

E-Gondola API - Web Service

A API do ecommerce E-Gondola utiliza um webservice com o protocolo SOAP.

A chave de acesso do modo de produção, são disponibilizados mediante solicitação via email de suporte e estará disponível no endereço abaixo:

<https://api.egondola.app/ws?wsdl>

Atenção: ao implantar a API em modo de produção, realize backup do banco de dados através do sistema administrativo no menu “Administração > Backup do Banco de Dados”. O uso dos métodos, é de inteira responsabilidade do solicitante da chave de acesso do modo de produção. O EGondola não se responsabiliza pelo mau uso dos dados.

Suporte

Dúvidas deverão ser enviadas para o email: plataforma@egondola.com.br

Descrição da API

Toda a execução das funções é feita através da chamada do método descrito abaixo:

Método de Autenticação e Execução das Funções

- EGonWebServiceAuth (array(identificacao, nome_da_funcao, parametros));

Onde:

identificação: string contendo a chave de acesso a api

nome_da_funcao: string com o nome da função a ser executada

parametros: parâmetros passados para a função no formato json

- Lista de Funções

Os parâmetros marcados com *, são obrigatórios.

Os parâmetros de filtros, podem ser usados combinados, eles são agregados com o operador lógico “and”.

A codificação é utf-8.

Funções disponíveis

- listaProdutos
- listaCategorias
- listaMarcas
- listaClientes
- listaPedidos
- getPreco
- setPrecoValor
- setPrecoPorcentagem
- setSituacaoProduto
- getEstoqueProduto
- setEstoqueProduto
- setSituacaoPedido
- getSituacaoPedido
- addProduto
- addCategoria
- addMarca

Abaixo segue a descrição de cada função:

listaProdutos

Retorna uma lista de produtos cadastrados no ecommerce conforme os parâmetros de filtro utilizados.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia	Array de strings ou string	Pode ser informado um array de strings contendo as referencias dos produtos que deseja retornar os dados, caso seja apenas um valor, informar o valor em uma string	Array('123','124','125') Ou '123'
titulo	String	Nome ou parte do nome do produto	'sapato'
codigo	Array de inteiros ou inteiro	Pode ser informado um array com os códigos dos produtos cadastrados no ecommerce ou apenas um código como inteiro	Array(1, 10, 50) ou 66
categoria	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das categorias cadastradas no ecommerce ou apenas um código como inteiro	Array(40,43) ou 43
marca	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das marcas cadastradas no	Array(21,11,10) ou 10

		ecommerce ou apenas um código como inteiro	
tipo	String	Informar 'R' = resumido (padrão) ou 'D' = detalhado	'D'
pagina	Inteiro	Registro inicial, o primeiro registro começa do 0	5
Limite	Inteiro	Numero de registros que devem ser retornados	20

Parâmetros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
dados	Array	<p>Array com os seguintes dados do produto:</p> <p>Resumido:</p> <ul style="list-style-type: none"> - titulo:string(150) - código:inteiro(11) - referencia:string(30) - categoria:string(60) - marca:string(150) - preco:float(10,5) - estoque:inteiro(11) - peso:float(10,5) - altura:inteiro(11) - largura:inteiro(11) - comprimento(11) - situação:char(1) <p>Detalhado:</p> <ul style="list-style-type: none"> - titulo:string(150) - código:inteiro(11) - referencia:string(30) - categoria_id:inteiro(11) - categoria:string(60) - categoria_caminho:string(250) - marca_id:inteiro(11) - marca:string(150) - preco:float(10,5) - desconto_dado:float(10,5) - estoque:inteiro(11) - peso:float(10,5) - altura:inteiro(11) - largura:inteiro(11) - comprimento:inteiro(11) - destaque:char(1) - lançamento:char(1) - unidade:string(10) 	

		<pre> - prazo_adicional:inteiro(6) - frete_gratis:char(1) - data_cadastro:date(Y-m-d) - descricao:text - situacao:char(1) - opcoes:array(codigo:inteiro(11) referencia:string(20) opcao1:string(250) valor1:string(250) opcao2:string(250) valor2:string(250) preco:float(10,5) estoque:inteiro(11) peso:float(10,5) cubagem:inteiro(11)) </pre>	
--	--	--	--

listaCategorias

Retorna um lista de categorias cadastradas no ecommerce conforme os parâmetros de filtro utilizados.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
titulo	String	Nome ou parte do nome da categoria	'sapatos'
codigo	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das categorias no ecommerce ou apenas um codigo	Array(1, 10, 50) ou 66
pagina	Inteiro	Registro inicial, o primeiro registro começa do 0	5
Limite	Inteiro	Numero de registros que devem ser retornados	20

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
dados	Array	Array com os seguintes dados da categoria: - titulo:string(60)	

		- codigo:inteiro(11) - categoriapai_id:inteiro(11) - categoria_caminho:string(250)	
--	--	--	--

listaMarcas

Retorna um lista de marcas cadastradas no ecommerce conforme os parâmetros de filtro utilizados.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
titulo	String	Nome ou parte do nome da marca	'ferracini'
codigo	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das marcas no ecommerce ou apenas um código como inteiro	Array(1, 10, 50) ou 66
pagina	Inteiro	Registro inicial, o primeiro registro começa do 0	5
Limite	Inteiro	Numero de registros que devem ser retornados	20

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
dados	Array	Array com os seguintes dados da categoria: - titulo:string(150) - codigo:inteiro(11)	

listaClientes

Retorna um lista de clientes cadastradas no ecommerce conforme os parâmetros de filtro utilizados.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
nome	String	Nome ou parte do nome do cliente	'carlos'
codigo	Array de inteiros	Pode ser informado um	Array(1, 10, 50) ou

	ou inteiro	array com os códigos dos clientes no ecommerce ou apenas um código como inteiro	66
pagina	Inteiro	Registro inicial, o primeiro registro começa do 0	5
Limite	Inteiro	Numero de registros que devem ser retornados	20

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
dados	Array	<p>Array com os seguintes dados da categoria:</p> <ul style="list-style-type: none"> - codigo:inteiro(11) - email:string(100) - nome:string(150) - cpfcnpj:string(16) - tipo_cadastro:string(8) - inscricao:string(15) - data_nascimento:data(Y-m-d) - sexo:char(1) - fone_residencial:string(15) - fone_comercial:string(15) - fone_celular:string(15) - nome_responsavel:string(150) - cpf_responsavel:string(11) - situacao_cadastro:char(1) - end_cobranca_endereco:string(100) - end_cobranca_numero:string(5) - end_cobranca_bairro:string(80) - - end_cobranca_complemento:string(100) - end_cobranca_cidade:string(60) - end_cobranca_estado:string(2) - end_cobranca_cep:string(9) - end_entrega_endereco:string(100) - end_entrega_numero:string(5) - end_entrega_bairro:string(80) - end_entrega_complemento:string(100) - end_entrega_cidade:string(60) - end_entrega_estado:string(2) - end_entrega_cep:string(9) - data_cadastro:data(Y-m-d) - recebe_news:char(1) 	<p>(jurídica,física)</p> <p>(M,F)</p> <p>99-999999999</p> <p>99-999999999</p> <p>99-999999999</p> <p>(1,0)</p> <p>87500-000</p> <p>87500-000</p> <p>(1,0)</p>

listaPedidos

Retorna um lista de pedidos realizados no ecommerce conforme os parâmetros de filtro utilizados.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
numero	Array de inteiros ou inteiro	Pode ser informado um array com os códigos dos pedidos no ecommerce ou apenas um código como inteiro	Array(1, 10, 50) ou 66
situacao	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das situacoes dos pedidos ou apenas um código como inteiro Situações disponíveis: 1 - Aguardando pagamento 2 – Pagamento não confirmado 3 – Pedido em análise 4 – Pedido aceito 5 - Pedido em Separação 6 - Encaminhado para Transportadora 7 - Pedido Entregue 8 – Pedido Cancelado	Array(1, 2, 4) ou 1
datainicial	data	Uma data no formato aaaa-mm-dd	2014-01-20
datafinal	data	Uma data no formato aaaa-mm-dd	2014-01-25

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Int(11)	Numero de registros retornados	10
dados	Array	Array com os seguintes dados da categoria: - codigo:inteiro(11) -id_pedido:inteiro(11) - data_pedido:data(Y-m-d H:i:s) - situacao:inteiro(1) - data_pagamento:data(Y-m-d) - valor_acrescimo:float(10,2) - valor_desconto:float(10,2) - valor_total:float(10,2) - entrega: array(valor_frete:float(10,2)	19 ^(id +mod11) 1 (1,2,3,4,5,6,7,8)

		<pre> prazo_entrega:inteiro(6) peso:float(10,5) volume:inteiro(11) transportadora:string(50)) - cliente:array(codigo:inteiro(11) email:string(100) nome:string(150) cpfcnpj:string(16) tipo_cadastro:string(8) inscricao:string(15) data_nascimento:data(Y-m-d) sexo:string(1) fone_residencial:string(15) fone_comercial:string(15) fone_celular:string(15) nome_responsavel:string(150) cpf_responsavel:string(11) end_cobranca_endereco:string(100) end_cobranca_numero:string(5) end_cobranca_bairro:string(80) end_cobranca_complemento:string(100) end_cobranca_cidade:string(60) end_cobranca_estado:string(2) end_cobranca_cep:string(9)) - forma_pgto:array(codigo:inteiro(11) forma:string(100) parcelas:inteiro(6) valor_parcela:float(10,2) dados_cartao:string(400) outros:string(200) tipo:string(30)) - end_entrega_endereco:string(100) - end_entrega_numero:string(5) - end_entrega_bairro:string(80) - end_entrega_complemento:string(100) - end_entrega_cidade:string(60) - end_entrega_estado:string(2) - end_entrega_cep:string(9) - itens:array(codigo:string(30), id_item:inteiro(11) referencia:string(30) quantidade:inteiro(11) nome:string(150) unid:string(10) opcao:string(100) </pre>	
--	--	---	--

		<pre> preco:float(10,5) peso:float(10,5) cubagem:inteiro(11)) </pre>	
--	--	---	--

getPreco

Retorna os dados do preço do produto

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia*	String	Referencia do produto	'123'

Parâmetros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
dados	Array	Array com os seguintes dados da categoria: - codigo_completo:string(30) (código completo do produto no sistema) - referencia_principal:string(30) (referencia do produto principal) - referencia_opcao:string(30) (referencia da opção, quando houver) - preco:float(10,2) - desconto_dado:float(10,2) (esse valor já esta descontado do preco)	

setPrecoValor

Altera o preco do produto utilizando um valor direto, pode ser setado diretamente o preço de uma opção de produto utilizando sua referencia no ecommerce

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia*	String	Referencia do produto	'123'
tipo	String	'V' = valor direto (padrão), 'D' = usa o valor como	'V'

		desconto no campo desconto	
valor*	Float	Valor a ser utilizado	125.20
zera_desconto	String	'0' = não zera o campo desconto (padrão), '1' = zera o campo desconto	'1'
inlui_opcoes	String	1 = altera o valor de todas as opções quando a referencia é de um produto principal com opcoes, 0 = mantém os valores das opções (padrão)	'1'

Parâmetros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_afetados	Inteiro(11)	Retorna o número de registros afetados quando o resultado é 1	10

setPrecoPorcentagem

Altera os valores dos produtos utilizando um valor de porcentagem. Caso o valor seja positivo, será dado acréscimo aos produtos, caso seja negativo, será dado desconto nos produtos que foram filtrados utilizando os parâmetros de filtros.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia	String	Pode ser informado um array de strings contendo as referencias dos produtos que deseja alterar os preços, caso seja apenas um valor, informar o valor em uma string	Array('123','124','125') ou '123'
categoria	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das categorias cadastradas no ecommerce ou apenas um código como inteiro	Array(40,43) ou 43
marca	Array de inteiros ou inteiro	Pode ser informado um array com os códigos das marcas cadastradas no ecommerce ou apenas um código como inteiro	Array(21,11,10) ou 10
percent*	Float	Percentual para alteração	-5

		dos valores. Para acréscimo dos valores, informe um valor maior que zero, para desconto nos produtos, utilize um valor menor que zero	
depor	String	se igual a '1', no caso de desconto, o valor é alterado no campo desconto, '0' = mantem os valores (padrão)	'1'
incli_opcoes	String	1 = altera o valor de todas as opções quando a referencia é de um produto principal com opcoes, 0 = mantem os valores das opções (padrão)	'1'

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_afetados	Inteiro(11)	Retorna o número de registros afetados quando o resultado é 1	10

setSituacaoProduto

Altera a situação do produto principal, não é possível desativar apenas uma opção de produto, caso seja utilizado uma referencia de um opção, todo as opções desse produto serão desativadas.

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia*	String	Referencia do produto	'123'
situacao*	String	'1' = ativo, '0' = inativo	'1'

Exemplo de código enviado:

```
{"referencia":"55816","situacao":"0"}
```

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de	sucesso

		retorno	
total_afetados	Inteiro(11)	Retorna o número de registros afetados quando o resultado é 1	10

Exemplo de código retornado:

```
{"resultado": "1", "msg": "sucesso", "total_afetados": 1}
```

getEstoqueProduto

Retorna a quantidade em estoque do produto

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia*	String	Referencia do produto ou dá opção	'123'

Exemplo de código json enviado:

```
{"referencia": "55816"}
```

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	10
total_estoque	Inteiro(11)	Quantidade total do estoque do produto (soma das opções)	55
dados	Array	Array com os seguintes dados da categoria: - codigo_completo:string(30) (código completo do produto no sistema) - referencia_principal:string(30) (referencia do produto principal) - referencia_opcao:string(30) (referencia da opção, quando houver) - estoque:inteiro(11)	

Exemplo de código retornado:

```
{"resultado": "1", "msg": "sucesso", "total_registros": 1, "total_estoque": 4, "dados": [{"codigo_completo": "21.21", "referencia_principal": "55816", "referencia_opcao": "55816", "estoque": "4", "reserva": "0"}]}
```

setEstoqueProduto

Altera o estoque do produto ou de uma opção

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
referencia*	String	Referencia do produto ou dá opção	'123'
estoque	Float	Quantidade a ser utilizada na ação de movimentação do estoque	15
tipo	String	'V' = valor direto (padrão), 'S' = incrementa o valor ao estoque, 'D' = abate o valor do estoque	'S'

Exemplo de código json enviado:

```
{"referencia":"55816","estoque":"4"}
```

Parametros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_afetados	Inteiro(11)	Retorna o número de registros afetados quando o resultado é 1	10

Exemplo de código json retornado:

```
{"resultado":"1","msg":"sucesso","total_afetados":1}
```

setSituacaoPedido

Altera a situação do Pedido

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
codigo*	Inteiro	Código do pedido	123
nova_situacao	Inteiro	Código da nova situação: Situações disponíveis: 1 - Aguardando pagamento 2 – Pagamento não	4

		confirmado 3 – Pedido em análise 4 – Pedido aceito 5 - Pedido em Separação 6 - Encaminhado para Transportadora 7 - Pedido Entregue 8 – Pedido Cancelado	
envia_email	String(1)	'1' = Enviar, '0' = Não Enviar	'1'
data_pgto (* obrigatório quando a nova situação for igual a 3)	data	Uma data no formato aaaa-mm-dd	2014-01-20

Exemplo de código enviado:

```
{"codigo":"167","nova_situacao":"2"}
```

Parâmetros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_afetados	Inteiro(11)	Retorna o número de registros afetados quando o resultado é 1	10

Exemplo de código retornado:

```
{"resultado":"1","msg":"sucesso","total_afetados":1}
```

getSituacaoPedido

Altera a situação do Pedido

Parâmetros de entrada:

Nome	Tipo	Descrição	Exemplo
codigo*	Inteiro	Código do pedido	123

Exemplo de código enviado:

```
{"codigo":"167"}
```

Parâmetros de saída:

Os parâmetros de saída são no formato json onde:

Nome	Tipo	Descrição	Exemplo
------	------	-----------	---------

resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
total_registros	Inteiro(11)	Numero de registros retornados	1
dados	Array	Array com os seguintes dados: - situacao:inteiro(1) (código da situação atual) - data_situacao:datetime(yyyy-mm-dd hh:mm:ss) (data hora da situação) - texto_situacao:string(50) (titulo da situação) - possivel_nova_situacao:inteiro(1) (array com os códigos da possível nova situação)	

Exemplo de código retornado:

```
{ "resultado": "1", "msg": "sucesso", "total_registros": 1, "dados": { "situacao": "2", "data_situacao": "2017-04-07 17:33:53", "texto_situacao": "Pagamento n\u00e3o confirmado", "possivel_nova_situacao": [ "1", "2", "8" ] } }
```

addProduto

Adiciona novos produtos ou atualiza os já cadastrados caso o campo “ref” já esteja cadastro no sistema.

Parâmetros de entrada:

ATENÇÃO: A entrada consiste em um campo “dados” do tipo array, com os dados de um ou múltiplos registros de produtos.

Os campos do array “dados” seguem abaixo:

Nome	Tipo	Descrição	Exemplo
ref*	String(30)	Referencia do produto	'123'
titulo*	String(150)	Titulo do produto	Tênis zoom
codebar	inteiro(25)	Codigo de barras do produto	7898541000215
estoque*	Inteiro(11)	Estoque inicial do produto	10
limite_estoque*	Inteiro(11)	Limite para aviso de reposição do estoque	1
preco*	Float(10,2)	Preço do produto	55.10
desconto	Float(10,2)	Valor do desconto	10.00
descontoavista	Float(10,2)	Valor do desconto extra caso o produto seja vendido a vista	15.00
destaque	String(1)	1 para marcar o produto como destaque, 0 para não	1
lançamento	String(1)	1 para marcar o produto como lançamento, 0 para não	1

oferta	String(1)	1 para marcar o produto com oferta, 0 para não	1
peso*	Float(10,5)	Peso do produto em kg	0.500
Altura*	Inteiro(11)	Altura do produto em cm	20
largura*	Inteiro(11)	Largura do produto em cm	15
comprimento*	Inteiro(11)	Comprimento do produto em cm	40
unidade*	String(10)	Unidade de medida do produto	Pç
cod_fornecedor*	Inteiro(11)	Código do fornecedor previamente cadastrado no sistema	15
prazo_entrega_adicional	Inteiro(6)	Prazo de entrega adicional ao calculado pela transportadora	3
tags	String(250)	Tags do produto separados por virgula	Tênis, Nike,street
descricao	Text	Texto de descrição do produto que pode conter códigos html	Tênis street
titulo_agrupador	String(250)	Titulo ou código em comum entre os produtos para que eles seja agrupados na pagina de venda	tenisnike
variacao	String(80)	Titulo e valor da variação agrupadas pelo titulo_agrupador . O titulo e o valor devem ser separados por “:”	Cor:Preto
fretegratis	String(1)	1 para marcar o produto como frete grátis, e 0 para não	1
situacao*	String(1)	1 para marcar o produto como ativo, e 0 para deixá-lo inativo	1
categorias*	Array de inteiros(11) ou inteiro(11)	Pode ser informado um array com os códigos das categorias cadastradas no ecommerce ou apenas um código como inteiro	Array(40,43) ou 43
produtos_relacionados	Array de inteiros(11) ou inteiro(11)	Pode ser informado um array com os códigos(códigos gerados pelo ecommerce e não as referencias informadas) dos produtos já cadastradas no ecommerce ou apenas um código como inteiro	Array(228,357) ou 455
imagens	Array de	Pode ser informado um	Array('http://www.s

	strings(400) ou string(400)	array com urls de imagens ou uma url apenas. Utilize imagens no formato jpg	ite.com/img.jpg','http://www.site.com/img2.jpg') ou 'http://www.site.com.br/img.jpg'
grade Obs: as referencias já cadastradas no sistema, serão atualizadas	Array	<p>Array com os seguintes dados da grade:</p> <ul style="list-style-type: none"> - ref:String(30)* (referencia da variação) - preco:Float(10,2) (caso o preço seja o mesmo da capa do produto, informar "0") - peso:Float (10,5)(caso o peso seja o mesmo da capa do produto, informar "0") - altura:Integer(11) (caso a altura seja a mesma da capa do produto, informar "0") - largura:Integer(11) (caso a largura seja a mesma da capa do produto, informar "0") - comprimento:Integer(11) (caso o comprimento seja o mesmo da capa do produto, informar "0") - estoque:Integer(11)* - limite_estoque:Integer(11)* - opcao_titulo_1:String(250) * (informar o titulo da variação 1, ex: Cor, Tamanho, Voltagem, etc..) - opcao_valor_1:String(250) *(informar o valor da variação 1, ex: Azul, GG, 110v, etc) - opcao_titulo_2:String(250) (informar o titulo da segunda variação, ex: Cor, Tamanho, Voltagem, etc..) - opcao_valor_2:String(250) (informar o valor da segunda variação, ex: Azul, GG, 110v, etc) - imagem:String(400) (url 	

		da imagem da variação)	
--	--	------------------------	--

Exemplo de código json enviado:

```
{ "dados" : [ { "titulo" : "Tênis Nike SB Stefan Janoski Turbo Green", "ref" : "0006002.07",
"codebar" : "9990001681333", "estoque" : 1, "limite_estoque" : 1, "preco" : 346.89,
"desconto" : 0, "descontoavista" : 0, "destaque" : "0", "lançamento" : "0", "oferta" : "0", "peso"
: 10.00, "altura" : 15, "comprimento" : 30, "largura" : 20, "unidade" : "PC", "cod_fornecedor" :
32, "prazo_entrega_adicional" : 0, "tags" : "tênis, street, nike", "descricao" : "Tênis Nike SB
Stefan Janoski Turbo Green", "titulo_agrupador" : "Tênis Nike SB Stefan Janoski Turbo Green",
"variacao" : "Cor:Turbo Green", "fretegratis" : "0", "situacao" : "1", "categorias" : [ 31,43 ],
"imagens" : [ "http://www.lojadanike.com.br/ArquivoExibir/137681.jpg",
"http://media.wtennis.com.br/catalog/product/cache/2/image/800x800/9df78eab33525d08d
6e5fb8d27136e95/n/i/nike_air_max_finale__1_109_1003823_10.jpg" ], "grade" : [ { "ref" :
"0006002.07.33", "peso" : 0, "altura" : 0, "comprimento" : 0, "largura" : 0, "preco" : 0,
"estoque" : 31, "limite_estoque" : 1, "opcao_titulo_1" : "Tamanho", "opcao_valor_1" : "33",
"opcao_titulo_2" : "", "opcao_valor_2" : "", "imagem" :
"http://media.wtennis.com.br/catalog/product/cache/2/image/800x800/9df78eab33525d08d
6e5fb8d27136e95/n/i/nike_air_max_finale__1_109_1003823_10.jpg" }, { "ref" :
"0006002.07.34", "peso" : 0, "altura" : 0, "comprimento" : 0, "largura" : 0, "preco" : 0,
"estoque" : 21, "limite_estoque" : 1, "opcao_titulo_1" : "Tamanho", "opcao_valor_1" : "34",
"opcao_titulo_2" : "", "opcao_valor_2" : "", "imagem" :
"http://www.lojadanike.com.br/ArquivoExibir/137681.jpg" }, { "ref" : "0006002.07.39", "peso"
: 0, "altura" : 0, "comprimento" : 0, "largura" : 0, "preco" : 0, "estoque" : 20, "limite_estoque" :
1, "opcao_titulo_1" : "Tamanho", "opcao_valor_1" : "39", "opcao_titulo_2" : "",
"opcao_valor_2" : "", "imagem" : "http://www.lojadanike.com.br/ArquivoExibir/137681.jpg" }
] ] }
```

Parametros de saída:

ATENÇÃO: Os parâmetros de saída são retornados em um array com um item para cada produto enviado. Os dados estão no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
codigo	Inteiro(11)	Quando a referencia já esta cadastrada, o registro é atualizado e é retornado o código do produto no ecommerce, quando é um novo registro, é retornado o novo código gerado no sistema.	10

Exemplo de código json retornado:

```
[{"resultado":"1","msg":"sucesso, registro alterado","codigo":"433"}]
```

addCategoria

Adiciona ou altera uma categoria do ecommerce.

Parâmetros de entrada:

ATENÇÃO: A entrada consiste em um campo "dados" do tipo array, com os dados de um ou múltiplos registros de categorias.

Os campos do array "dados" seguem abaixo:

Nome	Tipo	Descrição	Exemplo
codigo	Inteiro(11)	Codigo da categoria no ecommerce. Quando informado, a intenção é alterar o registro	123
titulo*	String(60)	Titulo da categoria.	Calçados
codigo_catpai*	Inteiro(11)	Codigo da categoria "Pai". Quando é uma categoria principal, informar "0"	0
ordem_listagem	Inteiro(11)	Valor numérico que define a ordem de listagem. É utilizada a ordem ascendente dos valores.	10
descricao	Text	Descrição da categoria. Utilizar no máximo 240 caracteres	Os melhores calçados estão aqui.
keywords	Text	Palavras chaves relacionadas a categoria	Sapatos,tênis,sandalias

Exemplo de código json enviado:

```
{ "dados": [{"titulo": "Scarpin", "codigo_catpai": 1, "ordem_listagem": 20, "descricao": "descri\u00e7\u00e3o categoria scarpin", "keywords": "keywords categoria scarpin"}] }
```

Parametros de saída:

ATENÇÃO: Os parâmetros de saída são retornados em um array com um item para cada produto enviado. Os dados estão no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
codigo	Inteiro(11)	Quando a referencia já esta cadastrada, o registro é atualizado e é retornado o código do produto no ecommerce, quando é um novo registro, é retornado o novo código gerado no sistema.	10

Exemplo de código json retornado:

```
[{"resultado": "1", "msg": "sucesso, registro inserido", "codigo": 56}]
```

addMarca

Adiciona ou altera uma marca/fabricante do sistema.

Parâmetros de entrada:

ATENÇÃO: A entrada consiste em um campo “dados” do tipo array, com os dados de um ou múltiplos registros de categorias.

Os campos do array “dados” seguem abaixo:

Nome	Tipo	Descrição	Exemplo
codigo	Inteiro(11)	Codigo da marca no ecommerce. Quando informado, a intenção é alterar o registro	123
titulo*	String(150)	Titulo da marca.	Nike
ordem_listagem	Inteiro(11)	Valor numérico que define a ordem de listagem. É utilizada a ordem ascendente dos valores.	10

Exemplo de código json enviado:

```
{ "dados": [{"codigo":32,"titulo":"Rebook","ordem_listagem":20}, {"codigo":33,"titulo":"Nike","ordem_listagem":30}] }
```

Parametros de saída:

ATENÇÃO: Os parâmetros de saída são retornados em um array com um item para cada produto enviado. Os dados estão no formato json onde:

Nome	Tipo	Descrição	Exemplo
resultado	String(1)	1 para ok, e 0 para erro	1
msg	String(250)	Texto com a mensagem de retorno	sucesso
codigo	Inteiro(11)	Quando a referencia já esta cadastrada, o registro é atualizado e é retornado o código do produto no ecommerce, quando é um novo registro, é retornado o novo código gerado no sistema.	10

Exemplo de código json retornado:

```
[{"resultado":"1","msg":"sucesso, registro alterado","codigo":"32"}, {"resultado":"1","msg":"sucesso, registro alterado","codigo":"33"}]
```