>
  <ErrDescr>Invalid Product(s) in XMLRecords</ErrDescr>
  <IntCode>123</IntCode>
  <IntDescr>IDProduto not found</IntDescr>
  <Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
  ElapsedSeconds="0,000"/>
</API>
```

Sempre será retornado os registros anteriores que obtiveram êxito e por último o registro no qual ocorreu o erro.

Veja a lista com os possíveis códigos de erros (<IntCode>) e suas descrições (<IntDescr>):

111	Campo inválido	121	IDProdutoPai não deve ser informado pois a loja não tem sub-produtos
112	Campo duplicado	122	IDProdutoPai não encontrado
113	Excedeu máximo de registros	123	IDProduto não encontrado
114	Sem nome da categoria nem IDCategoria no registro para inclusão	124	IDProdutoPai não pode ser alterado

115	Comando inválido ou não informado	125	Problema na exclusão
116	Atingiu máximo de produtos do plano	126	IDProduto não encontrado em UPDATE
117	NomeCat do sub-produto não existe	127	IDCategoria não encontrado
118	Tipo de campo inválido, nunca deveria ocorrer	128	Não permite mudar o sub-produto de pai
119	Informado campo IDProduto na inclusão	129	Máximo de sub-produtos no produto atingido
120	IDProduto não informado para alteração e exclusão		

Na tag **<Stats>** são retornados os seguintes atributos:

Sent	Número de registros
Valid	Número de registros válidos
Included	Número de produtos incluídos
IncludedSub	Número de sub-produtos incluídos
Changed	Número de produtos alterados
ChangedSub	Número de sub-produtos alterados
Deleted	Número de produtos excluídos
DeletedSub	Número de sub-produtos excluídos
ElapsedSeconds	Tempo decorrido da operação

3.3 Inclusão de produtos

Estrutura do XML para inclusão do cadastro de produtos:

*(para inclusão o comando deve ser **I** e deve ser enviado, somente os campos que for utilizar)*

```
<Records>
  <Record>
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0001" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record>
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0002" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
</Records>
```

Obs: Para realizar a inclusão, os campos **Comando** e **NomeCat** são obrigatórios e o campo **IDProduto** **NÃO** deve ser utilizado. É possível incluir múltiplos produtos por vez.

O resultado desta operação será retornado em XML. Segue o modelo:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0001" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record Num="1">
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0002" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
ElapsedSeconds="0,000"/>
</API>
```

Obs: Note que o **IDProduto** gerado automaticamente pelo sistema é retornado neste XML.

Veja a lista com os possíveis códigos de erros (<ErrCode>) e suas descrições (<ErrDescr>):

0	OK	11	Report/Utility error
1	Store not found	12	Invalid StoreID for this logon
2	Invalid user	13	StoreID not informed
3	No Password	14	Store suspended
4	Logon Error	15	Demonstration period is over
5	Blocked IP address	16	XMLRecords not informed
6	Too many retries. User suspended for 3 minutes	17	Access denied
7	Logon Error. Next invalid logon will suspend user for 3 minutes.	18	Invalid Product(s) in XMLRecords
8	Invalid method	19	Invalid Order(s) in XMLRecords
9	Report/Utility not found or access denied	20	Reached limit of access per hour
10	API Access denied		

Se houver **erros**, o resultado desta operação será retornado em XML da seguinte forma:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0001" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="I" />
    <Field Name="NomeCat" Value="Camisetas" />
    <Field Name="CodProd" Value="ABC0002" />
    <Field Name="NomeProd" Value="Polo Seleção Italiana" />
    <Field Name="IDProduto" Value="888888" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
<ErrCode>18</ErrCode>
<ErrDescr>Invalid Product(s) in XMLRecords</ErrDescr>
<IntCode>119</IntCode>
<IntDescr>IDProduto should not be informed</IntDescr>
<Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
ElapsedSeconds="0,000"/>
</API>
```

Sempre será retornado os registros anteriores que obtiveram êxito e por último o registro no qual ocorreu o erro.

Segue a lista com os possíveis códigos de erros (<IntCode>) e suas descrições (<IntDescr>):

111	Campo inválido	121	IDProdutoPai não deve ser informado pois a loja não tem sub-produtos
112	Campo duplicado	122	IDProdutoPai não encontrado
113	Excedeu máximo de registros	123	IDProduto não encontrado
114	Sem nome da categoria nem IDCategoria no registro para inclusão	124	IDProdutoPai não pode ser alterado
115	Comando inválido ou não informado	125	Problema na exclusão
116	Atingiu máximo de produtos do plano	126	IDProduto não encontrado em UPDATE
117	NomeCat do sub-produto não existe	127	IDCategoria não encontrado
118	Tipo de campo inválido, nunca deveria ocorrer	128	Não permite mudar o sub-produto de pai
119	Informado campo IDProduto na inclusão	129	Máximo de sub-produtos no produto atingido
120	IDProduto não informado para alteração e exclusão		

Na tag <Stats> são retornados os seguintes atributos:

Sent	Número de registros
Valid	Número de registros válidos
Included	Número de produtos incluídos
IncludedSub	Número de sub-produtos incluídos
Changed	Número de produtos alterados
ChangedSub	Número de sub-produtos alterados
Deleted	Número de produtos excluídos
DeletedSub	Número de sub-produtos excluídos
ElapsedSeconds	Tempo decorrido da operação

4 XML de pedidos

Estrutura do XML com todos os campos que podem ser alterados na ficha do pedido.

```
<Records>
  <Record>
    <Field Name="NumPedido" Value="999999" />
    <Field Name="ObsCurta" Value="abcdefg" />
    <Field Name="Status" Value="3" />
    <Field Name="ObjSedex" Value="a9b9c9d9e9" />
    <Field Name="ChangeFlagAPI" Value="98" />
  </Record>
  <Record>
    <Field Name="NumPedido" Value="888888" />
    <Field Name="ObsCurta" Value="ijlmopq" />
    <Field Name="Status" Value="1" />
    <Field Name="ObjSedex" Value="a8b8c8d8e8" />
    <Field Name="ChangeFlagAPI" Value="99" />
  </Record>
</Records>
```

Obs: É possível enviar múltiplos pedidos por vez.

4.1 Alteração de pedidos

Estrutura do XML para alteração na ficha dos pedidos: *(para alteração, o campo NumPedido é obrigatório)*

```
<Records>
  <Record>
    <Field Name="NumPedido" Value="999999" />
    <Field Name="ObsCurta" Value="abcdefg" />
    <Field Name="Status" Value="4" />
    <Field Name="ObjSedex" Value="a9b9c9d9e9" />
    <Field Name="ChangeFlagAPI" Value="98" />
  </Record>
  <Record>
    <Field Name="NumPedido" Value="888888" />
    <Field Name="ObsCurta" Value="ijlmopq" />
    <Field Name="Status" Value="7" />
    <Field Name="ObjSedex" Value="a8b8c8d8e8" />
    <Field Name="ChangeFlagAPI" Value="99" />
  </Record>
</Records>
```

Obs: Além do campo **NumPedido** é obrigatório ter pelo menos mais um campo. É possível alterar múltiplos pedidos por vez. O resultado desta operação será retornado em XML. Segue o modelo:

```
<API>
  <Record Num="1">
    <Field Name="NumPedido" Value="999999" />
    <Field Name="ObsCurta" Value="abcdefg" />
    <Field Name="Status" Value="6" />
    <Field Name="ObjSedex" Value="a9b9c9d9e9" />
    <Field Name="ChangeFlagAPI" Value="98" />
  </Record>
  <Record Num="2">
    <Field Name="NumPedido" Value="888888" />
    <Field Name="ObsCurta" Value="ijlmopq" />
    <Field Name="Status" Value="5" />
    <Field Name="ObjSedex" Value="a8b8c8d8e8" />
    <Field Name="ChangeFlagAPI" Value="99" />
  </Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="0" Valid="0" Changed="0" ElapsedSeconds="0,000"/>
</API>
```

4.2 Chaves do XML

NumPedido - único campo obrigatório do XML de pedidos, que informa o número do pedido a alterar.

ObsCurta - campo para informar um pequeno texto de observação do pedido (até 10 caracteres). Em geral, a loja padroniza códigos para este campo.

Exemplo: FP-3-ne - tradução: falta produto - item 3 do pedido - não enviado

Status - status para o qual o pedido será alterado.

O status a enviar no método OrderUpdate é numérico, com os seguintes valores possíveis:

1	Novo	3	Em aprovação	5	Aprovado	7	Remetido
2	Cancelado	4	Pendente	6	Liberado		

ObjSedex - código do objeto dos Correios, para acompanhamento da remessa pela loja e pelo cliente

ChangeFlagAPI - utilizado para marcar os pedidos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos. Podem ser utilizados valores de 1 a 255. Este campo será automaticamente zerado sempre que ocorrerem alterações no pedido através da loja virtual ou do site administrativo, incluindo alterações de status e informações de pagamento retornadas por entidades externas (Cielo, Redecard, PagSeguro etc). Pedidos novos são criados com ChangeFlagAPI=0

Os seguintes relatórios de pedidos podem ser utilizados com o método ReportView para trazer a lista de pedidos da loja:

- ◆ **Lista de pedidos para alterações via API**
- ◆ **Lista de pedidos e detalhes para consultas via API**
- ◆ **Lista de pedidos alterados para consultas via API**

Nestes relatórios, é possível utilizar um filtro `ChangeFlagAPI=0` para trazer somente os pedidos novos. Após a importação, utilize o método `OrderUpdate` para alterar o valor de `ChangeFlagAPI` e evitar novas importações redundantes.

Veja a lista com os possíveis códigos de erros (`<ErrCode>`) e suas descrições (`<ErrDescr>`):

0	OK	11	Report/Utility error
1	Store not found	12	Invalid StoreID for this logon
2	Invalid user	13	StoreID not informed
3	No Password	14	Store suspended
4	Logon Error	15	Demonstration period is over
5	Blocked IP address	16	XMLRecords not informed
6	Too many retries. User suspended for 3 minutes	17	Access denied
7	Logon Error. Next invalid logon will suspend user for 3 minutes.	18	Invalid Product(s) in XMLRecords
8	Invalid method	19	Invalid Order(s) in XMLRecords
9	Report/Utility not found or access denied	20	Reached limit of access per hour
10	API Access denied		

Se houver **erros**, o resultado desta operação será retornado em XML da seguinte forma:

```
<API>
  <Record Num="1">
    <Field Name="NumPedido" Value="9999995" />
    <Field Name="ObsCurta" Value="abcdefg" />
    <Field Name="Status" Value="9" />
    <Field Name="ObjSedex" Value="a9b9c9d9e9" />
  </Record>
  <Record Num="2">
    <Field Name="NumPedido" Value="888888" />
    <Field Name="ObsCurta" Value="ijlmopq" />
    <Field Name="Status" Value="9" />
    <Field Name="ObjSedex" Value="a8b8c8d8e8" />
  </Record>
<ErrCode>19</ErrCode>
<ErrDescr>Invalid Order(s) in XMLRecords</ErrDescr>
<IntCode>53</IntCode>
<IntDescr>NumPedido not found</IntDescr>
<Stats Sent="0" Valid="0" Changed="0" ElapsedSeconds="0,000"/>
</API>
```

Sempre será retornado os registros anteriores que obtiveram êxito e por último o registro no qual ocorreu o erro.

Veja a lista com os possíveis códigos de erros (`<IntCode>`) e suas descrições (`<IntDescr>`):

51	Excedeu máximo de registros	55	Campo inválido
52	NumPedido não informado	56	Campo duplicado
53	NumPedido não encontrado	57	Sem campos para alterar
54	Status inválido	58	IDPedido não encontrado em UPDATE

Na tag `<Stats>` são retornados os seguintes atributos:

Sent	Número de registros
Valid	Número de registros válidos
Changed	Número de produtos alterados
ElapsedSeconds	Tempo decorrido da operação

5 Realização de testes

Disponibilizamos uma página de testes para facilitar a realização de testes de integração via navegador e assim conhecer melhor o funcionamento da API. Segue o endereço:

<https://www.rumo.com.br/sistema/adm/apitest.asp>

FAST commerce

Este é o site administrativo da loja.
O acesso é permitido somente aos usuários cadastrados.

Teclado virtual

~	1	2	3	4	5	6	7	8	9	0	=	+
q	w	e	r	t	y	u	i	o	p	.	/	→
a	s	d	f	g	h	j	k	l	;	'	↵	↵
~	z	x	c	v	b	n	m	<	,	>	.	/
Caps Lock	↑ Shift	Space										

Para maior segurança, informe sua senha clicando no teclado acima. Para digitar maiúsculas ou caracteres da parte superior das teclas, clique em **Caps Lock** ou **Shift**.

Login do usuário

StoreName:

StoreID:

Username:

Password:

Dados do formulário para API

Method:

ObjectID:

ObjectName:

Par1:

Par2:

Par3:

Par4:

OutputFormat:

Entrar

FastCommerce 5.2 © Copyright 1997-2010 Rumo

Informe os campos obrigatórios **StoreName**, **StoreID**, **Username** e **Password** no quadro **Login do usuário**. Em seguida, selecione o método desejado. Preencha os campos que serão exibidos de acordo com o método escolhido.

5.1 Testes do método "ReportView"

Neste teste, será possível ver todo o cadastro de produtos da loja no formato da API.

No campo **Method** selecione a opção **ReportView**. No campo **ObjectID** informe **425** (**ObjectID=425** é referente ao relatório **Lista de produtos para alterações via API**) e clique em **Entrar**.

The screenshot shows the FastCommerce administrative interface. At the top, there's a logo and a message: "Este é o site administrativo da loja. O acesso é permitido somente aos usuários cadastrados." Below this is a "Teclado virtual" (Virtual Keyboard) section. To the right is a "Login do usuário" (User Login) section with fields for StoreName (WMusic), StoreID (252), Username (Admin), and Password (masked with dots). Below the login section is a "Dados do formulário para API" (API Form Data) section. This section contains a "Method" dropdown menu set to "ReportView", an "ObjectID" field containing "731", an "ObjectName" field, four "Par" (Parameter) fields (Par1, Par2, Par3, Par4), and an "OutputFormat" dropdown set to "API". At the bottom of the form is an "Entrar" (Enter) button. The footer of the page reads "FastCommerce 5.2 © Copyright 1997-2010 Rumo".

Campos retornados pela API na chave <Report>:

ObjectName	Nome do relatório
StoreName	Nome da loja
ExecDate	Data de execução do relatório
FromRecord	Registro inicial retornado
ToRecord	Registro final retornado
ParName1, ParName2, ParName3, ParName4, ParName5, ParName6, ParName7, ParName8,	Nomes dos parâmetros
ParValue1, ParValue2, ParValue3, ParValue4, ParValue5, ParValue6, ParValue7, ParValue8	Valores dos parâmetros
Records	Registros retornados
ReadRecords	Registros lidos

Como resultado, teremos a seguinte estrutura:

```
<Report ObjectName="Lista de produtos para alterações via API" StoreName="Perfumes.com.br"
ExecDate="05/10/2012 19:25:49" FromRecord="1" ToRecord="500" ParName1="Categoria" ParValue1=""
ParName2="Ref/Nome/Descr" ParValue2="" ParName3="Disponível" ParValue3="" ParName4="Promoção" ParValue4=""
ParName5="De" ParValue5="" ParName6="Até" ParValue6="" ParName7="API Flag" ParValue7="" Records="500"
ReadRecords="3909" >
  <Record>
    <Field Name="Comando" Value="A" /> [
    <Field Name="NomeCat" Value="Acessórios" />
    <Field Name="IDCategoria" Value="999999" />
    <Field Name="IDProduto" Value="999999" />
    <Field ..... />
    <Field ..... />
    <Field ..... />
  </Record>
  <Record>
    <Field Name="Comando" Value="A" />
    <Field Name="NomeCat" Value="Jóias" />
    <Field Name="IDCategoria" Value="888888" />
    <Field Name="IDProduto" Value="777777" />
    <Field ..... />
    <Field ..... />
    <Field ..... />
  </Record>
</Report>
```

Retorno fracionado

O XML de retorno da API do FastCommerce tem limite 4Mb. Quando este limite for atingido, o código 11 será retornado pela API, com a seguinte mensagem:

Results page is bigger than 4Mb. Please try again, changing one or more parameters to restrict results.

Para restringir o resultado de relatórios que trazem muitas informações (exemplo: Lista de produtos), deve-se utilizar os parâmetros disponíveis para o relatório. Se mesmo utilizando os parâmetros o resultado ainda for extenso, existem parâmetros adicionais que permitem fracionar o retorno:

- Page (Página)
 - QtRecords (Quantidade de registros por página)
- ou
- FromRecord (Registro inicial solicitado do relatório)
 - ToRecord (Registro final solicitado do relatório)

Se passar os 4 parâmetros os preferenciais são Page e QtRecords.

Na chave Report do retorno do XML, é informada a quantidade de registros lidos "ReadRecords".

Por exemplo, se for solicitado o relatório "Lista de produtos para alterações via API" passando Page=1 e QtRecords=500, a chave Report informará a quantidade total de registros em ReadRecords, como no exemplo abaixo:

```
<Report ObjectName="Lista de produtos para alterações via API" StoreName="Perfumes.com.br"
ExecDate="05/10/2012 19:25:49" FromRecord="1" ToRecord="500" ParName1="Categoria" ParValue1=""
ParName2="Ref/Nome/Descr" ParValue2="" ParName3="Disponível" ParValue3="" ParName4="Promoção" ParValue4=""
ParName5="De" ParValue5="" ParName6="Até" ParValue6="" ParName7="API Flag" ParValue7="" Records="500"
ReadRecords="3909" >
```

Neste caso, a loja tem 3909 registros de produtos e foram retornados do registro 1 ao 500.

Passando Page=2 e QtRecords=500, serão retornados do registro 501 até o registro 1000, e assim por diante. Ou seja, os parâmetros Page=2 e QtRecords=500 equivalem a FromRecord=501 e ToRecord=1000.

Desta forma é possível programar um loop para ler os registros de forma fracionada, evitando assim atingir o limite de 4Mb do XML de retorno.

Seleção de campos retornados

Cada relatório retorna um conjunto específico de campos. Entretanto, nem sempre todos estes campos são utilizados pela sua aplicação.

Através do parâmetro Fields, é possível selecionar quais campos serão retornados pela API, para que o relatório não retorne campos que não serão utilizados pela sua aplicação. Isto otimizará a geração, o tráfego e o processamento dos resultados.

Por exemplo, o relatório "Lista de produtos para alterações via API" possui dezenas de campos:

Comando, NomeCat, IDCategoria, IDProduto, CodProd, CodBarrasProd, NomeProd, Estoque, ..., ProfundidadeProd, DataProdAlteracao, ChangeFlagProdAPI

Se a sua aplicação deseja receber apenas os campos IDProduto e Estoque, basta informar no parâmetro Fields os campos entre vírgulas: IDProduto,Estoque

O volume menor de dados otimizará o fluxo de dados entre as redes e o tempo de processamento.

5.2 Testes do método "UtilityExecute"

Neste teste, será possível ver todo o cadastro de produtos da loja no formato da API.

No campo **Method** selecione a opção **UtilityExecute**, no campo **ObjectID** informe **248**. No campo **Par4** informe **15** que será seu novo estoque de todos os produtos (**ObjectID=248** é referente ao utilitário **Alteração no estoque de produtos**) e clique em **Entrar**.

FAST commerce

Este é o site administrativo da loja.
O acesso é permitido somente aos usuários cadastrados.

Teclado virtual

Para maior segurança, informe sua senha clicando no teclado acima. Para digitar maiúsculas ou caracteres da parte superior das teclas, clique em **Caps Lock** ou **Shift**.

Login do usuário

StoreName: WMusic
StoreID: 252
Username: Admin
Password: *****

Dados do formulário para API

Method: UtilityExecute
ObjectID: 445
ObjectName:
Par1:
Par2:
Par3:
Par4: 15
OutputFormat: API

Entrar

FastCommerce 5.2 © Copyright 1997-2010 Rumo

Como resultado, teremos a seguinte estrutura, informando que 10 registros foram afetados:

```
<API>
  <ErrCod>0<ErrCod>
  <ErrDescr>OK<ErrDescr>
  <Regs>10<Regs>
</API>
```

5.3 Testes do método "OrderUpdate"

Neste teste, será possível ver todo o cadastro de produtos da loja no formato da API.

No campo **Method** selecione a opção **OrderUpdate**. No campo **XMLRecords** informe o XML com os registros que deseja alterar e clique em **Entrar**.

FAST commerce

Este é o site administrativo da loja.
O acesso é permitido somente aos usuários cadastrados.

Teclado virtual

Para maior segurança, informe sua senha clicando no teclado acima. Para digitar maiúsculas ou caracteres da parte superior das teclas, clique em Caps Lock ou Shift.

Login do usuário

StoreName: WMusic

StoreID: 252

Username: Admin

Password: *****

Dados do formulário para API

Method: OrderUpdate

XMLRecords:

```
<Records>
<Record>
<Field Name="NumPedido" Value="32659" />
<Field Name="ObsCurta" Value="fraude" />
<Field Name="Status" Value="5" />
<Field Name="ObjSedex" Value="a1b2c3d4" />
</Record>
<Record>
<Field Name="NumPedido" Value="32660" />
<Field Name="ObsCurta" Value="60000aaah1" />
<Field Name="Status" Value="5" />
<Field Name="ObjSedex" Value="OK" />
</Record>
</Records>
```

Entrar

FastCommerce 5.2 © Copyright 1997-2010 Rumo

Como resultado, teremos a seguinte estrutura:

```
<API>
<Record Num="1">
  <Field Name="NumPedido" Value="999999" />
  <Field Name="ObsCurta" Value="abcdefg" />
  <Field Name="Status" Value="4" />
  <Field Name="ObjSedex" Value="a9b9c9d9e9" />
</Record>
<Record Num="2">
  <Field Name="NumPedido" Value="888888" />
  <Field Name="ObsCurta" Value="poiuytr" />
  <Field Name="Status" Value="5" />
  <Field Name="ObjSedex" Value="a2b2c2d2e2" />
</Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="0" Valid="0" Changed="0" ElapsedSeconds="0,000"/>
</API>
```

5.4 Testes do método "ProductManagement"

Neste teste, será possível ver todo o cadastro de produtos da loja no formato da API.

No campo **Method** selecione a opção **ProductManagement**. No campo **XMLRecords** informe o XML com os registros que deseja alterar e clique em **Entrar**.

FAST commerce

Este é o site administrativo da loja.
O acesso é permitido somente aos usuários cadastrados.

Teclado virtual

Para maior segurança, informe sua senha clicando no teclado acima. Para digitar maiúsculas ou caracteres da parte superior das teclas, clique em **Caps Lock** ou **Shift**.

Login do usuário

StoreName: WMusic

StoreID: 252

Username: Admin

Password: *****

Dados do formulário para API

Method: ProductManagement

XMLRecords:

```
<Records>
<Record>
  <Field Name="Comando" Value="A" />
  <Field Name="NomeCat" Value="Camisetas Azul" />
  <Field Name="IDCategoria" Value="2715" />
  <Field Name="IDProduto" Value="34381" />
</Record>
<Record>
  <Field Name="Comando" Value="A" />
  <Field Name="NomeCat" Value="Simpsons" />
  <Field Name="IDCategoria" Value="2692" />
  <Field Name="IDProduto" Value="34374" />
</Record>
</Records>
```

Entrar

FastCommerce 5.2 © Copyright 1997-2010 Rumo

Como resultado, teremos a seguinte estrutura:

```
<API>
  <Record Num="1">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="999999" />
    <Field Name="NomeProd" Value="Polo Seleção Brazil" />
  </Record>
  <Record Num="2">
    <Field Name="Comando" Value="A" />
    <Field Name="IDProduto" Value="333333" />
    <Field Name="NomeProd" Value="Regata Seleção Brazil" />
  </Record>
  <ErrCode>0</ErrCode>
  <ErrDescr>OK</ErrDescr>
  <Stats Sent="0" Valid="0" Included="0" IncludedSub="0" Changed="0" ChangedSub="0" Deleted="0" DeletedSub="0"
  ElapsedSeconds="0,000"/>
</API>
```

5.5 Exemplo de VBS

Segue exemplo de integração em VBScript:

```
Option Explicit
```

```
Dim oHTTP, sHTTP, sParam
```

```
'Exemplo de relatório
sParam=""
BuildParam "StoreName","SP Departamentos"
BuildParam "StoreID",184
BuildParam "Username","Paulo"
BuildParam "Password","123456"
BuildParam "method","ReportView"
BuildParam "ObjectID",424 'ObjectID do relatório "Lista de pedidos para alterações via API"
BuildParam "Par1",4
BuildParam "OutputFormat",1
ExecutaAPI

'Exemplo de alteração de um pedido
sParam=""
BuildParam "StoreName","SP Departamentos"
BuildParam "StoreID",184
BuildParam "Username","Paulo"
BuildParam "Password","123456"
BuildParam "method","OrderUpdate"
BuildParam "XMLRecords","<Records><Record><Field Name=""NumPedido"" Value=""25013""/><Field
Name=""ObsCurta"" Value=""Cheque OK""/><Field Name=""Status"" Value=""6""/><Field Name=""ObjSedex""
Value=""SDX0002999""/></Record></Records>"
ExecutaAPI
```

```
Sub BuildParam(sName,sValue)
sParam=sParam & sName & "=" & URLEncode(sValue) & "&"
End Sub
```

```
Function URLEncode(sText)
Dim i,char,sOut,AscChar
For i=1 To Len(sText)
char=Mid(sText,i,1)
AscChar=Asc(char)
If AscChar=32 Then 'Converte espaço para +
sOut=sOut & "+"
ElseIf (AscChar<48 OR AscChar>122) OR (AscChar>57 AND AscChar<65) OR (AscChar>90 AND AscChar<97) Then
sOut=sOut & "%" & FormatZeros(Hex(AscChar),2)
Else
sOut=sOut & char
End If
Next
URLEncode=sOut
End Function
```

```
Function FormatZeros(Num,nz)
'Retorna string formatada com nz zeros à frente
Dim sNum,iTrunc,LenNum
sNum=Trim(Num)
LenNum=Len(sNum)
If LenNum>Int(nz) Then
iTrunc=LenNum
Else
iTrunc=nz
End If
FormatZeros=Right(String(nz,"0") & sNum,iTrunc)
End Function
```

```
Sub ExecutaAPI
Set oHTTP=CreateObject("WinHttp.WinHttpRequest.5.1")
oHTTP.Open "POST","https://www.rumo.com.br/sistema/adm/APILogon.asp",False
oHTTP.Option(0)="FastCommerce API Interface" 'Alterar o UserAgent, para evitar filtragem do FC
oHTTP.SetRequestHeader "Content-type","application/x-www-form-urlencoded"
oHTTP.Send sParam
WScript.Echo oHTTP.ResponseText
Set oHTTP=Nothing
End Sub
```

6

Dicionário

Seguem os tipos, campos e descrições da ficha do **Produto**:

Coluna/Campo	Tipo/Opções	Descrição
Comando	I=Inclusão A=Alteração E=Exclusão	Indica a ação a ser realizada
NomeCat	varchar(25)	Nome da categoria
IDProduto	int	ID interno do banco de dados
CodProd	varchar(15)	Código de referência do produto, exibido ao visitante
NomeProd	varchar(100)	Nome do produto
Peso	real	Peso do produto. A unidade (gramas ou quilogramas) é definida na página Dados da loja
Descricao	varchar(200)	Descrição curta do produto
DescrLongo	varchar(1024)	Descrição longa do produto
DescrURL	varchar(100)	URL para mais detalhes do produto
URLTarget	bit	Se TRUE, URL do campo DescrURL Será exibida em uma nova janela
Estoque	smallint	Quantidade de itens do produto em estoque
EstoqueMinimo	smallint	Quantidade mínima de itens do produto em estoque. Quando estoque ficar menor, será enviado e-mail para o lojista. Se 0, não tem mínimo.
Disponivel	bit	Se TRUE, produto está disponível no site
ICMS	real	Percentual de ICMS na origem do Produto, para geração de nota fiscal
Custo	money	Custo do produto, nunca exibido ao visitante, utilizado para cálculo nos relatórios de lucratividade.
Preco	money	Preço de venda varejo (B2C)
PrecoProm	money	Preço promocional no varejo (B2C)
PrecoB2B	money	Preço de venda atacado (B2B)
PrecoB2BProm	money	Preço promocional no atacado (B2B)
DataPromInicio	smalldatetime	Data de início da promoção, pode-se informar a hora. Ex: 10/05/2002 10:30
DataPromFim	smalldatetime	Data de término da promoção, pode-se informar a hora. Ex: 10/05/2002 10:30
Lancamento	bit	Se TRUE, indica que o produto é lançamento
EmDestaque	bit	Se TRUE, indica que o produto tem destaque no layout da loja. Ver tag <prod>
ImagemProd	varchar(200)	Nome do arquivo com a imagem principal do produto com extensão JPG, PNG, GIF ou SWF
ImagemDet	varchar(200)	Nome do arquivo com a imagem de detalhe do produto com extensão JPG, PNG, GIF ou SWF
ImagemAmp	varchar(200)	Nome do arquivo com a imagem ampliada do produto com extensão JPG, PNG, GIF ou SWF
Adicional1	varchar(1024)	ID para informação adicional sobre o produto, indicado em Descritor especial 1. Se multivalorado colocar entre vírgulas.
Adicional2	varchar(1024)	ID para informação adicional sobre o produto, indicado em Descritor especial 2. Se multivalorado colocar entre vírgulas.
Adicional3	varchar(1024)	ID para informação adicional sobre o produto, indicado em Descritor especial 3. Se multivalorado colocar entre vírgulas.
AdicionalD1	varchar(2048)	Texto da informação adicional sobre o produto, indicado em Descritor simples 1. Se multivalorado, colocar entre vírgulas.
AdicionalD2	varchar(2048)	Texto da informação adicional sobre o produto, indicado em Descritor simples 2. Se multivalorado, colocar entre vírgulas.
AdicionalD3	varchar(2048)	Texto da informação adicional sobre o produto, indicado em Descritor simples 3. Se multivalorado, colocar entre vírgulas.
Cores	varchar(1024)	ID das cores disponíveis no produto. Se multivalorado, colocar entre vírgulas.

OrdemProd	tinyint	Indica a ordem de exibição do produto. Ordenação decrescente.
IDProdutoPai	int	Se preenchido, indica que este é um sub-produto e qual o ID do produto Pai
RamoProd	tinyint	Indica o departamento no Shopping Virtuose.
MaisProd	varchar(50)	Indica termos de busca para produtos relacionados a estes entre vírgula. (cross-selling)
DataVencimento	smalldatetime	Indica a data de vencimento do produto, Quando será enviado e-mail de alerta ao lojista
DiasAvisoVencimento	smallint	Números de dias antes do vencimento do produto, quando será enviado o primeiro e-mail de alerta para o lojista
DiasReposicao	smallint	Número de dias, em média em que o produto é novamente adquirido pelo cliente. Decorridos estes dias após o pedido, um e-mail será enviado ao cliente para sugerir a reposição.
IDParceiroProd	int	Informe o ID do parceiro que terá exclusividade para o preço promocional do produto. Desta forma, é possível criar promoções válidas somente para visitantes cujo acesso seja proveniente de um parceiro ou revendedor específico.
MaxParcelasProd	tinyint	Número máximo de parcelas para pedidos que incluam este produto.
IDCategoria	int	ID interno da categoria cadastrada. No CSV de produtos, ao incluir um produto, é obrigatório o campo NomeCat (com o nome da categoria do produto), o sistema inclui o produto e cria a categoria caso ela não exista. Na API, ao incluir produtos na loja, o campo NomeCat ou IDCategoria deve ser informado. Se a categoria for existente pode ser informado o IDCategoria, sem passar o NomeCat.
XMLParcelasProd	tinyint	Indica o número de parcelas utilizadas para exibir o parcelamento na página que lista os produtos em XML (XMLProdutos.asp). A lista de produtos em XML é utilizado pelos portais de comparação de preços e shopping virtuais, para capturar os produtos da loja.
Embalavel	bit	Se TRUE, indica o cliente pode solicitar que este produto seja embalado para presente. Deve-se informar FALSE nos produtos que não poderão ser embalados para presente devido ao seu tamanho (exemplo: bicicleta) ou por alguma outra restrição. Este campo é utilizado se a loja utilizar as opções (4), (5) ou (6) no campo "Presente" da página "Envio & frete" do site administrativo onde o cliente tem a opção de informar quais produtos do pedido deseja embalar para presente.
DescrHTM	varchar(25)	Indica o nome do arquivo HTML contendo descrição adicional do produto. O conteúdo do arquivo HTM será incluído na posição da tag especial <DescrHTM>, caso esta tag exista no arquivo personalizado de exibição do layout de produtos na loja (EstiloProduto.htm)
MetaKeywordsProd	varchar(200)	Palavras-chave (keywords) específicas para o produto. Os termos devem ser separados por vírgula. Exemplo: nokia, celular com wi-fi, mp3, rede 3G, GPSSão inseridas em "meta tag keyword", no código fonte da loja.
CodBarrasProd	varchar(25)	Código de barras do produto
ChangeFlagProdAPI	tinyint	Utilizado para marcar os produtos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos. Veja mais detalhes na seção 3.1.
IsProdutoGrande	bit	Indica se o produto é considerado grande para cálculo de frete, usando peso cúbico e tabela para frete grande.
DataProdAlteracao	smalldatetime	Data/hora da última alteração do produto (exceto o cancelamento e descancelamento de pedidos)
IDRamo	int	Indica o ID do ramo
NomeRamo	varchar(80)	Nome do Ramo. Ex: Beleza & Saúde, Bermuda Infantil, Bonecos e Personagens, etc.
IDRamoPai	int	Indica o ID do ramo Pai, nulo se for o primeiro nível
IDCategoria	int	ID interno do banco de dados
Categoria	varchar(25)	varchar(25)
Ordem	tinyint	Indica a ordem de exibição. Ordenação decrescente.
IDCategoriaPai	int	Se preenchido, indica que esta é uma sub-categoria e qual o ID da categoria Pai.
Prods	real	Quantidade de produtos

Seguem os tipos, campos e descrições da ficha do **Clientes**:

Coluna/Campo	Tipo/Opções	Descrição
Cadastro em	smalldatetime	Indica a data de cadastramento do cliente
Tipo	Opções: PF ou PJ	Indica se o cliente é pessoa física ou jurídica
Empresa	varchar(50)	Nome da empresa, somente quando B2B
Contato	varchar(50)	Nome
E-mail	varchar(50)	E-mail
CNPJ/CPF	varchar(14)	CPF ou CNPJ

IE/RG	varchar(20)	Inscrição estadual ou RG
Ramo	varchar(30)	Profissão do cliente ou ramo de atuação da empresa
% Desc	real	Percentual de desconto global nos produtos para o cliente.
Endereço	varchar(70)	Endereço
Bairro	varchar(25)	Bairro
Cidade	varchar(20)	Cidade
Estado	char(2)	Estado
País	varchar(20)	País
CEP	varchar(9)	CEP
Telefone	varchar(20)	Telefone
FAX	varchar(20)	FAX
Obs curta	varchar(10)	Texto de apoio sobre o cliente, para uso interno da loja.
Data Nasc	smalldatetime	Data de nascimento
Pagtos	varchar(80)	Indica as formas de pagamento que este cliente pode utilizar, quando a loja tem formas de pagamento por cliente.
Sexo	Opções: PF ou PJ	Sexo do cliente
Última alteração	smalldatetime	Data/hora da última alteração do cliente
Logradouro	varchar(70)	Endereço do cliente
Endereço número	varchar(70)	Número do endereço
Endereço complemento	varchar(70)	Complemento do endereço
Celular	varchar(20)	Telefone móvel completo para contato e envio SMS (com DDD)
Campo 1	varchar(100)	Campo adicional 1
Campo 2	varchar(100)	Campo adicional 2
Campo 3	varchar(100)	Campo adicional 3

Seguem os tipos, campos e descrições da ficha do **Pedidos**:

Coluna/Campo	Tipo/Opções	Descrição
Nome Cliente	varchar(50)	Nome do cliente
E-mail Cliente	varchar(50)	E-mail do cliente
Telefone	varchar(20)	Telefone do cliente
Cidade	varchar(20)	Cidade do cliente
Estado	char(2)	Estado do cliente
Núm Pedido	int	Número do pedido
Produto(s) do Pedido	varchar(100)	Nome do produto
Ref.	varchar(15)	Referência do produto
Qtd	smallint	Quantidade de itens do produto
Total sem frete	money	Valor total do pedido, SEM o frete
Frete	money	Valor do frete
Total com frete	money	Valor total do pedido, COM o frete
Pagamento	varchar(50)	Opção de pagamento escolhida pelo cliente
Cartão	Opções: AMEX, Diners, Mastercard, VISA, Hipercard, Aura e Elo	Bandeira de cartão de crédito
Status	opções: Novo, Cancelado, em aprovação, Pendente, Aprovado, Liberado e Remetido	Status do pedido
Feito em	smalldatetime	Data em que o pedido foi feito
Obs curta	varchar(10)	Texto de apoio sobre o pedido, para uso interno da loja

Entregue Em	smalldatetime	Data de entrega do pedido ao destinatário pela transportadora
Objeto SEDEX	varchar(13)	Código fornecido pelos Correios para acompanhamento de remessas
MoipCodigo	varchar(12)	Código da transação retornada pelo MoIP. Ex: 11202666
MoipStatus	tinyint	Status da transação retornada pelo MoIP. Ex: 5 Os valores possíveis para o campo MoipStatus: (1) Autorizado: Pagamento autorizado. (2) Iniciado: Pagamento foi iniciado, porém sem confirmação de finalização até o momento (3) BoletoImpresso: Boleto visualizado pelo cliente (4) Concluído: Pagamento creditado em conta e disponível para saque. (5) Cancelado: Pagamento foi cancelado (6) EmAnalise: Pagamento em análise de risco (7) Estornado: Pagamento foi estornado (8) EmRevisao: Pagamento em revisão pelo Moip (9) Reembolsado: Pagamento foi reembolsado
AkatusID	varchar(36)	0 ID da transação na Akatus. Ex: cfbda716-906b-4f69-83f8-4cdae19cb3ee
AkatusStatus	varchar(50)	Status da transação na Akatus. Ex: Cancelado Os valores possíveis para o campo AkatusStatus: Aguardando Pagamento, Em Análise, Aprovado e Cancelado.
ChangeFlagAPI	tinyint	Utilizado para marcar os pedidos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos.

Seguem os tipos, campos e descrições da ficha do **Detalhes**:

Coluna/Campo	Tipo/Opções	Descrição
Núm	int	Número do pedido
Feito em	smalldatetime	Data em que o pedido foi feito
Nome	varchar(50)	Nome do cliente
E-mail	varchar(50)	E-mail do cliente
Empresa	varchar(50)	Nome da empresa, somente quando B2B
CPF/CNPJ	varchar(14)	CPF ou CNPJ
RG/IE	varchar(20)	Inscrição estadual ou RG
Endereço	varchar(70)	Endereço
Bairro	varchar(25)	Bairro
Cidade	varchar(20)	Cidade
Estado	char(2)	Estado
País	varchar(20)	País
CEP	varchar(9)	CEP
Telefone	varchar(20)	Telefone
FAX	varchar(20)	FAX
Nascido em	smalldatetime	Data de nascimento
Obs cliente	varchar(10)	Texto de apoio sobre o cliente, para uso interno da loja.
Nome Pedido	varchar(50)	Nome para entrega do pedido
E-mail Pedido	varchar(50)	E-mail para entrega do pedido
Endereço Pedido	varchar(70)	Endereço de entrega
Bairro Pedido	varchar(25)	Bairro de entrega
Cidade Pedido	varchar(20)	Cidade de entrega
Estado Pedido	char(2)	Estado de entrega
País Pedido	varchar(20)	País de entrega
CEP Pedido	varchar(9)	CEP de entrega
Telefone Pedido	varchar(20)	Telefone de entrega

FAX Pedido	varchar(20)	FAX de entrega
Total sem frete	money	Valor total do pedido, SEM o frete
Frete	money	Valor do frete
Pagamento	varchar(50)	Opção de pagamento escolhida pelo cliente
Cartão	Opções: AMEX, Diners, Mastercard, VISA, Hipercard, Aura e Elo	Bandeira de cartão de crédito
Parcelas	tinyint	Número de parcelas
Presente	opções: S ou N	Se é para presente
Status	opções: Novo, Cancelado, em aprova- ção, Pendente, Aprovado, Liberado e Remetido	Status do pedido
Obs Pedido	varchar(10)	Texto de apoio sobre o pedido, para uso interno da loja
Entregue em	smalldatetime	Data de entrega do pedido ao destinatário pela transportadora
Obj SEDEX	varchar(13)	Código fornecido pelos Correios para acompanhamento de remessas
Ref. Produto	varchar(15)	Referência do produto
Nome Produto	varchar(100)	Nome do produto
Qtd	smallint	Quantidade de itens do produto
Preço unit	money	Preço unitário do produto
Msg cliente	varchar(255)	Comentários feitos pelo cliente
Local de entrega	varchar(50)	Local de entrega do pedido
Parceiro	varchar(30)	Quando o pedido foi feito através de um parceiro, informa o nome do parceiro
ThisfTransacao	varchar(14)	Código de transação com cartão de crédito recebido da Thisf, se a loja utiliza
Esp1	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 1
Esp2	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 2
Esp3	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 3
Simp1	varchar(30)	Informação adicional sobre o produto, indicado em Descritor simples 1
Simp2	varchar(30)	Informação adicional sobre o produto, indicado em Descritor simples 2
Simp3	varchar(30)	Informação adicional sobre o produto, indicado em Descritor simples 3
DescontoTotalSF	real	Desconto em % aplicado no total do pedido SEM o frete
IDProduto	int	ID interno do produto no banco de dados
Valor embalagem	smallmoney	Valor para a embalagem de presente
Cor	varchar(30)	Cor solicitada do produto
Cod. barras	varchar(25)	Código de barras do produto
ObsCurta	varchar(10)	Campo para informar um pequeno texto de observação do pedido
ccNum	varchar(19)	Número do cartão de crédito
ccNome	varchar(30)	Nome gravado no cartão de crédito
ccSeg	varchar(4)	Código de segurança do cartão de crédito
ccDataExp	varchar(7)	Mês/Ano em que o cartão expira (Ex: 05/2001)
ccAuthCod	varchar(10)	Autorização da administradora do cartão
ccNumCV	varchar(9)	Número do comprovante de venda, emitido pela administradora do cartão
ccNumAutent	varchar(27)	Número de autenticação da venda, emitido pela administradora do cartão
ccNumPrg	varchar(1)	Retorno do Komerci SecureCode (0,1,2,3,4)
ccCodRet	varchar(2)	Código de retorno do processo de autorização da venda, informado pela administra- dora do cartão
ccORIGEM_BIN	char(3)	Cód país emissor Komerci
ccEndereco	varchar(100)	Endereço reportado pela Redecard via AVS
VisaTID	char(20)	Identificação da transação na Cielo
VisaLR	varchar(4)	Cód de retorno da Cielo

VisaARP	varchar(6)	Cód aprovação bancária informado pela Cielo, somente para pedidos autorizados.
VisaCOD	varchar(4)	Código do resultado da captura do pedido informado pela Cielo
VisaAuth	varchar(3)	Indica se o pedido foi autenticado pelo banco emissor: 0 ou vazio=transação sem autenticação 1=ok 2=negado 3=falha
VisaCAP	money	Valor capturado informado pela Cielo
VisaBank	varchar(4)	Código do banco emissor do cartão
ThisfAutorizacao	varchar(6)	Código de autorização recebido pela Thisf da administradora do cartão
MoipCodigo	varchar(12)	Código da transação retornada pelo MoIP
MoipStatus	tinyint	Status da transação retornada pelo MoIP
AkatusID	varchar(36)	O ID da transação na Akatus.
AkatusStatus	varchar(50)	Status da transação na Akatus
Tipo	Opções: PF ou PJ	Indica se o cliente é pessoa física ou jurídica
Logradouro	varchar(70)	Endereço do cliente
Endereço número	varchar(70)	Número do endereço
Endereço complemento	varchar(70)	Complemento do endereço
Celular	varchar(20)	Telefone móvel completo para contato e envio SMS (com DDD)
Campo 1	varchar(100)	Campo adicional 1
Campo 2	varchar(100)	Campo adicional 2
Campo 3	varchar(100)	Campo adicional 3
Logradouro Pedido	varchar(70)	Endereço
Endereço número Pedido	varchar(70)	Número do endereço
Endereço complemento Pedido	varchar(70)	Complemento do endereço
Campo 1 Pedido	varchar(100)	Campo adicional no pedido 1
Campo 2 Pedido	varchar(100)	Campo adicional no pedido 2
Campo 3 Pedido	varchar(100)	Campo adicional no pedido 3
ThisfAutorizacao	varchar(6)	Código de autorização recebido pela Thisf da administradora do cartão
Embalar Produto	Opções: S ou N	Indica se produto deve ser embalado
Valor embalagem produto	smallmoney	Valor para embalagem do produto
Desconto Cupom	real	Valor em \$ do desconto no pedido
ChangeFlagAPI	tinyint	Utilizado para marcar os pedidos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos.
Alterado em	smalldatetime	Data da última alteração do pedido
Logradouro	varchar(70)	Endereço do cliente

Seguem os tipos, campos e descrições da ficha de **Acessos**:

Method	Opções: ReportView, OrderUpdate, ProductManagement, NfeOperation, UtilityExecute.	Indica o método
Nome	varchar(80)	Indica o nome do relatório, utilitário.
Data execução	datetime	Data/hora de início do processamento
Usuário	varchar(20)	Nome do usuário
Regs	int	Quantidade de registros lidos, incluídos, alterados ou excluídos
Parâm 1	varchar(15)	Legenda para o parâmetro 1
Valor 1	varchar(100)	Valor informado no parâmetro 1
Parâm 2	varchar(15)	Legenda para o parâmetro 2
Valor 2	varchar(100)	Valor informado no parâmetro 2
Parâm 3	varchar(15)	Legenda para o parâmetro 3
Valor 3	varchar(100)	Valor informado no parâmetro 3

Parâm 4	varchar(15)	Legenda para o parâmetro 4
Valor 4	varchar(100)	Valor informado no parâmetro 4
Parâm 5	varchar(15)	Legenda para o parâmetro 5
Valor 5	varchar(100)	Valor informado no parâmetro 5
Parâm 6	varchar(15)	Legenda para o parâmetro 6
Valor 6	varchar(100)	Valor informado no parâmetro 6
Parâm 7	varchar(15)	Legenda para o parâmetro 7
Valor 7	varchar(100)	Valor informado no parâmetro 7
Parâm 8	varchar(15)	Legenda para o parâmetro 8
Valor 8	varchar(100)	Valor informado no parâmetro 8

7

Melhores práticas

Seguem algumas dicas para melhor utilização da API:

- 1) Para maximizar a performance na alteração de produtos, inclua no XML somente os campos obrigatórios Comando e ID-Produto e os campos que serão alterados. Não inclua no XML campos que não serão alterados.
 - 2) Existe um limite de até 300 chamadas à API por hora. Cada acesso contabiliza um pageview, independente da quantidade de registros incluídos, alterados ou removidos. Para maximizar o uso de cada chamada aos métodos OrderUpdate e ProductManagement, é possível incluir mais de um registro que será afetado em cada chamada. Por exemplo, é possível incluir, alterar e excluir vários produtos em uma mesma chamada ao método ProductManagement.
 - 3) Nos métodos UtilityExecute e ReportView, preencha os parâmetros de data sempre que estes existirem no objeto chamado. Informe sempre o menor período de tempo possível, para maximizar a performance e evitar timeouts.
 - 4) É possível informar a hora nos parâmetros de data. Se somente a data for informada, sem a hora, esta será considerada 00:00 (meia-noite).
- Exemplo:** Para obter todos os pedidos do dia 15/09/2011, informe De:15/9/2011 00:00 Até:16/9/2011 00:00 ou, de forma equivalente, De:15/9/2011 Até:16/9/2011.
- 5) Os relatórios do FastCommerce podem ser acessados via XML, através do método ReportView. Frequentemente incluímos novos campos nas dezenas de relatórios do FastCommerce, sem aviso prévio. Por esta razão, os sistemas integrados ao FastCommerce não devem ser afetados por estas inclusões de campos, independente da posição.

Uma programação bem feita de leitura de XML não deve ser "amarrada" com relação à posição do campo, dentro da mesma hierarquia. Se acrescentarmos campos antes ou depois do campo desejado, sem alterar a estrutura do XML, seu programa deve ser capaz de acessar normalmente os nós pré-existentes.

O XML não deve ser lido como uma grande "string". Ao invés disto, seu programa deve manipular o XML como uma estrutura hierárquica, com acesso direto e individualizado a qualquer nó, independente dos nós anteriores e posteriores. Uma sugestão é utilizar objeto para leitura e "parse" do XML, como por exemplo, o Microsoft.XMLDOM, que usamos no FastCommerce.

XML DOM Reference

<http://msdn.microsoft.com/pt-br/library/aa925430.aspx>

