

Sobre a API

Fastcommerce Developers Center

1 - Acessando a API
2 - Histórico de Versões

3 - Loja para testes
4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

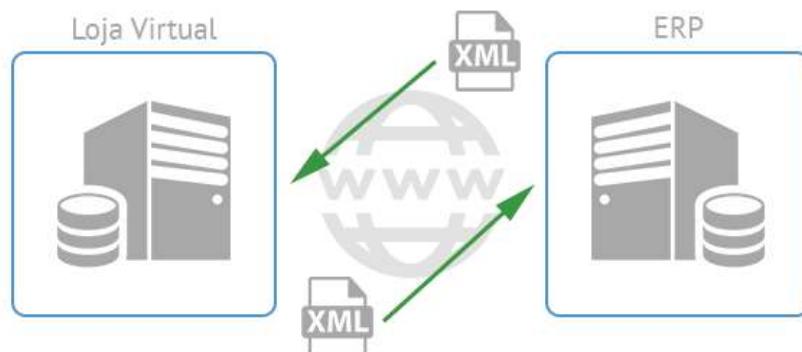
12 - Tratamento de Erros

Performance máxima nas vendas online.

Sobre a API

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.

Integração via API



1 - Acessando a API

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectId 510

ObjectId 425

ObjectId 429

ObjectId 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

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.

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:

Existe um limite no volume de acessos por hora aos métodos da API. Veja os detalhes no parágrafo 2 do item [Melhores Práticas](#)).

2 - Histórico de Versões

A versão atual da API é 2.3.

Versão	Data	Detalhes
2.3	01/06/2022	Inclusão do comando D para que seja possível excluir os arquivos associados ao cadastro de produto
2.2	05/12/2015	Alteração: ReadRecords não exibe mais o total de registros existentes na consulta e sim a quantidade de registros retornados
2.1	13/06/2015	Inserção: Nova opção de formato de relatório JSON (OutputFormat=6)
2.0	13/11/2012	<p>Inserção:</p> <ul style="list-style-type: none"> - 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) - Campos retornados pela API na chave,

	Versão	Data	Detalhes
Sobre a API			retorno fracionado e seleção de campo retornados.
1 - Acessando a API	1.9	08/10/2012	Inserção: Novos campos em Produto, Clientes, Pedidos, Detalhes e Acessos
2 - Histórico de Versões			Alteração: Limite de 60 para 300 acessos por hora.
3 - Loja para testes			Melhores Práticas
4 - Usuário para testes	1.8	24/05/2012	Inserção: Novo campo
5 - Melhores Práticas			ChangeFlagProdAPI na tabela de campo de produto
6 - Métodos da API			Inserção: Nova chave ChangeFlagAPI nos XMLs de exemplo de alteração de
7 - Permissões de acesso	1.7	14/03/2012	pedidos: <Field Name="ChangeFlagAPI" Value="99" />
8 - Exemplos de XML			Alteração: Novas bandeiras no campo cartão em Pedidos.
9 - Realizando Testes			Criação do documento
10 - Exemplos			
11 - Dicionário de Dados	1.6	11/10/2011	
ObjectId 510			
ObjectId 425	1.5	19/04/2010	
ObjectId 429			
ObjectId 426			
ObjectId 427			
ObjectId 248			O Fastcommerce não tem um ambiente de <i>sandbox</i> para
ObjectId 393			realizar testes. Sugerimos portanto que crie uma loja para testes
ObjectId 530			para desenvolver com a API.
12 - Tratamento de Erros			Acesse o endereço http://novaloja.fastcommerce.com.br para
			criar sua loja de testes. Após cadastrá-la, avise-nos qual é o
			nome e o ID desta nova loja através do e-mail
			comercial@fastcommerce.com.br para que possamos deixar a
			loja em implantação e a mesma não seja cancelada após o
			prazo de avaliação.

3 - Loja para testes

O Fastcommerce não tem um ambiente de *sandbox* para realizar testes. Sugerimos portanto que crie uma [loja para testes](#) para desenvolver com a API.

Acesse o endereço <http://novaloja.fastcommerce.com.br> para criar sua loja de testes. Após cadastrá-la, avise-nos qual é o nome e o ID desta nova loja através do e-mail comercial@fastcommerce.com.br para que possamos deixar a loja em implantação e a mesma não seja cancelada após o prazo de avaliação.

4 - Usuário para testes

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados

ObjectId 510

ObjectId 425

ObjectId 429

ObjectId 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

- 12 - Tratamento de Erros

O usuário que vai consumir a API precisa basicamente de acesso aos relatórios e utilitários da loja, pois é através dele que as consultas e alterações são feitas.

Essa etapa deve ser realizada por um usuário da loja com permissão para criar novos usuários.

Acesse a administração da loja e no menu "Usuários" clique na opção "Incluir usuário".

O cadastro pode ser preenchido como no exemplo abaixo.

5 - Melhores Práticas

Seguem algumas dicas para melhor utilização da API:

5.1) Para maximizar a performance na alteração de produtos, inclua no XML somente os campos obrigatórios Comando e IDProduto e os campos que serão alterados. Não inclua no XML campos que não serão alterados.

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

5.2) Existe um limite de até 300 chamadas à API por hora. 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. 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.*

5.3) 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*.

5.4) 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.

5.5) 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.

5.6) Uma programação bem feita de leitura de XML não deve ser "amarrada" com relação à posição do

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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.

5.7) 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.

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

5.8) O XML de envio da API do FastCommerce tem **limite de 1Mb.**

6 - Métodos da API

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 indicado em OutputFormat (padrão é 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:

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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

Caso o método seja ReportView ou UtilityExecute os campos são:

	Nome do Campo	Descrição
7 - Permissões de acesso	ObjectID	ID do Relatório/Utilitário (Opcional se ObjectName for informado)
8 - Exemplos de XML	ObjectName	Nome do Relatório/Utilitário (Opcional se ObjectID for informado)
9 - Realizando Testes	Par1	Parâmetro 1 do relatório/utilitário (Ex.: Data Inicial)
10 - Exemplos	Par2	Parâmetro 2 do relatório/utilitário (Ex.: Data Final)
11 - Dicionário de Dados	Par3	Parâmetro 3 do relatório/utilitário (Ex.: Nome do Cliente)
ObjectID 510	Par4	Parâmetro 4 do relatório/utilitário (Ex.: e-mail do cliente)
ObjectID 425	Par5	Parâmetro 5 do relatório/utilitário
ObjectID 429	Par6	Parâmetro 6 do relatório/utilitário
ObjectID 426	Par7	Parâmetro 7 do relatório/utilitário
ObjectID 427	Par8	Parâmetro 8 do relatório/utilitário
ObjectID 248	Page	Página que será retornada
ObjectID 393	QtRecords	Quantidade de registros retornados por página
ObjectID 530	FromRecord	Registro Inicial solicitado do relatório
12 - Tratamento de Erros	ToRecord	Registro final solicitado do relatório

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Fields Campos que serão retornados (Apenas ReportView)

Formato do retorno:

0: HTML

1: XML

2: AJAX

3: Excel

4: Word

5: XML API (formato default)

6: JSON

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/s
    <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">
    <input type="text" name="ObjectID" value="425">> <!--
    <input type="text" name="ObjectName" value="Lista de Produtos">
    <input type="text" name="Par1" value="Nome da categoria">
    <input type="text" name="Par2" value="Ref/Nome/Descrição">
    <input type="text" name="Par3" value="false">> <!--
    <input type="text" name="Par4" value="false">> <!--
    <input type="text" name="Par5" value="11/05/2016">
    <input type="text" name="Par6" value="12/05/2016">
    <input type="text" name="Par7" value="1">> <!-- opção de paginação
    <input type="text" name="Page" value="1">> <!-- opção de paginação
```

```
<input type="text" name="QtRecords" value="100"> <
</form>
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Se o ObjectId for informado, não é necessário informar o ObjectName.

Para ver a lista completa de relatórios e utilitários disponíveis, seus respectivos ObjectId e os parâmetros de cada um deles, utilize o relatório **(FC) Lista de relatórios e utilitários**.

Qualquer relatório ou utilitário pode ser executado, mas existem 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
- Dados da loja para consultas via API

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

Nome do Campo	Descrição
XMLRecords	Conteúdo do XML com todos seus registros

Exemplo de FORM:

```
<form name="Logon" action="https://www.rumo.com.br/s
<input type="text" name="StoreName" value="Nome da
<input type="text" name="StoreID" value="99999">
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

```
<input type="text" name="Username" value="Admin">
<input type="password" name="Password" value="*****">
<input type="text" name="method" value="OrderUpdate">
<input type="text" name="XMLRecords" value="Conteúdo XML">
</form>
```

7 - Permissões de acesso

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

Método	Permissão de usuário
OrderUpdate	Acesso aos utilitários de pedidos
ProductManagement	Acesso aos utilitários de produtos
ReportView	Acesso ao tipo de relatório solicitado
UtilityExecute	Acesso ao tipo de utilitário selecionado

Exemplo: Se o **ObjectID** escolhido for um relatório estatístico, o usuário precisa ter acesso aos relatórios estatísticos.

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.

Obs: É **extremamente recomendado** criar um usuário específico para cada aplicação que for utilizar a API. Este usuário deve ter nome de usuário e **senha fortes e únicas**. Insira as permissões de acordo com que a aplicação irá utilizar. Por exemplo, se a aplicação irá apenas acessar os dados de produtos, basta inserir acesso a relatórios de produtos e em todos os demais campos deixar sem acesso.

8 - Exemplos de XML

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectID 510

 ObjectID 425

 ObjectID 429

 ObjectID 426

ObjectID 427

 ObjectID 248

 ObjectID 393

 ObjectID 530

12 - Tratamento de Erros

8.1 - Inclusão de produtos

Estrutura do XML para inclusão do cadastro de produtos:

Para inclusão o comando deve ser I e devem ser enviados 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 Brazi
        <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 Itali
        <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>

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

```

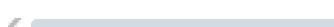
<Record Num="1">
  <Field Name="COMANDO" Value="I" />
  <Field Name="NOMECAT" Value="Camisetas" />
  <Field Name="CODPROD" Value="ABC0001" />
  <Field Name="NOMEPROD" Value="Polo Seleção Brazi" />
  <Field Name="CHANGEFLAGPRODAPI" Value="1" />
  <Field Name="IDPRODUTO" Value="888888" />
</Record>

<Record Num="2">
  <Field Name="COMANDO" Value="I" />
  <Field Name="NOMECAT" Value="Camisetas" />
  <Field Name="CODPROD" Value="ABC0002" />
  <Field Name="NOMEPROD" Value="Polo Seleção Itali" />
  <Field Name="CHANGEFLAGPRODAPI" Value="1" />
  <Field Name="IDPRODUTO" Value="999999" />
</Record>

<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="2" Valid="2" Included="2" IncludedSut="2" />

```

</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>**):

ErrCode	ErrDescr
0	OK
1	Store Not Found
2	Invalid User
3	No Password
4	Logon Error

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

5	Blocked IP Address
6	Too many retries. User suspended for 3 minutes
7	Logon Error. Next invalid logon will suspend user for 3 minutes.
8	Invalid method 19 Invalid Order(s) in XMLRecords
9	Report/Utility not found or access denied 20
10	Reached limit of access per hour
11	API Access denied
12	Report/Utility error
13	Invalid StoreID for this logon
14	StoreID not informed
15	Store suspended
16	Demonstration period is over
17	XMLRecords not informed
18	Access denied
	Invalid Product(s) in XMLRecords

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 Brazi
    <Field Name="CHANGEFLAGPRODAP" Value="1" />
    <Field Name="IDPRODUTO" Value="888888" />
  </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 Itali
    <Field Name="CHANGEFLAGPRODAP" Value="1" />
```

```

<Field Name="IDPRODUTO" Value="999999" />
</Record>
<ErrCode>18</ErrCode>
<ErrDescr>Invalid Product(s) in XMLRecords</ErrDes
<IntCode>119</IntCode>
<IntDescr>IDProduto should not be informed</IntDes
<Stats Sent="2" Valid="1" Included="1" IncludedSu&lt;
</API>

```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Sempre será retornado os registros anteriores que obtiveram êxito e por último o registro no qual ocorreu o erro. Neste caso, no segundo produto foi enviado o campo IDProduto indevidamente para uma inclusão de produto.

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

IntCode	IntDescr
111	Campo inválido
112	Campo duplicado
113	Excedeu máximo de registros
114	Sem nome da categoria nem IDCategoria no registro para inclusão
115	Comando inválido ou não informado
116	Atingiu máximo de produtos do plano
117	NomeCat do sub-produto não existe
118	Tipo de campo inválido, nunca deveria ocorrer
119	Informado campo IDProduto na inclusão
120	IDProduto não informado para alteração e exclusão
121	IDProdutoPai não deve ser informado pois a loja não tem sub-produtos
122	IDProdutoPai não encontrado
123	IDProduto não encontrado
124	IDProdutoPai não pode ser alterado

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectId 510

 ObjectId 425

 ObjectId 429

 ObjectId 426

ObjectId 427

 ObjectId 248

 ObjectId 393

 ObjectId 530

12 - Tratamento de Erros

125	Problema na exclusão
126	IDProduto não encontrado em UPDATE
127	IDCategoria não encontrado
128	Não permite mudar o sub-produto de pai
129	Máximo de sub-produtos no produto atingido

Na tag <Stats> são retornados os seguintes atributos:

Atributo	Descrição
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 deletados
DeletedSub	Número de sub-produtos deletados
ElapsedSeconds	Tempo decorrido da operação

8.2 - 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 Brazi" />
    <Field Name="ChangeFlagProdAPI" Value="1" />
  </Record>
  <Record>
    <Field Name="Comando" Value="A" />
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

```
<Field Name="IDProduto" Value="888888" />
<Field Name="NomeProd" Value="Polo Seleção Itali
<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 Brazi
<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 Itali
<Field Name="CHANGEFLAGPRODAPI" Value="1" />
</Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="2" Valid="2" Included="0" IncludedSu&
</API>
```



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

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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>

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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>**):

	ErrCode	ErrDescr
0	OK	
1	Store not found	
2	Invalid User	
3	No Password	
4	Logon Error	
5	Blocked IP Address	
6	Too many retries. User suspended for 3 minutes	
7	Logon Error. Next invalid logon will suspend user for 3 minutes.	
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	
11	Report/Utility error	

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

12	Invalid StoreID for this logon
13	StoreID not informed
14	Store suspended
15	Demonstration period is over
16	XMLRecords not informed
17	Access denied
18	Invalid Product(s) in XMLRecords

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 Brazi
    <Field Name="CHANGEFLAGPRODAP" Value="1" />
  </Record>
  <Record Num="2">
    <Field Name="COMANDO" Value="A" />
    <Field Name="IDPRODUTO" Value="888888" />
    <Field Name="NOMEPROD" Value="Polo Seleção Itali
    <Field Name="CHANGEFLAGPRODAP" Value="1" />
  </Record>
  <ErrCode>18</ErrCode>
  <ErrDescr>Invalid Product(s) in XMLRecords</ErrDes
  <IntCode>111</IntCode>
  <IntDescr>Invalid field:NOMEDOCAMPO Record:2</Int[
  <Stats Sent="2" Valid="1" Included="0" IncludedSu[
</API>
```

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

IntCode	IntDescr
---------	----------

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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

Na tag <Stats> são retornados os seguintes atributos:

Atributo	Descrição
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

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
- 12 - Tratamento de Erros

8.3 - Exclusão de produtos

Estrutura do XML para exclusão do cadastro de produtos:

*Para exclusão o comando deve ser **D** ou **E** e devem ser enviados somente os campos Comando e IDProduto.*

Comando **D**: efetua a exclusão do cadastro do produto, as imagens e arquivo html de descrição.

Comando **E**: efetua apenas a exclusão do cadastro do produto. As imagens e arquivo html de descrição permanecem no gerenciador de arquivos da loja.

```
<Records>
  <Record>
    <Field Name="Comando" Value="D" />
    <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="D" />
```

```

<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="2" Valid="2" Included="0" IncludedSuf="0" />
</API>

```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

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

ErrCode	ErrDescr
0	OK
1	Store not found
2	Invalid User
3	No Password
4	Logon Error
5	Blocked IP Address
6	Too many retries. User suspended for 3 minutes
7	Logon Error. Next invalid logon will suspend user for 3 minutes.
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
11	Report/Utility error
12	Invalid StoreID for this logon
13	StoreID not informed
14	Store suspended
15	Demonstration period is over

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

16	XMLRecords not informed
17	Access denied
18	Invalid Product(s) in XMLRecords

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" />
</Record>
<Record Num="2">
  <Field Name="COMANDO" Value="E" />
  <Field Name="IDPRODUTO" Value="888888" />
</Record>
<ErrCode>18</ErrCode>
<ErrDescr>Invalid Product(s) in XMLRecords</ErrDes
<IntCode>123</IntCode>
<IntDescr>IDProduto not found</IntDescr>
<Stats Sent="2" Valid="1" Included="0" IncludedSut
</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>**):

IntCode	IntDescr
111	Campo inválido
112	Campo duplicado
113	Excedeu máximo de registros
114	Sem nome da categoria nem IDCategoria no registro para inclusão
115	Comando inválido ou não informado

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

116	Atingiu máximo de produtos do plano
117	NomeCat do sub-produto não existe
118	Tipo de campo inválido, nunca deveria ocorrer
119	Informado campo IDProduto na inclusão
120	IDProduto não informado para alteração e exclusão
121	IDProdutoPai não deve ser informado pois a loja não tem sub-produtos
122	IDProdutoPai não encontrado
123	IDProduto não encontrado
124	IDProdutoPai não pode ser alterado
125	Problema na exclusão
126	IDProduto não encontrado em UPDATE
127	IDCategoria não encontrado
128	Não permite mudar o sub-produto de pai
129	Máximo de sub-produtos no produto atingido

Na tag <Stats> são retornados os seguintes atributos:

	Atributo	Descrição
ObjectId 426	Sent	Número de registros
ObjectId 427	Valid	Número de registros válidos
ObjectId 248	Included	Número de produtos incluídos
ObjectId 393	IncludedSub	Número de sub-produtos incluídos
ObjectId 530	Changed	Número de produtos alterados
12 - Tratamento de Erros	ChangedSub	Número de sub-produtos alterados
	Deleted	Número de produtos excluídos
	DeletedSub	Número de sub-produtos excluídos
	ElapsedSeconds	Tempo decorrido da operação

8.4 - Alteração de pedidos

Estrutura do XML para alteração na ficha dos pedidos:

*Para alteração, o campo **NumPedido** é obrigatório.*

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

```
<Records>
  <Record>
    <Field Name="NumPedido" Value="999999" />
    <Field Name="ObsCurta" Value="abcdefg" />
    <Field Name="Status" Value="4" />
    <Field Name="ObjSedex" Value="a9b9c9d9e9" />
    <Field Name="NumNFe" Value="12345678901234567890" />
    <Field Name="DataEntrega" Value="10/10/2018" />
    <Field Name="ObsLog" Value="Alteração de pedido." />
    <Field Name="ChangeFlagAPI" Value="98" />
  </Record>
  <Record>
    <Field Name="NumPedido" Value="888888" />
    <Field Name="ObsCurta" Value="ijlmnopq" />
    <Field Name="Status" Value="7" />
    <Field Name="ObjSedex" Value="a8b8c8d8e8" />
    <Field Name="NumNFe" Value="12345678901234567890" />
    <Field Name="DataEntrega" Value="10/10/2018" />
    <Field Name="ObsLog" Value="Alteração de pedido." />
    <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" />
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

```

<Field Name="NUMNFE" Value="12345678901234567890">
<Field Name="DATAENTREGA" Value="10/10/2018" />
<Field Name="OBSLOG" Value="Alteração de pedido." />
<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="NUMNFE" Value="12345678901234567890">
<Field Name="DATAENTREGA" Value="10/10/2018" />
<Field Name="OBSLOG" Value="Alteração de pedido." />
<Field Name="CHANGEFLAGAPI" Value="99" />
</Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="2" Valid="2" Changed="2" ElapsedSecor
</API>
```

8.5 - 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

ObsLog - Campo para informar mais detalhes sobre a observação do pedido (até 255 caracteres).

Status - Status para o qual o pedido será alterado.

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

	Valor	Status
Sobre a API	1	Novo
1 - Acessando a API	2	Cancelado
2 - Histórico de Versões	3	Em aprovação
3 - Loja para testes	4	Pendente
4 - Usuário para testes	5	Aprovado
5 - Melhores Práticas	6	Liberado
6 - Métodos da API	7	Remetido

7 - Permissões de acesso	ObjSedex - Código do objeto dos Correios, para acompanhamento da remessa pela loja e pelo cliente
8 - Exemplos de XML	NumNFe - Número/Chave da nota fiscal eletrônica com 44 posições
9 - Realizando Testes	DataEntrega - Utilizado para informar a data da entrega do pedido
10 - Exemplos	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
11 - Dicionário de Dados	Os seguintes relatórios de pedidos podem ser utilizados com o método ReportView para trazer a lista de pedidos da loja:
12 - Tratamento de Erros	<ul style="list-style-type: none"> • Lista de pedidos para alterações via API • Lista de pedidos e detalhes para consultas via API • Lista de pedidos alterados para consultas via API <p>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.</p>

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

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

	ErrCode	ErrDescr
1 - Acessando a API	0	OK
2 - Histórico de Versões	1	Store not found
3 - Loja para testes	2	Invalid User
4 - Usuário para testes	3	No Password
5 - Melhores Práticas	4	Logon Error
6 - Métodos da API	5	Blocked IP Address
7 - Permissões de acesso	6	Too many retries. User suspended for 3 minutes
8 - Exemplos de XML	7	Logon Error. Next invalid logon will suspend user for 3 minutes.
9 - Realizando Testes	8	Invalid method 19 Invalid Order(s) in XMLRecords
10 - Exemplos	9	Report/Utility not found or access denied 20
11 - Dicionário de Dados	10	Reached limit of access per hour
ObjectID 510	10	API Access denied
ObjectID 425	11	Report/Utility error
ObjectID 429	12	Invalid StoreID for this logon
ObjectID 426	13	StoreID not informed
ObjectID 427	14	Store suspended
ObjectID 248	15	Demonstration period is over
ObjectID 393	16	XMLRecords not informed
ObjectID 530	17	Access denied
12 - Tratamento de Erros	18	Invalid Product(s) in XMLRecords

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

<API>

```
<Record Num="1">
  <Field Name="NUMPEDIDO" Value="999999" />
  <Field Name="OBSCURTA" Value="abcdefg" />
  <Field Name="STATUS" Value="9" />
```

Sobre a API

- 1 - Acessando a API
 - 2 - Histórico de Versões
 - 3 - Loja para testes
 - 4 - Usuário para testes
 - 5 - Melhores Práticas
 - 6 - Métodos da API
 - 7 - Permissões de acesso
 - 8 - Exemplos de XML
 - 9 - Realizando Testes
 - 10 - Exemplos
 - 11 - Dicionário de Dados

```
<Field Name="OBJSEDEX" Value="a9b9c9d9e9" />
<Field Name="NUMNFE" Value="12345678901234567890" />
<Field Name="DATAENTREGA" Value="10/10/2018" />
</Record>
<Record Num="2">
<Field Name="NUMPEDIDO" Value="8888881" />
<Field Name="OBSCURTA" Value="ijlmopq" />
<Field Name="STATUS" Value="9" />
<Field Name="OBJSEDEX" Value="a8b8c8d8e8" />
<Field Name="NUMNFE" Value="12345678901234567890" />
<Field Name="DATAENTREGA" Value="10/10/2018" />
</Record>
<ErrCode>19</ErrCode>
<ErrDescr>Invalid Order(s) in XMLRecords</ErrDescr>
<IntCode>53</IntCode>
<IntDescr>NumPedido not found</IntDescr>
<Stats Sent="2" Valid="1" Changed="1" ElapsedSecor
</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>**):

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

Na tag <Stats> são retornados os seguintes atributos:

- Sobre a API
- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
- 12 - Tratamento de Erros

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

9 - Realizando Testes

Disponibilizamos um arquivo com a coleção de requisições dos métodos da API para o aplicativo Postman.

O ambiente possui um conjunto de variáveis que são armazenadas como pares chave/valor com os dados para acesso à API que podem ser alterados para os dados da loja em que os testes serão feitos (*StoreName*, *StoreId*, *Username* e *Password*).

[Clique aqui](#) para acessar o Postman (**Obs:** acentos só funcionam se codificados em ISO-8859-1 e o Postman só usa UTF-8)

Clique nos nomes dos arquivos para baixar as coleções:

[API_Fastcommerce.postman_collection.json](#)

[API_Fastcommerce.postman_environment.json](#)

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

[API Test](#)

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

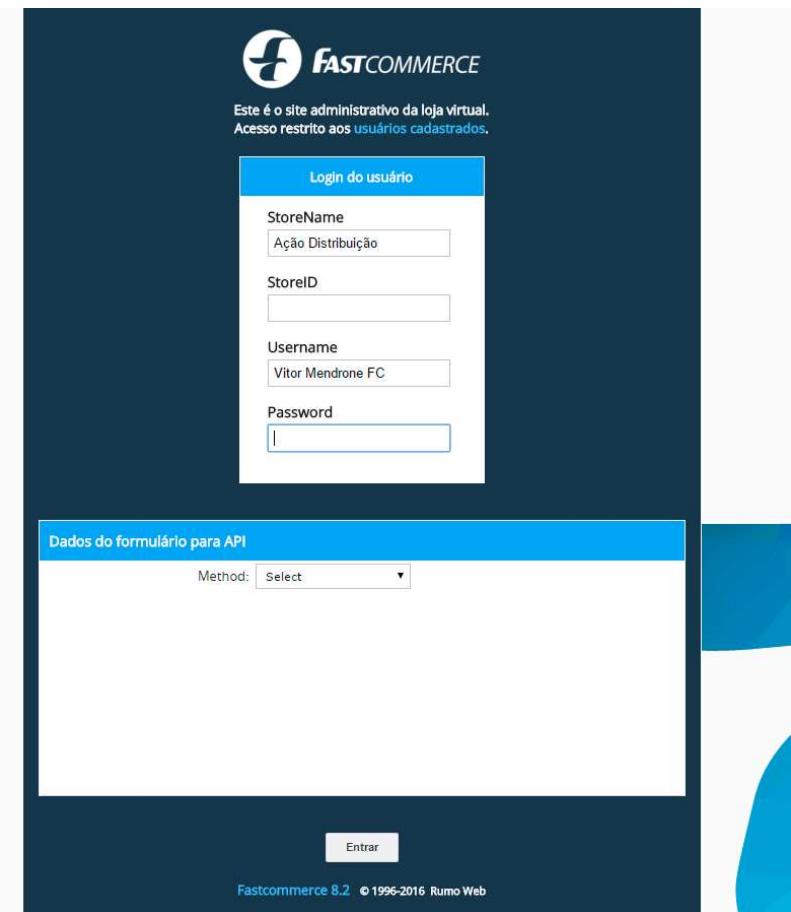
ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

- 12 - Tratamento de Erros



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.

9.1 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**.

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
- 12 - Tratamento de Erros

The screenshot shows the Fastcommerce Admin interface. At the top, there's a logo and a message: "Este é o site administrativo da loja virtual. Acesso restrito aos usuários cadastrados." Below this is the "Login do usuário" (User Login) form with fields for StoreName, StoreID, Username, and Password. To the right of the login form is a large red arrow pointing towards the "Dados do formulário para API" (API Form Data) screen. This second screen has a blue header "Dados do formulário para API". It contains a dropdown "Method" set to "ReportView", a text input "ObjectID" containing "731", and a dropdown "OutputFormat" set to "API". Below these are several input fields labeled "Par1" through "Par8". At the bottom of this screen are "Page:" and "QtRecords:" inputs, both with a dropdown arrow, and a "Entrar" (Enter) button.

Campos retornados pela API na chave <Report>:

Campo	Descrição
ObjectName	Nome do relatório
StoreName	Nome da loja
ExecDate	Data de execução do relatório
FromRecord	Registro inicial retornado

- Sobre a API
- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
- 12 - Tratamento de Erros

ToRecord	ParName1, ParName2, ParName3, ParName4, ParName5, ParName6, ParName7, ParName8	Registro final retornado
	ParValue1, ParValue2, ParValue3, ParValue4, ParValue5, ParValue6, ParValue7, ParValue8	Nomes dos parâmetros
Records		Valores dos parâmetros
ReadRecords		Registros Retornados
		Registros Retornados

Como resultado, teremos a seguinte estrutura:

```
<Report ObjectName="Lista de produtos para alterações">
  <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

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

O XML de retorno da API do FastCommerce tem **limite de 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 retornados em "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 de registros retornados em ReadRecords, como no exemplo abaixo:

```
<Report ObjectName="Lista de produtos para alterações">
```



No exemplo acima, foram retornados 500 registros de produtos e foram retornados do registro 1 ao 500. Solicitando Page=2 e QtRecords=500, serão retornados do registro 501 até o registro 1000, e assim por diante.

A dupla de parâmetros Page=2 e QtRecords=500 equivalem aos parâmetros FromRecord=501 e ToRecord=1000.

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Se forem solicitados 500 registros e retornarem menos do que 500 registros em ReadRecords, a página retornada foi a última.

Exemplo:

```
<Report ObjectName="Lista de produtos para alterações via API">
```

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.

9.2 Seleção de campos retornados

Cada relatório retorna uma 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.

9.3 Método UtilityExecute

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Neste teste, o estoque de todos os produtos serão alterados para 15.

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**.

The screenshot shows the Fastcommerce Admin interface. The top part is the 'Login do usuário' (User Login) screen, which includes fields for StoreName (Exemplo API), StoreID (55555), Username (admin), and Password. A red arrow points from step 4 to the 'Password' field. The bottom part is the 'Dados do formulário para API' (API Form Data) screen. It has a dropdown for Method set to 'UtilityExecute'. The ObjectId field contains '445'. The Par4 field contains '15', which is highlighted with a blue selection bar. A red arrow points from step 5 to the 'Method' dropdown. Another red arrow points from step 6 to the 'ObjectId' field. A third red arrow points from step 7 to the 'Par4' field. At the bottom of the form is a 'Entrar' (Enter) button.

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

<API>

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

<ErrCod>0<ErrCod>

<ErrDescr>OK<ErrDescr>

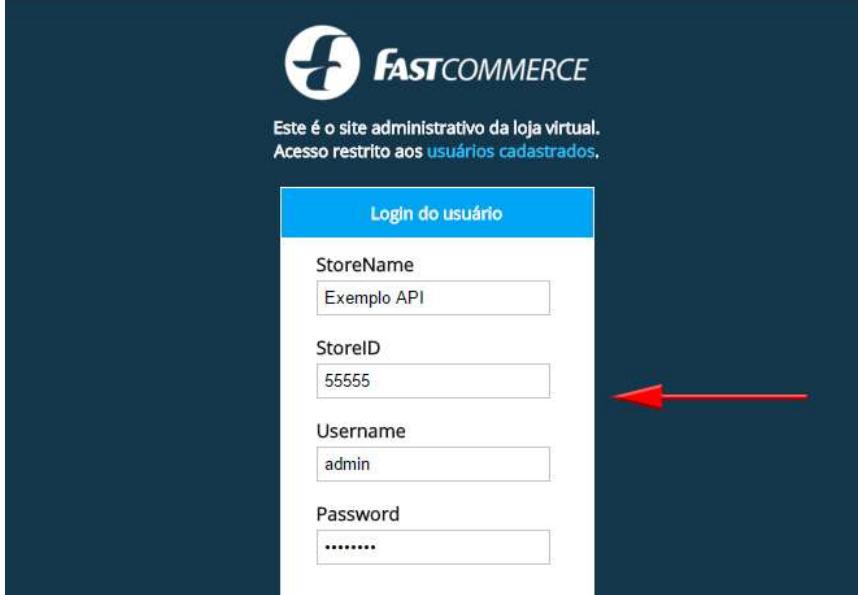
<Regs>10<Regs>

</API>

9.4 Método OrderUpdate

Neste teste, serão alterados dois pedidos

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



Dados do formulário para API

Method: **OrderUpdate** (highlighted by a red arrow)

XMLRecords:

```
<Records>
<Record>
<Field Name="NumPedido" Value="99999" />
<Field Name="ObsCurta" Value="Fraude" />
<Field Name="Status" Value="5" />
<Field Name="ObjSedex" Value="a1b2c3d4" />
</Record>
<Record>
<Field Name="NumPedido" Value="88888" />
<Field Name="ObsCurta" Value="Exemplo" />
<Field Name="Status" Value="5" />
<Field Name="ObjSedex" Value="sh5552223333" />
</Record>
</Records>
```

Entrar

Fastcommerce 8.2 © 1996-2016 Rumo Web

Como resultado, teremos a seguinte estrutura:

Sobre a API

- 1 - Acessando a API
 - 2 - Histórico de Versões
 - 3 - Loja para testes
 - 4 - Usuário para testes
 - 5 - Melhores Práticas
 - 6 - Métodos da API
 - 7 - Permissões de acesso
 - 8 - Exemplos de XML
 - 9 - Realizando Testes
 - 10 - Exemplos
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
 - 12 - Tratamento de Erro

```
<API>
<Record Num="1">
  <Field Name="NUMPEDIDO" Value="99999" />
  <Field Name="OBSCURTA" Value="Fraude" />
  <Field Name="STATUS" Value="5" />
  <Field Name="OBJSEDEX" Value="a1b2c3d4" />
  <Field Name="NUMNFE" Value="12345678901234567890" />
  <Field Name="DATAENTREGA" Value="10/10/2018" />
</Record>
<Record Num="2">
  <Field Name="NUMPEDIDO" Value="88888" />
  <Field Name="OBSCURTA" Value="Exemplo" />
  <Field Name="STATUS" Value="5" />
  <Field Name="OBJSEDEX" Value="sh5552223333" />
  <Field Name="NUMNFE" Value="12345678901234567890" />
  <Field Name="DATAENTREGA" Value="10/10/2018" />
</Record>
<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="2" Valid="2" Changed="2" ElapsedSecor
</API>
```

A set of small, light-gray navigation icons typically found in LaTeX Beamer presentations, including symbols for back, forward, search, and table of contents.

9.5 Método ProductManagement

12 - Tratamento de Erros

Neste teste, serão alterados dois produtos.

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

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados
 - ObjectId 510
 - ObjectId 425
 - ObjectId 429
 - ObjectId 426
 - ObjectId 427**
 - ObjectId 248
 - ObjectId 393
 - ObjectId 530
- 12 - Tratamento de Erros

The screenshot shows the Fastcommerce Admin interface. At the top, there's a logo and a message: "Este é o site administrativo da loja virtual. Acesso restrito aos usuários cadastrados." Below this is a "Login do usuário" form with fields for StoreName (Exemplo API), StoreID (55555), Username (admin), and Password (*****). Two red arrows point to the "StoreID" field and the "Username" field.

Below the login is a "Dados do formulário para API" section. It has a dropdown for "Method: ProductManagement". Two red arrows point to this dropdown and the XML code below it. The XML code is as follows:

```

<Records>
<Record>
<Field Name="Comando" Value="A" />
<Field Name="NomeCat" Value="Camisetas" />
<Field Name="IDCategoria" Value="2222" />
<Field Name="IDProduto" Value="99999" />
</Record>
<Record>
<Field Name="Comando" Value="A" />
<Field Name="NomeCat" Value="Bermudas" />
<Field Name="IDCategoria" Value="3333" />
<Field Name="IDProduto" Value="88888" />
</Record>
</Records>

```

At the bottom right of this section is a "Entrar" button. The footer of the page says "Fastcommerce 8.2 © 1996-2016 Rumo Web".

Como resultado, teremos a seguinte estrutura:

```

<API>
<Record Num="1">
<Field Name="COMANDO" Value="A" />
<Field Name="NOMEAT" Value="Camisetas" />
<Field Name="IDCATEGORIA" Value="2222" />
<Field Name="IDPRODUTO" Value="99999" />
</Record>
<Record Num="2">
<Field Name="COMANDO" Value="A" />
<Field Name="NOMEAT" Value="Bermudas" />
<Field Name="IDCATEGORIA" Value="3333" />
<Field Name="IDPRODUTO" Value="88888" />
</Record>

```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

```

<ErrCode>0</ErrCode>
<ErrDescr>OK</ErrDescr>
<Stats Sent="2" Valid="2" Included="0" IncludedSuf="0" />
</API>

```

10 - Exemplos

VB Script

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 "ListarDepartamentos"
```

```
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='ItemID' Value='1' /><Field Name='Quantity' Value='10' /></Record></Records>"
```

```
Name=""ObsCurta"" Value=""Cheque OK""/><Field Name=""Value=""SDX0002999""/></Record></Records>"
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectId 510

 ObjectId 425

 ObjectId 429

 ObjectId 426

ObjectId 427

 ObjectId 248

 ObjectId 393

 ObjectId 530

12 - Tratamento de Erros

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)
```

```
    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
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

11 - Dicionário de Dados

ObjectID 510

Relatório Dados da loja para consultas via API

	Campo	Tipo/Opções	Descrição
ControlaEstoque	bit		Se TRUE, indica que a loja tem controle de estoque dos produtos
MoedaVendas			US\$ = Dólar americano A\$ = Dólar australiano EUR = Euro ¥ = Iene £ = Libra esterlina ARG\$ = Peso argentino Mex\$ = Peso mexicano R\$ = Real NIS = Shekel
	IndexadorPrecos	real	Valor multiplicador para todos os preços

da loja

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

			categoria, linha, departamento, que a loja utiliza para agrupar os produtos da loja. Ex: categoria família, griffe, coleção, marca, seção, grupo
ExibeSubCategorias	bit		Se TRUE, indica que a loja tem cadastro de subcategorias
ExibeSubProdutos	bit		Se TRUE, indica que a loja tem cadastro de subprodutos
ExibeCoresProdutos	bit		Se TRUE, indica que a loja cadastrava cores nos produtos
NomeAdicional1	varchar(15)		Nome de campo adicional para produtos (sempre no singular). Descritor especial 1.
NomeAdicional2	varchar(15)		Nome de campo adicional para produtos (sempre no singular). Descritor especial 2.
NomeAdicional3	varchar(15)		Nome de campo adicional para produtos (sempre no singular). Descritor especial 3.

Sobre a API			Nome de campo adicional para produtos (sempre no singular). Descritor simples 1.
1 - Acessando a API			
2 - Histórico de Versões			Nome de campo adicional para produtos (sempre no singular). Descritor simples 2.
3 - Loja para testes			
4 - Usuário para testes	NomeAdicionalD1	varchar(15)	Nome de campo adicional para produtos (sempre no singular). Descritor simples 3.
5 - Melhores Práticas			
6 - Métodos da API			
7 - Permissões de acesso			
8 - Exemplos de XML	NomeAdicionalD2	varchar(15)	Nome de campo adicional para produtos (sempre no singular). Descritor simples 4.
9 - Realizando Testes			
10 - Exemplos			
11 - Dicionário de Dados	UnidadePeso	g kg lb mm cm m in ft yd	Unidade de peso dos produtos
ObjectId 510			
ObjectId 425			
ObjectId 429			
ObjectId 426	UnidadeTamanho	m in ft yd	Unidade de tamanho dos produtos
ObjectId 427			
ObjectId 248			
ObjectId 393			
ObjectId 530			
12 - Tratamento de Erros			ObjectId 425
			Relatório Lista de produtos para alterações via API

Campo	Tipo/Opções	Descrição
	I=Inclusão	
Comando	A=Alteração E=Exclusão	Indica a ação a ser realizada
NomeCat	varchar(25)	Nome da categoria
IDCategoria	int	ID interno da categoria cadastrada. Ao incluir um

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

	IDProduto	int	produto é obrigatório o campo NomeCat
	CodProd	varchar(15)	ID único que representa o produto internamente no banco de dados
	CodBarrasProd	varchar(25)	Código de referência do produto, exibido ao visitante
	NomeProd	varchar(100)	Código de barras do produto
	Peso	real	Nome do produto
			Peso do produto. A unidade (gramas ou quilogramas) é definida na página Dados da loja. É possível consultar a unidade de peso pelo relatório "Dados da loja para consultas via API" [ObjectId 510]
objectId 510	Descricao	varchar(200)	Descrição curta do produto
objectId 425	DescrLonga	varchar(1024)	Descrição longa do produto
objectId 429			Indica o nome do arquivo HTML contendo descrição adicional do produto. O conteúdo do arquivo HTM será incluído na posição da tag especial <DescrHTML>, caso esta tag exista no arquivo personalizado de exibição do layout de produtos na loja (EstiloProduto.htm)
objectId 426	DescrHTM	varchar(25)	
objectId 393			tag especial <DescrHTML>, caso esta tag exista no arquivo personalizado de exibição do layout de produtos na loja (EstiloProduto.htm)
objectId 530			
12 - Tratamento de Erros	DescrURL	varchar(100)	URL para mais detalhes do produto
	MetaKeywordsProd	varchar(200)	Palavras-chave (keywords) específicas para o produto. Os termos devem ser separados por vírgula.

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

TitleProd

varchar(100)

Exemplo: nokia,celular com wi-fi,mp3,rede 3G,GPSSão inseridas em "meta tag keyword", no código fonte da loja.

Informe neste campo o conteúdo da tag <title> que será inserido no código fonte da página de detalhe do produto. O objetivo principal é melhorar o posicionamento (ranking) nas pesquisas feitas no Google e demais sites de busca (SEO).

MetaDescriptionProd

varchar(320)

Informe neste campo o conteúdo da tag <meta description> que será inserido no código fonte da página de detalhe do produto. O objetivo principal é melhorar o posicionamento (ranking) nas pesquisas feitas no Google e demais sites de busca (SEO).

URLProd

varchar(150)

Informe neste campo a parte final da URL personalizada da página de detalhes do produto, para otimização do seu posicionamento no Google e demais sites de busca (SEO). Este termo será exibido após a barra / no endereço da página. Por exemplo, ao preencher este campo com porta-lapis, a URL personalizada deste

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectId 510

 ObjectId 425

 ObjectId 429

 ObjectId 426

ObjectId 427

 ObjectId 248

 ObjectId 393

 ObjectId 530

12 - Tratamento de Erros

			produto será www.minhaloja.com.br/porta- lapis. Utilize somente caracteres alfanuméricos, sem acentos ou espaços, que devem ser substituídos por hifens. Este campo será utilizado somente se a opção URLs personalizadas da página Dados da loja estiver marcada. Execute o utilitário Define URLs personalizadas de produtos para preencher em todos os produtos.
11 - Dicionário de Dados	URLTarget	bit	Se TRUE, URL do campo DescrURL Será exibida em uma nova janela
ObjectId 510	Estoque	smallint	Quantidade de itens do produto em estoque
ObjectId 425			Quantidade mínima de itens do produto em estoque.
ObjectId 429			Quando estoque ficar menor, será enviado e-mail para o lojista. Se 0, não tem mínimo.
ObjectId 426	EstoqueMinimo	smallint	Se TRUE, produto está disponível no site
ObjectId 427			Percentual de ICMS na origem do Produto, para geração de nota fiscal
ObjectId 248	Disponivel	bit	Custo do produto, nunca exibido ao visitante, utilizado para cálculo nos relatórios de lucratividade.
ObjectId 393	ICMS	real	Preço de venda varejo (B2C)
ObjectId 530			
12 - Tratamento de Erros	Custo	money	
	Preco	money	

Sobre a API	PrecoProm	money	Preço promocional no varejo (B2C)
1 - Acessando a API	PrecoB2B	money	Preço de venda atacado (B2B)
2 - Histórico de Versões	PrecoB2BProm	money	Preço promocional no atacado (B2B)
3 - Loja para testes	DataPromInicio	smalldatetime	Data de inicio da promoção, pode-se informar a hora. Ex: 10/05/2002 10:30
4 - Usuário para testes	DataPromFim	smalldatetime	Data de término da promoção, pode-se informar a hora. Ex: 10/05/2002 10:30
5 - Melhores Práticas			
6 - Métodos da API			
7 - Permissões de acesso			
8 - Exemplos de XML			
9 - Realizando Testes			
10 - Exemplos			
11 - Dicionário de Dados			
ObjectID 510	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.
ObjectID 425			
ObjectID 429			
ObjectID 426			
ObjectID 427			
ObjectID 248	MaxParcelasProd	tinyint	Número máximo de parcelas para pedidos que incluam este produto.
ObjectID 393			
ObjectID 530			
12 - Tratamento de Erros	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.

Sobre a API	Lancamento	bit	Se TRUE, indica que o produto é lançamento
1 - Acessando a API	EmDestaque	bit	Se TRUE, indica que o produto tem destaque no layout da loja. Ver tag <prod>
2 - Histórico de Versões			
3 - Loja para testes			
4 - Usuário para testes	ImagenProd	varchar(200)	Nome do arquivo com a imagem principal do produto com extensão JPG, PNG, GIF ou SWF
5 - Melhores Práticas			
6 - Métodos da API			
7 - Permissões de acesso	ImagenDet	varchar(200)	Nome do arquivo com a imagem de detalhe do produto com extensão JPG, PNG, GIF ou SWF
8 - Exemplos de XML			
9 - Realizando Testes			
10 - Exemplos			
11 - Dicionário de Dados	ImagenAmp	varchar(200)	Nome do arquivo com a imagem ampliada do produto com extensão JPG, PNG, GIF ou SWF
ObjectId 510			
ObjectId 425			ID para informação adicional sobre o produto, indicado em Descritor especial 1. Se multivvalorado colocar entre vírgulas. A lista de Ids pode ser consultada no relatório "Lista de descritores especiais de produtos"
ObjectId 429			[ObjectId 393]
ObjectId 426			
ObjectId 427	Adicional1	varchar(1024)	ID para informação adicional sobre o produto, indicado em Descritor especial 2. Se multivvalorado colocar entre vírgulas. A lista de Ids pode ser consultada no relatório "Lista de descritores especiais de produtos"
ObjectId 248			[ObjectId 393]
ObjectId 393			
ObjectId 530			
12 - Tratamento de Erros	Adicional2	varchar(1024)	

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectId 510

 ObjectId 425

 ObjectId 429

 ObjectId 426

ObjectId 427

 ObjectId 248

 ObjectId 393

 ObjectId 530

12 - Tratamento de Erros

AdicionalID1

AdicionalID2

AdicionalID3

IDsFiltroItem

Cores

OrdemProd

varchar(2048)

varchar(2048)

varchar(2048)

varchar(512)

varchar(1024)

tinyint

ID para informação adicional sobre o produto, indicado em Descritor especial 3. Se multivvalorado colocar entre vírgulas. A lista de Ids pode ser consultada no relatório "Lista de descritores especiais de produtos" [Object ID 393]

Texto da informação adicional sobre o produto, indicado em Descritor simples 1. Se multivvalorado, colocar entre vírgulas.

Texto da informação adicional sobre o produto,

indicado em Descritor simples 2. Se multivvalorado, colocar entre vírgulas.

Texto da informação adicional sobre o produto,

indicado em Descritor simples 3. Se multivvalorado, colocar entre vírgulas.

ID do item do filtro utilizado neste produto. Se

multivvalorado, colocar entre vírgulas (até 40). A lista de "ID do Item" pode ser consultada no relatório "Lista de filtros de produtos" [Object ID 530]

ID das cores disponíveis no produto. Se multivvalorado, colocar entre vírgulas.

Indica a ordem de exibição do produto. Ordenação

Sobre a API	IDProdutoPai	int	decrecente.
1 - Acessando a API			Se preenchido, indica que este é um sub-produto e qual o ID do produto Pai
2 - Histórico de Versões	RamoProd	tinyint	Indica o departamento no Shopping MilhoAzul.com.br
3 - Loja para testes			Indica termos de busca para produtos relacionados a estes entre vírgula. (cross-selling)
4 - Usuário para testes	MaisProd	varchar(50)	
5 - Melhores Práticas			
6 - Métodos da API	VendaConjunta	varchar(50)	
7 - Permissões de acesso			
8 - Exemplos de XML			
9 - Realizando Testes			
10 - Exemplos	DiasReposicao	smallint	
11 - Dicionário de Dados			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.
ObjectId 510			
ObjectId 425			
ObjectId 429			
ObjectId 426	DataVencimento	smalldatetime	Indica a data de vencimento do produto, Quando será enviado e-mail de alerta ao lojista
ObjectId 427			
ObjectId 248			
ObjectId 393	DiasAvisoVencimento	smallint	Números de dias antes do vencimento do produto, quando será enviado o primeiro e-mail de alerta para o lojista
ObjectId 530			
12 - Tratamento de Erros	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

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

			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.
IsProdutoGrande	bit		Indica se o produto é considerado grande para cálculo de frete, usando peso cúbico e tabela para frete grande.
LarguraProd	real		Largura do produto. A unidade (mm, cm, m, etc) é definida na página Dados da loja. É possível consultar a unidade de tamanho pelo relatório "Dados da loja para consultas via API" [ObjectId 510]
AlturaProd	real		Altura do produto. A unidade (mm, cm, m, etc) é definida na página Dados da loja. É possível consultar a unidade de tamanho pelo relatório "Dados da loja para consultas via API" [ObjectId 510]
ProfundidadeProd	real		Profundidade do produto. A unidade (mm, cm, m, etc) é definida na página Dados da loja. É possível consultar a unidade de tamanho pelo relatório "Dados da loja para

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

	CodVideo	varchar(30)	consultas via API" [ObjectId 510]
			Código do vídeo no site de vídeos como por exemplo o YouTube. Ex: qtUVA0VYXXk
	ProdMPN	varchar(50)	Código do fabricante (Manufacturer Part Number)
			Data/hora da última alteração do produto (exceto o cancelamento e descancelamento de pedidos)
	DataProdAlteracao	smalldatetime	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	IDCatAdicional1	int	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	IDCatAdicional2	int	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	IDCatAdicional3	int	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	IDCatAdicional4	int	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	IDCatAdicional5	int	ID interno da categoria cadastrada. Para informar categoria adicional do produto.
	ChangeFlagProdAPI	tinyint	Utilizado para marcar os produtos que já foram recebidos pelo ERP, evitando que sejam

novamente trazidos. Veja
mais detalhes [Aqui](#)

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

- ObjectId 248
- ObjectId 393
- ObjectId 530
- 12 - Tratamento de Erros

ObjectID 429

Relatório Categorias e IDs para cadastro de produtos via API

	Campo	Tipo/Opções	Descrição
	IDCategoria	int	ID interno da categoria no banco de dados
	Categoria	varchar(25)	Nome da categoria
	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

ObjectID 426

Relatório Lista de clientes para consultas via API

	Campo	Tipo/Opções	Descrição
	Cadastro em	smalldatetime	Indica a data de cadastramento do cliente
	Última alteração	smalldatetime	Data/hora da última alteração 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
Sobre a API	Ramo	varchar(30)	Profissão do cliente ou ramo de atuação da empresa
1 - Acessando a API	% Desc	real	Percentual de desconto global nos produtos para o cliente.
2 - Histórico de Versões			
3 - Loja para testes	Endereço	varchar(70)	Endereço completo (Logradouro + número + complemento)
4 - Usuário para testes	Logradouro	varchar(70)	Endereço do cliente
5 - Melhores Práticas	Endereço número	varchar(70)	Número do endereço
6 - Métodos da API	Endereço complemento	varchar(70)	Complemento do endereço
7 - Permissões de acesso	Bairro	varchar(25)	Bairro
8 - Exemplos de XML	Cidade	varchar(20)	Cidade
9 - Realizando Testes	Estado	char(2)	Estado
10 - Exemplos	País	varchar(20)	País
11 - Dicionário de Dados	CEP	varchar(9)	CEP
ObjectId 510	Telefone	varchar(20)	Telefone
ObjectId 425			
ObjectId 429	Celular	varchar(20)	Telefone móvel completo para contato e envio SMS (com DDD)
ObjectId 426	FAX	varchar(20)	FAX
ObjectId 427	Obs curta	varchar(10)	Texto de apoio sobre o cliente para uso interno da loja
ObjectId 248			
ObjectId 393	Pagtos	varchar(80)	Indica as formas de pagamento que este cliente pode utilizar quando a loja tem formas de pagamento por cliente
ObjectId 530			
12 - Tratamento de Erros	Sexo	Opções: F ou M	Gênero do cliente (M=masculino F=feminino)
	Data Nasc	smalldatetime	Data de nascimento
	Campo 1	varchar(100)	Campo adicional 1

Campo 2	varchar(100)	Campo adicional 2
Campo 3	varchar(100)	Campo adicional 3

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

ObjectID 427

Relatório **Lista de pedidos e detalhes para consultas via API**

	Campo	Tipo/Opções	Descrição
	Núm	int	Número do pedido
7 - Permissões de acesso	Feito em	smalldatetime	Data em que o pedido foi feito
	Nome	varchar(50)	Nome do cliente
	E-mail	varchar(50)	E-mail do cliente
10 - Exemplos	Tipo	Opções: PF ou PJ	Indica se o cliente é pessoa física ou jurídica
11 - Dicionário de Dados			
ObjectId 510			Nome da empresa, somente quando B2B
ObjectId 425	Empresa	varchar(50)	
ObjectId 429			
ObjectId 426	CPF/CNPJ	varchar(14)	CPF ou CNPJ
ObjectId 427	RG/IE	varchar(20)	Inscrição estadual ou RG
ObjectId 248			
ObjectId 393	Endereço	varchar(70)	Endereço completo do cliente
ObjectId 530	Logradouro	varchar(70)	Endereço do cliente
12 - Tratamento de Erros	Endereço número	varchar(70)	Número do endereço
	Endereço complemento	varchar(70)	Complemento do endereço
	Bairro	varchar(25)	Bairro
	Cidade	varchar(20)	Cidade do cliente
	Estado	char(2)	Estado do cliente
	País	varchar(20)	País do cliente
	CEP	varchar(9)	CEP do cliente

Sobre a API**1 - Acessando a API****2 - Histórico de Versões****3 - Loja para testes****4 - Usuário para testes****5 - Melhores Práticas****6 - Métodos da API****7 - Permissões de acesso****8 - Exemplos de XML****9 - Realizando Testes****10 - Exemplos****11 - Dicionário de Dados**

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Telefone	varchar(20)	Telefone do cliente
Celular	varchar(20)	Celular do cliente
FAX	varchar(20)	FAX
Nascido em	smalldatetime	Data de nascimento
		Texto de apoio sobre
Obs cliente	varchar(10)	o cliente, para uso interno da loja
Campo 1	varchar(100)	Campo adicional 1
Campo 2	varchar(100)	Campo adicional 2
Campo 3	varchar(100)	Campo adicional 3
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 completo de entrega
Logradouro Pedido	varchar(70)	Endereço de entrega
Endereço número Pedido	varchar(70)	Número do endereço de entrega
Endereço complemento Pedido	varchar(70)	Complemento do 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
Campo 1 Pedido	varchar(100)	Campo adicional no pedido 1
Campo 2 Pedido	varchar(100)	Campo adicional no pedido 2

Sobre a API	Campo 3 Pedido	varchar(100)	Campo adicional no pedido 3
1 - Acessando a API	Total sem frete	money	Valor total do pedido, SEM o frete
2 - Histórico de Versões	Frete	money	Valor do frete
3 - Loja para testes			Opção de pagamento escolhida pelo cliente
4 - Usuário para testes	Pagamento	varchar(50)	
5 - Melhores Práticas			
6 - Métodos da API		Opções:	
7 - Permissões de acesso		AMEX	
8 - Exemplos de XML		Diners	
9 - Realizando Testes	Cartão	Mastercard	Bandeira de cartão de crédito
10 - Exemplos		VISA	
11 - Dicionário de Dados		Hipercard	
ObjectID 510		Aura	
ObjectID 425		Elo	
ObjectID 429			
ObjectID 426			
ObjectID 427		Opções:	
ObjectID 248		Outras	
ObjectID 393	Tecnologia Cartão	Cielo	
ObjectID 530		Redecard	
12 - Tratamento de Erros		CobreBem	
		Wirecard	
		PagSeguro	Indica a tecnologia utilizada para
		MercadoPago	
		Stone	processar a
		Stelo	transação com
		PayU	cartão
		Pagar.me	
		Getnet	
		Yapay	
		Juno	
		iugu	
		Não definida	
	Tecnologia Boleto	Opções:	Indica a tecnologia utilizada para
		Bradesco	
		Itaú	processar a

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Genérico transação com

Real boleto

Banco do

Brasil

Stelo

PayU

PagSeguro

Wirecard

MercadoPago

Pagar.me

Getnet

Yapay

Juno

iugu

Não definida

Opções:
Genérico
Não definida

Indica a tecnologia utilizada para processar a transação com Pix

Tecnologia Pix

ccNum varchar(19)

Número do cartão de crédito

ccNome

varchar(30)

Nome gravado no cartão de crédito

ccSeg

varchar(4)

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

Sobre a API	ccNumAutent	varchar(27)	Número de autenticação da venda, emitido pela administradora do cartão
1 - Acessando a API			
2 - Histórico de Versões			Retorno do KomerCi
3 - Loja para testes	ccNumPrg	varchar(1)	SecureCode
4 - Usuário para testes			(0,1,2,3,4)
5 - Melhores Práticas			Código de retorno do processo de autorização da venda, informado pela administradora do cartão
6 - Métodos da API			
7 - Permissões de acesso	ccCodRet	varchar(2)	
8 - Exemplos de XML			
9 - Realizando Testes			
10 - Exemplos	ccORIGEM_BIN	char(3)	Cód país emissor KomerCi
11 - Dicionário de Dados			Endereço reportado pela Redecard via AVS
ObjectId 510	ccEndereco	varchar(100)	
ObjectId 425			PaymentId do pagamento do pedido na Cielo API
ObjectId 429	CieloAPIPaymentId	varchar(36)	
ObjectId 426			Status do pagamento do pedido na Cielo API
ObjectId 427			Identificação da transação na Cielo
ObjectId 248	CieloAPIStatus	varchar(2)	
ObjectId 393			Cód de retorno da Cielo
ObjectId 530	VisaTID	char(20)	Cód aprovação bancária informado pela Cielo, somente para pedidos autorizados.
12 - Tratamento de Erros	VisaLR	varchar(4)	
	VisaARP	varchar(6)	
	VisaCOD	varchar(4)	Código do resultado da captura do pedido

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

			informado pela Cielo
			Indica se o pedido foi autenticado pelo banco emissor: 0 ou vazio=transação sem autenticação 1=ok 2=negado 3=falha
VisaAuth		varchar(3)	
			Valor capturado informado pela Cielo
VisaCAP		money	
			Valor cancelado informado pela Cielo
CieloCAN		money	
			Código do banco emissor do cartão
VisaBank		varchar(4)	
			Número identificador único da transação para e.Rede
eRedeTID		varchar(20)	
			Status da transação na e.Rede
eRedeStatus		varchar(20)	
			Código de retorno da transação
eRedeCode		varchar(3)	returnCode
			retornado pela e.Rede
ObjectID 427			
			Número da autorização da transação retornada pelo emissor do cartão para e.Rede
ObjectID 248			
ObjectID 393			
ObjectID 530	eRedeAuthCode	varchar(6)	
			Código do resultado da captura do pedido informado pela e.Rede
12 - Tratamento de Erros	eRedeCOD	varchar(12)	
			Valor capturado informado pela e.Rede
	eRedeCAP	money	

Sobre a API			Código da transação retornada pelo Moip/Wirecard
1 - Acessando a API	MoipCodigo	varchar(12)	Status da transação retornada pelo Moip/Wirecard. Ex: 5
2 - Histórico de Versões			Os valores possíveis para o campo MoipStatus:
3 - Loja para testes			(1) Autorizado: Pagamento autorizado.
4 - Usuário para testes			(2) Iniciado: Pagamento foi iniciado, porém sem confirmação de finalização até o momento
5 - Melhores Práticas			(3) BoletolImpresso: Boleto visualizado pelo cliente
6 - Métodos da API			(4) Concluído: Pagamento creditado em conta e disponível para saque.
7 - Permissões de acesso			(5) Cancelado: Pagamento foi cancelado
8 - Exemplos de XML			(6) EmAnalise: Pagamento em análise de risco
9 - Realizando Testes			(7) Estornado: Pagamento foi estornado
10 - Exemplos			(8) EmRevisao: Pagamento em revisão pelo Moip
11 - Dicionário de Dados			
ObjectId 510			
ObjectId 425			
ObjectId 429			
ObjectId 426			
ObjectId 427			
ObjectId 248			
ObjectId 393			
ObjectId 530			
12 - Tratamento de Erros			

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectId 510

ObjectId 425

ObjectId 429

ObjectId 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

PagSeguroID

varchar(36)

Código da transação

gerada pelo

PagSeguro

Status da transação

retornada pelo

PagSeguro:

(1) Aguardando
pagamento

(2) Em análise

(3) Paga

(4) Disponível

(5) Em disputa

(6) Devolvida

(7) Cancelada

PagSeguroStatus

tinyint

(2) Em análise

(3) Paga

(4) Disponível

(5) Em disputa

(6) Devolvida

(7) Cancelada

MercadoPagoid

varchar(50)

Código da transação

gerada pelo

MercadoPago

Status da transação

retornada pelo

MercadoPagoStatus

varchar(50)

MercadoPago. Ex:

approved, cancelled,
etc

Situação da

transação na Stone:

0=sem transação

1=transação

aprovada sem

captura

2=capturado

3=cancelado

StoneTransactSit

tinyint

Identificação da

transação definida

pela Stone. Usada

na consulta, captura

StoneRcptTxId

varchar(14)

(9) Reembolsado:

Pagamento foi
reembolsado

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

 ObjectId 510

 ObjectId 425

 ObjectId 429

 ObjectId 426

ObjectId 427

 ObjectId 248

 ObjectId 393

 ObjectId 530

12 - Tratamento de Erros

			e cancelamento da transação
	CieloOrderNumber	varchar(32)	Código da transação gerada pelo Cielo Checkout
	CieloCheckoutStatus	tinyint	Status da transação retornada pelo Checkout Cielo (1=Pendente, 2=Pago, 3=Negado, 5=Cancelado, 6=Não Finalizado, 7=Autorizado)
	PayUOrderId	varchar(36)	O identificador do pedido gerado pela PayU
	PayUStatus	varchar(20)	Ex: APPROVED, PENDING, DECLINED, etc
	PagarMeld	varchar(20)	Id da transação no Pagar.me
	PagarMeStatus	varchar(20)	Status da transação no Pagar.me
	GetnetPaymentId	varchar(36)	Código de identificação do pagamento na Getnet
	GetnetStatus	varchar(20)	Status do pagamento na Getnet
	GetnetAuthCode	varchar(20)	Código da autorização de crédito na Getnet

Sobre a API			Código de transação do adquirente Getnet
1 - Acessando a API	GetnetTransactionId	varchar(20)	transaction_id do pedido retornado pela Yapay
2 - Histórico de Versões	YapayTransactionId	varchar(30)	pedido retornado pela Yapay
3 - Loja para testes	YapayStatus	varchar(20)	Status da transação na Yapay
4 - Usuário para testes	JunoChargeCode	varchar(50)	Code na Juno
5 - Melhores Práticas	JunoPaymentId	varchar(50)	PaymentId na Juno
6 - Métodos da API	JunoPaymentStatus	varchar(30)	Status da transação na Juno
7 - Permissões de acesso	IuguInvoiceId	varchar(100)	ID da fatura na iugu
8 - Exemplos de XML	IuguStatus	varchar(30)	Status na iugu
9 - Realizando Testes	Meli_order_id	varchar(30)	id do pedido no MercadoLivre
10 - Exemplos			Status do pedido no MercadoLivre:
11 - Dicionário de Dados			confirmed, payment_required, payment_in_process, paid, cancelled, invalid
ObjectId 510			
ObjectId 425	Meli_status	varchar(30)	
ObjectId 429			
ObjectId 426			
ObjectId 427			
ObjectId 248	SkyHubOrderCode	varchar(100)	Número do pedido retornado pela SkyHub
ObjectId 393			
ObjectId 530	SkyHubOrderStatusLabel	varchar(50)	Descrição do status na SkyHub
12 - Tratamento de Erros	Parcelas	tinyint	Número de parcelas
	Juros a.m.	real	Juros ao mês cobrados no parcelamento do pedido
	Presente	Opções: S ou N	Indica se pedido é para presente

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

			Opções:
	Status	aprovado	Status do pedido
		Pendente	
		Aprovado	
		Liberado	
		Remetido	
			Texto de apoio sobre o pedido, para uso interno da loja
	Obs Pedido	varchar(10)	
	Entregue em	smalldatetime	Data de entrega do pedido ao destinatário pela transportadora
			Código fornecido pelos Correios para acompanhamento de remessas
	Obj SEDEX	varchar(13)	
	Ref. Produto	varchar(15)	Referência do produto
	Cód. Barras	varchar(25)	Código de barras do produto
	Nome Produto	varchar(100)	Nome do produto
	Categoria	varchar(25)	Nome da categoria
	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
	NomeLocal	varchar(40)	Nome do local de envio informado pelo

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectId 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

	Parceiro	varchar(30)	cliente (ex: casa da mãe, escritório, etc)
	ThisfTransacao	varchar(14)	Quando o pedido for feito através de um parceiro, informa o nome do parceiro
	ThisfAutorizacao	varchar(6)	Código de transação com cartão de crédito recebido da Thisf, se a loja utiliza
	Esp1	varchar(30)	Código de autorização recebido pela Thisf da administradora do cartão
	Esp2	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 1
	Esp3	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 2
	Simp1	varchar(30)	Informação adicional sobre o produto, indicado em Descritor especial 3
	Simp2	varchar(30)	Informação adicional sobre o produto, indicado em Descritor simples 1
			Informação adicional sobre o produto, indicado em Descritor simples 2

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

	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
	Embalar produto	Opções: S ou N	Indica se produto deve ser embalado
	Valor embalagem produto	smallmoney	Valor para embalagem do produto
	Valor embalagem	smallmoney	Valor para a embalagem de presente
	Cor	varchar(30)	Cor solicitada do produto
	RuleName	varchar(50)	Regra promocional do item na cesta
	Desconto cupom	real	Valor em \$ do desconto no pedido
	NumNFe	varchar(44)	Número/Chave da nota fiscal eletrônica com 44 posições.
	ChangeFlagAPI	tinyint	Utilizado para marcar os pedidos que já foram recebidos pelo ERP, evitando que sejam novamente trazidos.
	Sexo	varchar(1)	Pode ser M, F ou branco

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Custo unit	money	Preço de custo unitário do produto
Obs Frete	varchar(100)	Observação que a loja cadastrou no frete ou que retorna do sistema usado para cálculo de frete. Ex: prazo normal de entrega na faixa de CEP

ObjectID 248

Relatório **Lista de classes de produtos para cadastro via API**

Campo	Tipo/Opções	Descrição
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

ObjectID 393

Relatório **Lista de descritores especiais de produtos**

Campo	Tipo/Opções	Descrição
-------	-------------	-----------

Sobre a API

- 1 - Acessando a API
- 2 - Histórico de Versões
- 3 - Loja para testes
- 4 - Usuário para testes
- 5 - Melhores Práticas
- 6 - Métodos da API
- 7 - Permissões de acesso
- 8 - Exemplos de XML
- 9 - Realizando Testes
- 10 - Exemplos
- 11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

- 12 - Tratamento de Erros

ID Adicional1	int	Indica o ID do item do descritor.
Item	varchar(30)	O nome do campo pode ser ID Adicional1, ID Adicional2 ou ID Adicional3 dependendo de qual descritor está sendo consultado (deve ser utilizado o Par1 de filtro de relatório para indicar o descritor)
		Nome do item

ObjectID 530

Relatório Lista de filtros de produtos

Campo	Tipo/Opções	Descrição
ID do Filtro	int	Indica o ID do filtro.
Nome do filtro	varchar(30)	Nome do filtro apresentado ao visitante (ex: voltagem)
Descrição	varchar(50)	Descrição do filtro
ObjectId 426		Indica se o filtro de refere a todas as categorias da loja (Geral) ou por categoria
ObjectId 427	Opções: Geral ou Categoria	
ObjectId 248		
ObjectId 393		
ObjectId 530		
Exibe na loja	Opções: Sim ou Não	Indica se o filtro está ativo
Exibe no pedido	Opções: Não, Só item e Nome e item	Indica se o filtro é exibido na loja
Ordem Filtro	tinyint	Indica a ordem de exibição. Se 0, ordem não

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectId 510

ObjectId 425

ObjectId 429

ObjectId 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

ID do Item	int	é alterada. Ordenação pelo maior.
Nome do item	varchar(30)	Indica o ID de cada item do filtro. Estes IDs são informados no campo IDsFiltroltem do cadastro de produtos
Item Ativo	Opções: Sim ou Não	Indica se o item do filtro está ativo
Ordem Item	tinyint	Indica a ordem de exibição. Se 0, ordem não é alterada. Ordenação pelo maior.

12 - Tratamento de Erros

Se for enviado um XML com múltiplos registros e existir erro em um dos registros, a API irá processar até ocorrer o erro, retornando os registros onde a execução ocorreu sem problema e por último o registro que causou o erro.

No exemplo abaixo existem 5 registros. O erro está no registro 3. Foi definido o comando U, que é um comando inexistente. Os dois primeiros registros foram processados (Deleted="2") e os dois últimos não foram. O erro ocorreu no terceiro registro:

```
<Records>
<Record>
  <Field Name="Comando" Value="E" />
  <Field Name="IDProduto" Value="5206796" />
</Record>
<Record>
  <Field Name="Comando" Value="E" />
  <Field Name="IDProduto" Value="5206797" />
```

Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectId 510

ObjectId 425

ObjectId 429

ObjectId 426

ObjectId 427

ObjectId 248

ObjectId 393

ObjectId 530

12 - Tratamento de Erros

```
</Record>
<Record>
  <Field Name="Comando" Value="U" />
  <Field Name="IDProduto" Value="5206798" />
</Record>
<Record>
  <Field Name="Comando" Value="E" />
  <Field Name="IDProduto" Value="5206799" />
</Record>
<Record>
  <Field Name="Comando" Value="E" />
  <Field Name="IDProduto" Value="5206800" />
</Record>
</Records>
```

Retorno da API:

```
<API>
<Record Num="1">
  <Field Name="COMANDO" Value="E"/>
  <Field Name="IDPRODUTO" Value="5206796"/>
</Record>
<Record Num="2">
  <Field Name="COMANDO" Value="E"/>
  <Field Name="IDPRODUTO" Value="5206797"/>
</Record>
<Record Num="3">
  <Field Name="COMANDO" Value="U"/>
  <Field Name="IDPRODUTO" Value="5206798"/>
</Record>
<ErrCod>18</ErrCod>
<ErrDescr>Invalid Product(s) in XMLRecords</ErrDes
<IntCod>115</IntCod>
<IntDescr>Invalid command</IntDescr>
<Stats Sent="5" Valid="2" Included="0" IncludedSuł
</API>
```



Sobre a API

1 - Acessando a API

2 - Histórico de Versões

3 - Loja para testes

4 - Usuário para testes

5 - Melhores Práticas

6 - Métodos da API

7 - Permissões de acesso

8 - Exemplos de XML

9 - Realizando Testes

10 - Exemplos

11 - Dicionário de Dados

ObjectID 510

ObjectID 425

ObjectID 429

ObjectID 426

ObjectID 427

ObjectID 248

ObjectID 393

ObjectID 530

12 - Tratamento de Erros

Made by Fastcommerce Tech Support Team