

# Net Agencia Digital API v1.2.7

## Olá, tudo bem?!

Abaixo você encontrará uma lista atual dos métodos disponíveis na versão 1.2.7. Se precisar de ajuda ou suporte, por favor, entre em contato através do e-mail [suporte@netrmc.com.br](mailto:suporte@netrmc.com.br). Para usar a API, você deverá contratar o acesso e para isso, entre em contato com o nosso departamento comercial.

Todas as requisições são antes validadas com uma token de acesso (exceto funções de consultas públicas), obtida pela autenticação com usuário e uma chave de acesso.

Por favor, esteja ciente de que isso ainda não é um produto totalmente liberado e, como tal, não somos capazes de fornecer um suporte completo por enquanto. Se você tiver problemas, no entanto, não hesite em contactar através do e-mail [suporte@netrmc.com.br](mailto:suporte@netrmc.com.br) e faremos o nosso melhor para ajudar, embora possa demorar um pouco mais do que o normal.

## Funções Públicas

Alguns métodos são disponibilizados na API onde não há necessidade de autenticação, podendo ser usados por quaisquer integrações.

São estas:

- **Consulta de Frete** - Consultar frete baseado no CEP.

**Atenção:** Alterações destes métodos podem ser feitas a qualquer momento, sem aviso prévio, podendo ter o acesso público bloqueado temporariamente ou indefinidamente.

## Autenticação

Antes de efetuar qualquer requisição na API, primeiro é necessário realizar a autenticação, codificados como [HTTP autenticação básica](#); ou seja, a requisição de autenticação deve incluir uma Authorization: Basic xxxx, onde xxxx é a codificação Base64 de uma string construída como identifier:secret, onde identificador (identifier) é ou ID de usuário e a chave (secret) é a sua chave de acesso da API fornecida em seu painel de controle.

Nós sugerimos fortemente que você use a chave de acesso da API com segurança, uma vez que diminui a chance de ataque onde possa ter seus dados e até mesmo o acesso a informações sigilosas comprometidos. Você pode a qualquer momento alterar a sua chave de autenticação em seu painel de controle e evite utilizar a mesma em mais de uma integração.

Ao gerar a chave de acesso para as requisições, ela terá 15 minutos de validade, sendo renovado a cada requisição. Após expirada a chave por inatividade, será necessário realizar novamente a autenticação. A chave não poderá ser gerada novamente caso esteja ativa.

# Limite de acesso/requisição

O limite de acesso definido em nossa API tem o objetivo de prevenir abusos de serviços de terceiros ou usuários. Ao atingir o limite, será dada uma resposta 429 Too Many Requests.

Nosso limite foi definido com o único objetivo de amenizar a frequência de requisição a nossa infra-estrutura, suprimindo assim o excesso de requisições.

A limitação é por cliente, ou seja, uma mesma integração pode realizar várias requisições em múltiplos clientes e cada qual com a sua limitação.

Período

É possível fazer requisições até atingir o limite descrito no período abaixo.

|           |                 |
|-----------|-----------------|
| 1 segundo | 3 requisições   |
| 1 minuto  | 180 requisições |

Toda requisição irá retornar os headers descritos na tabela abaixo.

| Chave               | Valor | Descrição  |
|---------------------|-------|--|
| api-request-counter | 1     | Contador de requisições no período                           |
| api-request-limit   | 180   | Valor limite de requisições                                  |
| api-request-expire  | 60    | Quantos segundos foi definido para expirar aquela requisição |

## REFERENCE

## Autenticação

/oauth

POST <http://www.seudominio.com.br/ws/v1/oauth>

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

```
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("");
Response response = client.target("http://www.seudominio.com.br/ws/v1/oauth")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("Authorization", "Basic ABC123")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
401
    HEADERS
    WWW-Authenticate:Basic realm="protected"
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "token": "1a1b2c3d4e5f6g7h...",
  "expires_in": 1234567890
}
```

## Formas de Entrega

Consulta e lista das formas de entrega e valores caso seja demandado.

# Lista

## GET

GET <http://www.seudominio.com.br/ws/v1/shipping/list>

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/shipping/list")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
```

```
  "status": 200,
```

```
  "result": [
```

```
    {
```

```
{
  "id": "module_code",
  "title": "Title...",
  "image": "///cdn.iset.io/assets/...",
  "description": "Module description...",
  "status": true
}
]
```

Obtem a lista de formas de entrega disponíveis na loja, com o status de cada módulo conforme configurado pelo lojista.

## Consultar Frete

A consulta de frete deve ser feita em modo post, porém os parâmetros passados na query string seguem uma formatação necessária.

Exemplo: Formato da URL da requisição sem especificar um módulo.

`http://www.seudominio.com.br/ws/v1/shipping/31310530`

Exemplo: Formato da URL da requisição especificando um módulo.

`http://www.seudominio.com.br/ws/v1/shipping/31310530/sedex`

### POST

`POST http://www.seudominio.com.br/ws/v1/shipping/postcep(/method)`

### Parameters

postcep  
CEP para consulta do frete. Example: 31310530.  
Number  
method  
Método específico para consulta. Example: sedex.  
String

## Request

### ATTRIBUTES

order\_total  
required  
number  
Valor total do pedido em formato de moeda.  
1.23  
order\_weight  
required  
number

Peso total do pedido em formato de moeda com 3 decimais.

0.321

products

required

array

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

import javax.ws.rs.client.Client;

import javax.ws.rs.client.ClientBuilder;

import javax.ws.rs.client.Entity;

import javax.ws.rs.core.Response;

import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();

Entity payload = Entity.json("{ \"order\_total\": 1.23, \"order\_weight\": 0.321, \"products\": [ { \"quantity\": 1, \"weight\": 0.321, \"dimensions\": \"16#2#11\", \"free\_shipping\": false, \"block\_shipping\": \"sedex;pac,...\" } ]}");

Response response =

client.target("http://www.seudominio.com.br/ws/v1/shipping/{postcep}/{method}")

.request(MediaType.APPLICATION\_JSON\_TYPE)

.post(payload);

System.out.println("status: " + response.getStatus());

System.out.println("headers: " + response.getHeaders());

System.out.println("body:" + response.readEntity(String.class));

## Response

400

HEADERS

Content-Type:application/json

BODY

{

"status": 400,

"message": "Error validation message..."

```
}
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "result": [
    {
      "code": "sedex",
      "title": "SEDEX",
      "image": "///cdn.iset.io/...",
      "description": "Seu produto será enviado assim que confirmado o pagamento...",
      "cost": 12.3
    }
  ]
}
```

Consulta o frete baseado no CEP informado, retornando as formas de entrega disponíveis com os valores respectivos.

### Required Parameters

|         |   |
|---------|---|
| postcep | CEP a ser consultado.<br>Formato esperado (0-9): 99999999 |
|---------|---|

### Optional Parameters

|        |  |
|--------|--|
| method | Nome do módulo para consulta em específico.<br>Formato: (a-z,_) - min. 3, max. 20. |
|--------|--|

## Formas de Pagamento

Consulta e lista das formas de pagamento e valores caso seja demandado.

## Lista

GET

GET<http://www.seudominio.com.br/ws/v1/payment/list>

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.core</groupId>
```

```
//   <artifactId>jersey-client</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.media</groupId>
```

```
//   <artifactId>jersey-media-json-jackson</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/payment/list")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "result": {
    "boleto": [
      {
        "id": "boleto",
        "title": "Banco Real",
        "image": "/imagens/boleto/real.gif",
        "status": true
      }
    ]
  }
}
```



```
    }
  ],
  "credito": [
    {
      "id": "mercadopago_visa",
      "title": "Visa",
      "image": "/imagens/modulos/ico_visa.gif",
      "status": true
    },
    {
      "id": "mercadopago_master",
      "title": "Mastercard",
      "image": "/imagens/modulos/ico_mastercard.gif",
      "status": true
    },
  ],
  "outros": [
    {
      "id": "custompay",
      "title": "Faturado",
      "image": "//cdn.iset.io/assets/...",
      "status": true
    }
  ]
}
```

Obtem a lista de formas de pagamento disponíveis na loja conforme configurado pelo lojista.

## Clientes

Cadastro / Consulta de clientes na loja.

### Cadastro

#### Required Parameters

|           |  |
|-----------|--|
| type      | Tipo de cadastro do cliente. (F = Física / J = Jurídica) |
| email     | E-mail de contato do cliente.                            |
| firstname | Primeiro nome do cliente.                                |
| lastname  | Sobrenome do cliente.                                    |

|          |  |
|----------|--|
| cpf cnpj | CPF / CNPJ do cliente com formatação.  |
| phone    | Apenas o telefone principal é obrigatório.<br>Este campo é do tipo array, sendo o primeiro elemento o telefone principal e o segundo elemento o celular. |
| address  | Endereço de cadastro do cliente.   |

#### Optional Parameters

|              |   |
|--------------|---|
| gender       | Gênero do cliente. (M = Masculino / F = Feminino).    |
| civil_status | Estado Civil. (Campo livre: casado, solteiro, etc...) |
| dob          | Data de nascimento.<br>Formatação: YYYY-MM-DD         |
| rg           | RG de identificação do cliente.                       |

#### Inserir

PUT <http://www.seudominio.com.br/ws/v1/customer/insert>

## Request

### ATTRIBUTES

type

required

string

Tipo de cadastro do cliente. (F = Física / J = Jurídica)

F

email

required

string

E-mail de contato do cliente.

email@dominio.com

firstname

required

string

Primeiro nome do cliente.

Nome

```
lastname
required
string
Sobrenome do cliente.
Sobrenome
gender
string
Gênero do cliente. (M=Masculino/F=Feminino).
M
civil_status
string
Estado Cível. (Campo livre: casado, solteiro, etc...)
Casado
dob
string
01-01 (optional, string) - Data de nascimento.
1900
cpf|cnpj
required
string
CPF / CNPJ do cliente com formatação.
string
rg
string
RG de identificação do cliente.
string
phone
array
address
object
Production
Java
Try
// Maven : Add these dependencies to your pom.xml (java6+)
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
```

```

import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{
    \"type\": \"F\",
    \"email\": \"email@dominio.com\",
    \"firstname\": \"Primeiro Nome\",
    \"lastname\": \"Sobrenome\",
    \"gender\": \"M\",
    \"civil_status\": \"Casado\",
    \"dob\": \"1900-01-01\",
    \"cpf\": \"123.456.789-00\",
    \"phone\": [
        {
            \"ddd\": 31,
            \"telephone\": 99999999,
        },
        {
            \"ddd\": 31,
            \"telephone\": 999999999,
        }
    ],
    \"address\": {
        \"postcode\": 31310530,
        \"street\": \"Rua nome da esquina\",
        \"number\": 123,
        \"complement\": \"casa\",
        \"suburb\": \"Bairro\",
        \"city\": \"Cidade\",
        \"state\": \"MG\"
    }
}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/customer/insert")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .put(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));

```

## Response

400

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "customerId": 123
}
```

- Se existir algum cliente com o mesmo e-mail informado, será retornado o código do mesmo.

## Atualizar

POST <http://www.seudominio.com.br/ws/v1/customer/update/id>

## Request

### ATTRIBUTES

field

required

string

Description.

value

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{}");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/customer/update/id")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "customerId": 123
}
```

Note implemented yet.

## Endereços

---

Required Parameters

---

|                     |   |
|---------------------|---|
| firstname           | Primeiro nome do cliente.   |
| lastname            | Sobrenome do cliente.   |
| postcode            | CEP do endereço.  |
| address1            | Logradouro.   |
| number              | Número.   |
| suburb              | Bairro.   |
| city                | Cidade.   |
| state               | Estado.   |
| country             | Nome do país.   |
| Optional Parameters |   |
| address2            | Complemento do logradouro.  |
| company             | Nome da empresa.  |
| default             | Define se o endereço a ser inserido, deverá ser marcado como padrão do cliente. |

## Listar/Consultar

GET <http://www.seudominio.com.br/ws/v1/customer/{cld}/address/{list|ald}>

## Parameters

cld

ID de cadastro de algum cliente Example: 143.

Number

ald

ID de cadastro de algum endereço do cliente Example: 6.

Number

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

```
// <artifactId>jersey-client</artifactId>
// <version>2.8</version>
// </dependency>
// <dependency>
// <groupId>org.glassfish.jersey.media</groupId>
// <artifactId>jersey-media-json-jackson</artifactId>
// <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Response response =
client.target("http://www.seudominio.com.br/ws/v1/customer/{cld}/address/{list|ald}")
.request(MediaType.TEXT_PLAIN_TYPE)
.header("access-token", "1a1b2c3d4e5f6g7h...")
.get();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "totalFound": 2,
  "result": [
    {
```



```
{
  "id": 6,
  "company": "iSET",
  "firstname": "Iset",
  "lastname": "Teste",
  "postcode": "31000-000",
  "address1": "Rua Do vigário",
  "address2": "",
  "number": "22",
  "suburb": "bairro",
  "city": "Belo Horizonte",
  "state": "MG",
  "country": "Brasil"
}
]
```

## Adicionar

PUT [http://www.seudominio.com.br/ws/v1/customer/\(id\)/address](http://www.seudominio.com.br/ws/v1/customer/(id)/address)

## Parameters

id  
ID de cadastro de algum cliente Example: 6.  
Number

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"firstname\": \"Primeiro nome\", \"lastname\": \"Segundo nome\", \"company\": \"Empresa\", \"postcode\": \"31000000\", \"address1\": \"Al. dos Passadores\", \"address2\": \"paralela a av. dos carros\", \"number\": 140, \"suburb\": \"Sao Bento\", \"city\": \"Belo Horizonte\", \"state\": \"MG\", \"country\": \"BR\", \"default\": true}");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/customer/{id}/address")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .put(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
        Content-Type:application/json
BODY
{
    "status": 400,
    "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
    "status": 200,
    "customerId": 6,
    "addressId": 41
}
```

# Categorias

## Consulta

### Árvore

GET [http://www.seudominio.com.br/ws/v1/category/tree\(/id\)](http://www.seudominio.com.br/ws/v1/category/tree(/id))

### Parameters

**id**  
ID de cadastro de alguma categoria específica Example: 143.  
Number

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.core</groupId>
```

```
//   <artifactId>jersey-client</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.media</groupId>
```

```
//   <artifactId>jersey-media-json-jackson</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response =
```

```
client.target("http://www.seudominio.com.br/ws/v1/category/tree({/id})")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

400

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "response": {
    "174": {
      "id": 174,
      "name": "Bonés",
      "img": "",
      "img_h": "",
      "level": 1,
      "parent_id": 0,
      "subCategories": {
        "175": {
          "id": 175,
          "name": "Official",
          "img": "",
          "img_h": "",
          "level": 2,
          "parent_id": 174,
          "subCategories": []
        }
      }
    }
  }
}
```

## Info. de uma Categoria

GET<http://www.seudominio.com.br/ws/v1/category/id>

## Parameters

id  
ID de cadastro de alguma categoria específica Example: 174.  
Number

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.core</groupId>
```

```
//   <artifactId>jersey-client</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.media</groupId>
```

```
//   <artifactId>jersey-media-json-jackson</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/category/{id}")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

400

HEADERS

Content-Type:application/json

BODY

```
{
```

```
  "status": 400,
```

```
  "message": "Error validation message..."
```

```
}
```

## Response

200

### HEADERS

Content-Type:application/json

### BODY

```
{
  "status": 200,
  "response": {
    "id": 174,
    "parent_id": 0,
    "title": "Bonés",
    "subtitle": null,
    "seo_title": "Bonés BR Shop www.brshops.com",
    "seo_keywords": "Bonés BR Shop www.brshops.com",
    "seo_description": "Bonés BR Shop www.brshops.com",
    "slug": "bones",
    "discount": 0,
    "hidden": 0
  }
}
```

## Cadastro

Gerenciamento e consulta de categorias da loja.

### Required Parameters

|      |  |
|------|--|
| name | Nome da categoria.<br>Min. 4 e Máx. 50 caracteres. |
|------|--|

### Optional Parameters

|            |  |
|------------|--|
| subtitle   | Subtítulo da categoria.<br>Min. 3 e sem limite máximo de caracteres.                             |
| parent_id  | Código da categoria pai a quem esta categoria<br>pertence.<br>Valor 0 (padrão) = Categoria raiz. |
| sort_order | Ordenação da categoria.<br>Uso apenas quando o tipo de ordem configurado for<br>manual.          |

|          |   |
|----------|---|
| discount | Desconto a ser oferecido a todos os produtos relacionados a esta categoria. |
| hidden   | Categoria oculta. (true false).   |

Inserir

POST <http://www.seudominio.com.br/ws/v1/category/insert>

## Request

### ATTRIBUTES

name

required

string

Nome da categoria.

Camisetas Libra

subtitle

string

Subtítulo da categoria.

Camisetas da marca libra com fios de ouro entrelaçados com prata e bronze...

parent\_id

number

Código da categoria pai. 0 = Categoria raiz.

0

sort\_order

number

Ordenação da categoria.

0

discount

number

Desconto a ser oferecido aos produtos.

1.5

hidden

boolean

Categoria oculta. (true|false).

false

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

```
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"name\": \"Camisetas Libra\", \"subtitle\": \"Camisetas da marca libra com fios de ouro entrelaçados com prata e bronze...\", \"parent_id\": 0, \"sort_order\": 0, \"discount\": 1.5, \"hidden\": false}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/category/insert")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "categ_id": 123
}
```



## Atualizar

POST <http://www.seudominio.com.br/ws/v1/category/update/id>

## Parameters

id  
ID de cadastro da categoria a ser alterada. Example: 123.  
Number

## Request

### ATTRIBUTES

name  
required  
string  
Nome da categoria.  
Camisetas Libra  
subtitle  
string  
Subtítulo da categoria.  
Camisetas da marca libra com fios de ouro entrelaçados com prata e bronze...  
parent\_id  
number  
Código da categoria pai. 0 = Categoria raiz.  
0  
sort\_order  
number  
Ordenação da categoria.  
0  
discount  
number  
Desconto a ser oferecido aos produtos.  
1.5  
hidden  
boolean  
Categoria oculta. (true|false).  
false  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
//   <groupId>org.glassfish.jersey.core</groupId>  
//   <artifactId>jersey-client</artifactId>  
//   <version>2.8</version>

```
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"name\": \"Camisetas Libra\", \"subtitle\": \"Camisetas da marca libra com fios de ouro entrelaçados com prata e bronze...\", \"parent_id\": 0, \"sort_order\": 0, \"discount\": 1.5, \"hidden\": false}");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/category/update/{id}")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "categ_id": 123
}
```

# Produtos

Gerenciamento e consulta de produtos.

## Cadastro

O cadastramento de produtos ainda encontra-se em base de testes e de forma simplória, em breve será implementado novas informações para cadastro.

### Required Parameters

|          |  |
|----------|--|
| status   | Status de cadastro do produto. Aceito: 1/0 |
| name     | Nome do produto.                           |
| price    | Preço de venda do produto.                 |
| quantity | Quantidade em estoque.                     |

### Optional Parameters

|             |  |
|-------------|--|
| cod_ref     | Código de referência do produto (SKU).   |
| gtin        | Código GTIN do produto.  |
| mpn         | Código MPN do produto.   |
| ncm         | Código NCM do produto.   |
| model       | Modelo/Marca do produto.   |
| check_stock | Método de controle do estoque do produto.<br>product = Monitorar estoque.<br>attributes = Monitorar estoque por variações/atributos.<br>disabled = Não monitorar estoque.<br>Padrão: product |
| allow_buy   | Permitir a compra deste produto.<br>Padrão: true   |

|                    |   |
|--------------------|---|
| weight             | Peso do produto.  |
| dimensions         | Dimensões do produto.<br>Array com valores numéricos, sendo<br>(Comprimento/Largura/Altura) |
| quantity_max_buy   | Quantidade máxima de compra deste produto.<br>Padrão: 0                                     |
| quantity_min_buy   | Quantidade mínima de compra deste produto.<br>Padrão: 0                                     |
| quantity_critical  | Quantidade critica em que o estoque deve chegar para realizar a notificação.<br>Padrão: 0   |
| handling_days      | Número de dias para manuseio do produto.<br>Padrão: 0                                       |
| description_small  | Breve descrição do produto.   |
| description_full   | Descrição completa do produto.  |
| description_mobile | Descrição completa do produto para a plataforma mobile (Caso possua o serviço ativo).       |
| manufacturer_id    | Código de cadastro do fabricante do produto.  |
| availability       | Texto customizado para disponibilidade do produto.  |
| categories         | Conjunto de IDs das categorias que o produto será relacionado.                              |
| attributes         | Conjunto de atributos que o produto será relacionado.                                       |

|                 |  |
|-----------------|--|
| attrGroupId     | Código do grupo de atributos a ser relacionado com o produto.                                    |
| allowGiftWrap   | Produto aceita embalagem para presente.<br>Padrão: false   |
| isFreeShipping  | Produto com frete-grátis.<br>Padrão: false   |
| highlight       | Produto em destaque.<br>Padrão: false  |
| hotDeal         | Produto em super oferta.<br>Padrão: false  |
| newRelease      | Produto lançamento.<br>Padrão: false   |
| exclusive       | Produto exclusivo.<br>Padrão: false  |
| slug            | URL amigável.<br>Exemplo: nome-do-produto-inovador<br>Padrão: gerado automático pela plataforma. |
| seo_title       | Título customizado para o produto.   |
| seo_description | Descrição meta customizada para o produto.   |
| seo_keywords    | Paravras chave meta, customizado para o produto.   |
| images          | Imagens do produto.  |
| specifications  | Características/Especificações do produto.   |

Inserir

POST <http://www.seudominio.com.br/ws/v1/product/insert>

## Request

### ATTRIBUTES

status  
required  
number  
Status do produto.  
1  
allow\_buy  
string  
Permitir a compra deste produto.  
true  
cod\_ref  
string  
Código de referência do produto.  
ABC123  
gtin  
string  
Código de GTIN do produto.  
91922010291  
mpn  
string  
Código de MPN do produto.  
9999999  
ncm  
string  
Código de NCM do produto.  
999999  
name  
required  
string  
Nome do produto.  
Nome do produto  
model  
string  
Modelo/Marca do produto.  
Modelo  
price  
required  
number  
Preço de venda do produto.  
2.99  
weight  
number  
Peso do produto.  
0.3  
dimensions

array  
description\_small  
string  
Breve descrição do produto.  
Breve descrição  
description\_full  
string  
Completa descrição do produto.  
Completa descrição  
description\_mobile  
string  
Completa descrição do produto para a plataforma mobile.  
Completa descrição  
check\_stock  
string  
Tipo de monitoramento do estoque.  
product  
quantity  
required  
number  
Quantidade em estoque do produto.  
5  
quantity\_max\_buy  
number  
Quantidade máxima de compra deste produto.  
0  
quantity\_min\_buy  
number  
Quantidade mínima de compra deste produto.  
0  
quantity\_critical  
number  
Quantidade crítica em que o estoque deve chegar para realizar a notificação.  
0  
handling\_days  
number  
Número de dias para manuseio do produto.  
0  
manufacturer\_id  
number  
Código de cadastro do fabricante deste produto.  
0  
availability  
string  
Texto de disponibilidade deste produto.  
Em estoque  
categories  
array

attributes  
array  
attrGroupIid  
number  
Código do grupo de atributos a ser relacionado com o produto.  
0  
allowGiftWrap  
string  
Permite embalar para presente (sim/não = true/false).  
false  
isFreeShipping  
string  
Frete-Grátis (sim/não = true/false).  
false  
highlight  
string  
Destaque (sim/não = true/false).  
false  
hotDeal  
string  
Super Oferta (sim/não = true/false).  
false  
newRelease  
string  
Lançamento (sim/não = true/false).  
false  
exclusive  
string  
Exclusivo (sim/não = true/false).  
false  
images  
array  
specifications  
array  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
//   <groupId>org.glassfish.jersey.core</groupId>  
//   <artifactId>jersey-client</artifactId>  
//   <version>2.8</version>  
// </dependency>  
// <dependency>  
//   <groupId>org.glassfish.jersey.media</groupId>  
//   <artifactId>jersey-media-json-jackson</artifactId>  
//   <version>2.8</version>  
// </dependency>



```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{
    \"status\": 1,
    \"cod_ref\": \"ABC123\",
    \"name\": \"Nome de um produto a ser cadastrado\",
    \"model\": \"Modelo\",
    \"price\": 2.99,
    \"check_stock\": \"product\",
    \"quantity\": 5,
    \"quantity_max_buy\": 0,
    \"quantity_min_buy\": 0,
    \"quantity_critical\": 3,
    \"categories\": [
        123,456
    ],
    \"attrGroupId\": 1,
    \"attributes\": [
        {
            \"optionsGroupId\": 1,
            \"isDefault\": 1,
            \"codref\": \"ABC566\",
            \"stock\": 0,
            \"optionsIds\": {
                6544,98777
            }
        },
        {
            \"optionsGroupId\": 1,
            \"isDefault\": 0,
            \"codref\": \"DEF123\",
            \"stock\": 0,
            \"optionsIds\": {
                0911,3455
            }
        }
    ],
    \"images\": {
        \"https://www.dominio.com.br/image1.jpg\",
        \"https://www.dominio.com.br/image2.jpg\"
    },
}
```

```

        \"specifications\": [
            {
                \"key\": \"Grupo\",
                \"value\": \"Valor1,valor2,valor3\",
                \"hidden\": false,
                \"print\": true
            }
        ]
    });
    Response response = client.target("http://www.seudominio.com.br/ws/v1/product/insert")
        .request(MediaType.TEXT_PLAIN_TYPE)
        .header("access-token", "1a1b2c3d4e5f6g7h...")
        .post(payload);

    System.out.println("status: " + response.getStatus());
    System.out.println("headers: " + response.getHeaders());
    System.out.println("body:" + response.readEntity(String.class));

```

## Response

```

200
    HEADERS
    Content-Type:application/json
BODY
{
    "status": 200,
    "productId": 123
}

```

## Atualizar

POST <http://www.seudominio.com.br/ws/v1/product/update/id>

## Parameters

id  
ID de cadastro do produto a ser alterado. Example: 123.  
Number

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

```
// <groupId>org.glassfish.jersey.core</groupId>
// <artifactId>jersey-client</artifactId>
// <version>2.8</version>
// </dependency>
// <dependency>
// <groupId>org.glassfish.jersey.media</groupId>
// <artifactId>jersey-media-json-jackson</artifactId>
// <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{
    \"status\": 0,
    \"model\": \"Nova marca\",
}");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/product/update/{id}")
.request(MediaType.TEXT_PLAIN_TYPE)
.header("access-token", "1a1b2c3d4e5f6g7h...")
.post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
        Content-Type:application/json
    BODY
    {
        "status": 200,
        "productId": 123
    }
```

Qualquer campo discriminado na tabela detalhada para cadastro de um produto pode ser alterado.

Deve ser passado pelo menos 1 parâmetro para alteração;

# Consulta/Listagem

Pesquisa rápida

POST <http://www.seudominio.com.br/ws/v1/product/search>

## Request

### ATTRIBUTES

query

required

string

Termo de busca.

camisa

limit

number

Limite do resultado da pesquisa.

10

img\_w

number

Largura máxima da mini imagem.

50

img\_h

number

Altura máxima da mini imagem.

50

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

//   <groupId>org.glassfish.jersey.core</groupId>

//   <artifactId>jersey-client</artifactId>

//   <version>2.8</version>

// </dependency>

// <dependency>

//   <groupId>org.glassfish.jersey.media</groupId>

//   <artifactId>jersey-media-json-jackson</artifactId>

//   <version>2.8</version>

// </dependency>

import javax.ws.rs.client.Client;

import javax.ws.rs.client.ClientBuilder;

import javax.ws.rs.client.Entity;

import javax.ws.rs.core.Response;

import javax.ws.rs.core.MediaType;

```
Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"query\": \"camisa\", \"limit\": 10, \"img_w\": 50, \"img_h\": 50}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/search")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "keywords": "bone",
  "productsFound": 1,
  "products": [
    {
      "id": 7121,
      "name": "Tênis Hocks Del Mar Mid Camisa Marrom",
      "cod_ref": "1523",
      "model": "Del Mar Mid",
      "image": "//cdn.iset.io/...",
      "quantity": 10,
      "price_float": 213,
      "price_old": false,
      "price": "R$ 213,00"
    },
    {
      "id": 9585,
      "name": "Camisa Vans Micro Feminino Blue",
      "cod_ref": "916F3201",
      "model": "Micro",
      "image": "//cdn.iset.io/...",
      "quantity": 2,
      "price_float": 239.9,
      "price_old": false,
      "price": "R$ 239,90"
    }
  ]
}
```

Pesquisa rápida e resumida de produtos baseado no termo pesquisado. Para informações completas, utilize o método de Listagem: /list.

#### Required Parameters

|       |                        |
|-------|------------------------|
| query | Termo a ser pesquisado |
|-------|------------------------|

#### Optional Parameters

|       |   |
|-------|---|
| limit | Limite do resultado da pesquisa. Padrão: 10   |
| img_w | Largura máxima da mini imagem de exibição do produto.<br>Padrão: (definido no painel de controle da loja) |
| img_h | Altura máxima da mini imagem de exibição do produto.<br>Padrão: (definido no painel de controle da loja)  |

#### Listagem

POST <http://www.seudominio.com.br/ws/v1/product/list>

## Request

### ATTRIBUTES

page

number

Padrão: 1

1

perPage

number

Padrão: 20

20

order

string

Coluna para ordenação do resultado.

products\_id

order\_type

string

Tipo de ordenação da coluna de ordem definida.

```

desc
Production
Java
Try
// Maven : Add these dependencies to your pom.xml (java6+)
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"page\": 1, \"perPage\": 20, \"order\": \"products_id\", \"order_type\": \"desc\"}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/list")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));

```

## Response

```

200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "page": 1,
  "perPage": 20,
  "pages": 2,
  "totalFound": 20,
  "totalCount": 22,
  "result": [

```

```
{
  "id": 18662,
  "date_added": "2015-11-23 14:36:02",
  "last_modified": "2016-06-13 17:21:47",
  "status": false,
  "cod_ref": "",
  "name": "Camiseta Starter Regata 1971 Grey",
  "model": "1971",
  "price": 12.91,
  "price_sale": 10.91,
  "weight": 1.2,
  "dimensions": {
    "length": 16,
    "height": 2,
    "width": 11
  },
  "stock_check_method": "attributes",
  "quantity": 1,
  "quantity_max_buy": 0,
  "quantity_min_buy": 0,
  "quantity_critical": 0,
  "description_small": "Breve descrição",
  "description_full": "Descrição completa",
  "orders": 0,
  "views": 0,
  "availability": "Imediato",
  "allow_buy": true,
  "points": 0,
  "image_default": "//cdn.iset.io/assets/...1.jpg",
  "images": [
    "//cdn.iset.io/assets/...2.jpg",
    "//cdn.iset.io/assets/...3.jpg",
    "//cdn.iset.io/assets/...4.jpg"
  ],
  "categories_path": "Limpa Estoque > 70% off",
  "categories": [
    {
      "id": "675",
      "name": "Limpa Estoque",
      "parent_id": "0"
    },
    {
      "id": "696",
      "name": "70% off",
      "parent_id": "675"
    }
  ],
  "attributes": [
```



```
{
  "id": "5201048",
  "isDefault": "0",
  "status": "1",
  "cod_ref": "",
  "stock": "0",
  "image": false,
  "priceDiff": {
    "": "0"
  },
  "weithDiff": {
    "": "0"
  },
  "price": 741.21,
  "weight": 0.34,
  "options": [
    {
      "title": "Tamanho",
      "value": "M"
    }
  ]
},
{
  "id": "5201049",
  "isDefault": "0",
  "status": "1",
  "cod_ref": "",
  "stock": "0",
  "image": false,
  "priceDiff": {
    "": "0"
  },
  "weithDiff": {
    "": "0"
  },
  "price": 122.3,
  "weight": 0.34,
  "options": [
    {
      "title": "Tamanho",
      "value": "G"
    }
  ]
}
],
{
  "id": 18661,
```

```
"date_added": "2015-11-23 14:34:58",
"last_modified": null,
"status": false,
"cod_ref": "",
"name": "Camiseta Starter Todos São Manos Grey",
"model": "Todos São Manos",
"price": 0,
"weight": 1.2,
"dimensions": {
  "length": 16,
  "height": 2,
  "width": 11
},
"stock_check_method": "attributes",
"quantity": 1,
"quantity_max_buy": 0,
"quantity_min_buy": 0,
"quantity_critical": 0,
"description_small": "Breve descrição",
"description_full": "Descrição completa",
"orders": 0,
"views": 0,
"availability": "Imediato",
"allow_buy": true,
"points": 0,
"image_default": "//cdn.iset.io/assets/...1.jpg",
"images": [
  "//cdn.iset.io/assets/...2.jpg",
  "//cdn.iset.io/assets/...3.jpg",
  "//cdn.iset.io/assets/...4.jpg"
],
"categories_path": "Limpa Estoque > 70% off",
"categories": [
  {
    "id": "675",
    "name": "Limpa Estoque",
    "parent_id": "0"
  },
  {
    "id": "696",
    "name": "70% off",
    "parent_id": "675"
  }
],
"attributes": [
  {
    "id": "5201048",
    "isDefault": "0",
```

```
    "status": "1",
    "cod_ref": "",
    "stock": "0",
    "image": false,
    "priceDiff": {
      "": "0"
    },
    "weithDiff": {
      "": "0"
    },
    "price": 2.3,
    "weight": 0.31,
    "options": [
      {
        "title": "Tamanho",
        "value": "M"
      }
    ]
  },
  {
    "id": "5201049",
    "isDefault": "0",
    "status": "1",
    "cod_ref": "",
    "stock": "0",
    "image": false,
    "priceDiff": {
      "": "0"
    },
    "weithDiff": {
      "": "0"
    },
    "price": 133.3,
    "weight": 1.34,
    "options": [
      {
        "title": "Tamanho",
        "value": "G"
      }
    ]
  }
]
}
```

Os produtos podem ser paginados, o limite padrão de retorno é de 50 produtos. O total de produtos encontrados é retornado na coluna totalFound. Você pode utilizar os seguintes parâmetros no post da requisição:

#### Optional Parameters

|                  |   |
|------------------|---|
| dateAddedFrom    | A partir da data de cadastro<br>YYYY-MM-DD.   |
| catегId          | Código de uma categoria dos<br>produtos a serem filtrados.  |
| stockLessThan    | Estoque abaixo de X.  |
| stockGreaterThan | Estoque acima de X.   |
| freeShipping     | O produto tem frete grátis.<br>Valor aceito: true ou false.   |
| status           | Status dos produtos.<br><br>Valor aceito: true ou false.  |
| order            | Coluna de ordenação da<br>listagem.<br>Valor padrão: products_id<br>Colunas aceitas: orders_id,<br>customers_name,<br>customers_email |
| order_type       | Tipo de ordenação da coluna<br>definida.<br>Valor padrão: desc<br>Valores aceitos: asc<br>(crescente) ou desc<br>(decrecente).        |
| page             | Número da página de<br>resultados. Padrão: 1  |
| perPage          | Número de itens por página a<br>ser obtido. Máximo: 200   |

## Obter produto por ID

GET <http://www.seudominio.com.br/ws/v1/product/id>

### Parameters

**id**  
ID do produto a ser consultado. Example: 18661.  
Number

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/{id}")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

### Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "totalFound": 1,
  "totalCount": 1,
  "limit": null,
  "result": [
    {
      "id": 18661,
      "date_added": "2015-11-23 14:34:58",
      "last_modified": null,
      "status": false,
      "cod_ref": "",
      "name": "Camiseta Starter Todos São Manos Grey",
      "model": "Todos São Manos",
      "price": 133.3,
      "price_sale": 133.3,
      "weight": 1.2,
      "dimensions": {
        "length": 16,
        "height": 2,
        "width": 11
      },
      "quantity": 1,
      "quantity_max_buy": 0,
      "quantity_min_buy": 0,
      "quantity_critical": 0,
      "stock_check_method": 3,
      "orders": 0,
      "views": 0,
      "availability": "Imediato",
      "allow_buy": true,
      "points": 0,
      "image_default": "//cdn.iset.io/assets/...1.jpg",
      "images": [
        "//cdn.iset.io/assets/...2.jpg",
        "//cdn.iset.io/assets/...3.jpg",
        "//cdn.iset.io/assets/...4.jpg",
      ],
      "categories_path": "Limpa Estoque > 70% off",
      "categories": [
        {
          "id": "675",
          "name": "Limpa Estoque",
          "parent_id": "0"
        },
        {
          "id": "696",
```

```
        "name": "70% off",
        "parent_id": "675"
    }
],
"attributes": [
    {
        "id": "5201048",
        "isDefault": "0",
        "status": "1",
        "cod_ref": "",
        "stock": "0",
        "image": false,
        "priceDiff": {
            "": "0"
        },
        "weithDiff": {
            "": "0"
        },
        "price": 133.3,
        "weight": 1.2,
        "options": [
            {
                "Tamanho": "M"
            }
        ]
    },
    {
        "id": "5201049",
        "isDefault": "0",
        "status": "1",
        "cod_ref": "",
        "stock": "0",
        "image": false,
        "priceDiff": {
            "": "0"
        },
        "weithDiff": {
            "": "0"
        },
        "price": 133.3,
        "weight": 1.2,
        "options": [
            {
                "Tamanho": "G"
            }
        ]
    }
]
```

```
}  
]  
}
```

## Atributos/Variações

Consulta/Cadastro de atributos dos produtos.

Consultar

GET <http://www.seudominio.com.br/ws/v1/product/id/attributes>

### Parameters

id  
ID de cadastro do produto a ser consultado. Example: 123.  
Number

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response =
```

```
client.target("http://www.seudominio.com.br/ws/v1/product/{id}/attributes")
```

```
.request(MediaType.TEXT_PLAIN_TYPE)
```

```
.header("access-token", "1a1b2c3d4e5f6g7h...")
```



```
.get();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "totalCount": 2,
  "result": [
    {
      "id": 3990174,
      "status": true,
      "codref": "",
      "quantity": 2,
      "image": "",
      "isDefault": true,
      "poids": [
        {
          "id": 1,
          "name": "Tamanho",
          "value": "7"
        }
      ]
    },
    {
      "id": 3990175,
      "status": true,
      "codref": "",
      "quantity": 1,
      "image": "",

```

```
{
  "isDefault": false,
  "poids": [
    {
      "id": 2,
      "name": "Tamanho",
      "value": "8"
    }
  ]
}
```

A coluna poids do retorno é os IDs de cadastro das opções de junção para aquele atributo. Exemplo: Tamanho + opção M tem o ID = 1, o ID 1 será o utilizado no cadastro de atributos daquele produto.

## Inserir

PUT<http://www.seudominio.com.br/ws/v1/product/id/attributes/insert>

## Parameters

id  
ID de cadastro do produto para inserção do novo atributo. Example: 123.  
Number

## Request

### ATTRIBUTES

optionsGroupId  
required  
number  
ID do grupo de atributos.  
1  
isDefault  
number  
Opção de atributo padrão (pré-selecionada).  
0  
codref  
string  
Código de referência do atributo.  
ABC566  
stock  
number  
Quantidade em estoque desta variação do produto.  
0

optionsIds  
required  
array  
priceDiff  
string  
Alteração de preço (Aceito: add/rem/fixed).  
NULL  
price  
number  
Valor do preço a ser alterado conforme campo priceDiff.  
0  
weightDiff  
string  
Alteração de peso (Aceito: add/rem/fixed).  
NULL  
weight  
number  
Valor do peso a ser alterado conforme campo priceDiff.  
0  
image  
string  
URL da imagem para o atributo.  
<https://www.dominio.com/image1.jpg>  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
// <groupId>org.glassfish.jersey.core</groupId>  
// <artifactId>jersey-client</artifactId>  
// <version>2.8</version>  
// </dependency>  
// <dependency>  
// <groupId>org.glassfish.jersey.media</groupId>  
// <artifactId>jersey-media-json-jackson</artifactId>  
// <version>2.8</version>  
// </dependency>

```
import javax.ws.rs.client.Client;  
import javax.ws.rs.client.ClientBuilder;  
import javax.ws.rs.client.Entity;  
import javax.ws.rs.core.Response;  
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();  
Entity<String> payload = Entity.text("{ \"optionsGroupId\": \"5201048\", \"isDefault\": 0,  
\"coderef\": \"ABC123\", \"stock\": 1, \"optionsIds\": [ 202, 203 ]}");
```

```
Response response =
client.target("http://www.seudominio.com.br/ws/v1/product/{id}/attributes/insert")
.request(MediaType.TEXT_PLAIN_TYPE)
.header("access-token", "1a1b2c3d4e5f6g7h...")
.put(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "productId": 123,
  "type": "new",
  "attributeId": 4443
}
```

Inserção de um atributo ao produto.  
Consulte os códigos das opções para a parametrização correta.

### Atualizar

POST <http://www.seudominio.com.br/ws/v1/product/id/attributes/update>

## Parameters

id  
ID de cadastro do produto para atualização dos atributos. Example: 22635.  
Number

## Request

### ATTRIBUTES

isDefault  
string  
Opção de atributo padrão (true = Sim / false = Não).  
false  
codref  
string  
Código de referência do atributo.  
ABC566

stock  
number  
Quantidade em estoque desta variação do produto.  
0  
priceDiff  
string  
Alteração de preço (Aceito: add/rem/fixed).  
NULL  
price  
number  
Valor do preço a ser alterado conforme campo priceDiff.  
0  
weightDiff  
string  
Alteração de peso (Aceito: add/rem/fixed).  
NULL  
weight  
number  
Valor do peso a ser alterado conforme campo priceDiff.  
0  
status  
number  
Status do atributo (1 = Ativo / 0 = Inativo).  
0  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
//   <groupId>org.glassfish.jersey.core</groupId>  
//   <artifactId>jersey-client</artifactId>  
//   <version>2.8</version>  
// </dependency>  
// <dependency>  
//   <groupId>org.glassfish.jersey.media</groupId>  
//   <artifactId>jersey-media-json-jackson</artifactId>  
//   <version>2.8</version>  
// </dependency>  
  
import javax.ws.rs.client.Client;  
import javax.ws.rs.client.ClientBuilder;  
import javax.ws.rs.client.Entity;  
import javax.ws.rs.core.Response;  
import javax.ws.rs.core.MediaType;  
  
Client client = ClientBuilder.newClient();

```
Entity<String> payload = Entity.text("[ { \"id\": 5216873, \"isDefault\": true, \"weight\": 1.342, \"weightDiff\": \"add\" }, { \"id\": 5216874, \"isDefault\": false, \"stock\": 10, \"status\": 1 }]");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/product/{id}/attributes/update")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "productId": 22635,
  "recordsChanged": [
    5216873,
    5216874
  ]
}
```

Atualizar atributos de um produto.  
 Consulte os códigos das opções para a parametrização correta.

## Controle de Estoque

Consulta/Atualização de estoque de produtos ou atributos em massa ou individual.  
 A atualização de estoque através da API não efetua (por enquanto) a notificação para os clientes que desejaram receber o aviso de disponibilidade e não atualiza o estoque no Mercado Livre caso a loja tenha a integração ativa.

Consultar

```
GEThttp://www.seudominio.com.br/ws/v1/product/stock
```

## Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/stock")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .header("products", "17297;10807")
    .get();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "result": [
```

```

{
  "id": 17297,
  "name": "Boné New York Snapback White/Black",
  "quantity": 90,
  "checkStock": "product"
},
{
  "id": 10807,
  "name": "Camiseta Vans Nofiler Navy",
  "quantity": 0,
  "checkStock": "attributes",
  "attributes": [
    {
      "status": true,
      "id": 1607860,
      "cod_ref": "",
      "quantity": 1
    },
    {
      "status": false,
      "id": 1607861,
      "cod_ref": "",
      "quantity": 0
    }
  ]
}
]
}

```

#### Required Header Parameters

|              |   |
|--------------|---|
| access-token | Chave de acesso obrigatória para a requisição   |
| products     | ID de cadastro dos produtos (separados por ponto e virgual) a serem consultados.<br>Formato esperado: 123;456;... |

Atualizar

POST <http://www.seudominio.com.br/ws/v1/product/stock>



## Request

### ATTRIBUTES

```
products
required
array
Production
Java
Try
// Maven : Add these dependencies to your pom.xml (java6+)
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"products\": [ {   \"type\": \"product\",   \"id\": 17297,   \"quantity\": 10 }, {   \"type\": \"attribute\",   \"id\": 1607860,   \"quantity\": 5 } ]}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/stock")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
```

```
"status": 400,
"message": "Error validation message..."
}
```

## Response

200

### HEADERS

Content-Type:application/json

### BODY

```
{
  "status": 200,
  "recordsChanged": 2
}
```

---

### Required Parameters

|          |   |
|----------|---|
| products | Objeto do tipo array contendo os produtos e/ou atributos a serem atualizados. |
|----------|---|

## Atualização de Preço

Atualização de preço dos produtos em massa ou individual.

POST

POST <http://www.seudominio.com.br/ws/v1/product/price>

## Request

### ATTRIBUTES

products

required

array

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

```
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"products\": [ {   \"id\": 17297,   \"new_price\": 103.9   }, {   \"id\": 10807,   \"new_price\": 149.8   } ]}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/price")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "recordsChanged": 2
}
```

---

## Required Parameters

---

|          |  |
|----------|--|
| products | Objeto do tipo array contendo os produtos a serem atualizados. |
|----------|--|

## Promoções

### POST

POST <http://www.seudominio.com.br/ws/v1/product/promo>

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

//   <groupId>org.glassfish.jersey.core</groupId>

//   <artifactId>jersey-client</artifactId>

//   <version>2.8</version>

// </dependency>

// <dependency>

//   <groupId>org.glassfish.jersey.media</groupId>

//   <artifactId>jersey-media-json-jackson</artifactId>

//   <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Entity<String> payload = Entity.text("");
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/product/promo")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .post(payload);
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "result": ""
}
```

Not implemented yet! :D

## Atributos/Variações

Gerenciamento e consulta de atributos / variações da loja.

### Grupos

Obtem os grupos de atributos cadastrados.

/attributes/groups

### Atributos de um Grupo

Obtem todas as informações relacionados a um grupo de atributos com o ID informado.

/attributes/{id}

## Cadastro

Required Parameters

|           |               |
|-----------|---------------|
| groupName | Nome do grupo |
|-----------|---------------|

|         |  |  |
|---------|--|--|
| options | Array de objetos contendo as opções de atributos do grupo. |  |
|         | type   | Tipo de seleção das opções.<br>Aceito: radio, dropdown, swatch ou texture. |
|         | name   | Nome da opção de seleção.<br>Exemplo: Cor                                  |
|         | value  | valor da opção de seleção.<br>Exemplo: Vermelho                            |
|         | optionSort   | Ordem de exibição da opção de atributo.                                    |
|         | valueSort  | Ordem de exibição do valor da opção do atributo.                           |

Inserir

POST <http://www.seudominio.com.br/ws/v1/attributes/insert>

## Request

### ATTRIBUTES

groupName

required

string

Nome do grupo a ser cadastrado.

Teste de novo grupo

options

required

array

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

//   <groupId>org.glassfish.jersey.core</groupId>

//   <artifactId>jersey-client</artifactId>

//   <version>2.8</version>

```
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"groupName\": \"Teste de novo grupo\", \"options\": [
{   \"type\": \"swatch\",   \"name\": \"Cor\",   \"value\": \"#be0000\",   \"optionSort\": 1,
\"valueSort\": 1  }, {   \"type\": \"swatch\",   \"name\": \"Cor\",   \"value\": \"#000000\",
\"optionSort\": 1,   \"valueSort\": 2  }, {   \"type\": \"radio\",   \"name\": \"Tamanho\",
\"value\": \"M\",   \"optionSort\": 2,   \"valueSort\": 1  }, {   \"type\": \"radio\",
\"name\": \"Tamanho\",   \"value\": \"G\",   \"optionSort\": 2,   \"valueSort\": 2  } ]}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/attributes/insert")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
  "groupId": 22,
  "groupName": "Teste de novo grupo",
  "options": [
    {
      "id": 246,
      "name": "Cor",
      "value": "#be0000",
      "optionSort": "1",
      "valueSort": "1"
    },
    {
```

```
{
  "id": 247,
  "name": "Cor",
  "value": "#000000",
  "optionSort": "1",
  "valueSort": "2"
},
{
  "id": 248,
  "name": "Tamanho",
  "value": "M",
  "optionSort": "2",
  "valueSort": "1"
},
{
  "id": 249,
  "name": "Tamanho",
  "value": "G",
  "optionSort": "2",
  "valueSort": "2"
}
]
```

## Atualizar

POST <http://www.seudominio.com.br/ws/v1/attributes/update/id>

## Parameters

id  
ID do grupo de atributos a ser atualizado. Example: 22.  
Number

## Request

### ATTRIBUTES

groupName  
required  
string  
Nome do grupo a ser cadastrado.  
Grupo alterado  
options  
required  
array  
Production



Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.core</groupId>
```

```
//   <artifactId>jersey-client</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.media</groupId>
```

```
//   <artifactId>jersey-media-json-jackson</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Entity<String> payload = Entity.text("{ \"groupName\": \"Grupo alterado\", \"options\": [ {  
  \"type\": \"swatch\",    \"name\": \"Cor\",    \"value\": \"#be0000\",    \"optionSort\": 1,  
  \"valueSort\": 1  }, {    \"type\": \"swatch\",    \"name\": \"Cor\",    \"value\": \"#000000\",  
  \"optionSort\": 1,    \"valueSort\": 2  } ]}");
```

```
Response response =
```

```
client.target("http://www.seudominio.com.br/ws/v1/attributes/update/{id}")
```

```
  .request(MediaType.TEXT_PLAIN_TYPE)
```

```
  .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
  .post(payload);
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{  
  "status": 200,  
  "groupId": 21,  
  "newGroupName": "Alterado grupo",  
  "newOptions": [  
    {  
      "id": 250,
```

```

        "name": "Cor",
        "value": "#be0000",
        "optionSort": "1",
        "valueSort": "1"
    },
    {
        "id": 251,
        "name": "Cor",
        "value": "#000000",
        "optionSort": "1",
        "valueSort": "2"
    }
]
}

```

Ao efetuar a atualização de um grupo, todos os atributos nele contidos são removidos e inserido os novos.

## Fabricantes/Fornecedores

Consulta/Cadastro de fornecedores/fabricantes dos produtos.

### Consultar

Consultar

GET <http://www.seudominio.com.br/ws/v1/manufacturer/list>

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Response response = client.target("http://www.seudominio.com.br/ws/v1/manufacturer/list")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .get();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
400
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 400,
  "message": "Error validation message..."
}
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "response": [
    {
      "id": 136,
      "name": "Fornecedor 1",
      "date_added": "2015-10-17 12:56:08",
      "last_modified": null
    },
    {
      "id": 164,
      "name": "Fornecedor 2",
      "date_added": "2016-02-27 15:21:51",
      "last_modified": null
    }
  ]
}
```

```
]
}
```

## Cadastro

Inserir

POST <http://www.seudominio.com.br/ws/v1/manufacturer/insert>

### Request

#### ATTRIBUTES

name

required

string

Nome do fornecedor a ser cadastrado.

Fornecedor 3

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Entity<String> payload = Entity.text("{
```

```
    \"name\": \"Fornecedor 3\",
```

```
}    \");
```

```
Response response =
```

```
client.target("http://www.seudominio.com.br/ws/v1/manufacturer/insert")
```

```
.request(MediaType.TEXT_PLAIN_TYPE)
.header("access-token", "1a1b2c3d4e5f6g7h...")
.post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "manufacturerId": 189
}
```

## Atualizar

POST <http://www.seudominio.com.br/ws/v1/manufacturer/update/id>

## Parameters

id  
ID de cadastro do fornecedor para atualização. Example: 189.  
Number

## Request

### ATTRIBUTES

name  
required  
string  
Nome do fornecedor a ser alterado.  
Fornecedor 3.1  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
// <groupId>org.glassfish.jersey.core</groupId>  
// <artifactId>jersey-client</artifactId>  
// <version>2.8</version>

```
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"name\": \"Fornecedor 3.1\\\"}");
Response response =
client.target("http://www.seudominio.com.br/ws/v1/manufacturer/update/{id}")
  .request(MediaType.TEXT_PLAIN_TYPE)
  .header("access-token", "1a1b2c3d4e5f6g7h...")
  .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "manufacturerId": 189
}
```

## Pedidos

### Novo pedido

Função de cadastro de um novo pedido simples.

A inserção de pedido através da API não efetua (por enquanto) a notificação para o cliente, não contabiliza os pontos pelo Programa de Pontos caso a loja tenha o recurso ativo, caso o pedido esteja relacionado e não realiza a sincronização de estoque no Mercado Livre.

Para obter o código do cliente, consulte a categoria Clientes desta documentação.

---

## Required Parameters

---

|                       |   |
|-----------------------|---|
| status                | ID de cadastro do status a ser definido para o pedido.<br>OU valores predefinidos: <ul style="list-style-type: none"><li>• pending - Pendente/Aguardando</li><li>• paid - Pago</li><li>• shipped - Enviado</li><li>• finished - Finalizado</li><li>• canceled - Cancelado</li></ul> |
| orderTotal            | Valor total do pedido.  |
| customerId            | Código de cadastro de cliente do pedido.  |
| shipping[method]      | Método de entrega do pedido. (consulte as formas de entrega disponíveis).   |
| shipping[cost]        | Valor do custo de frete do pedido.  |
| shipping[address]     | Endereço de entrega do pedido.  |
| payment[method]       | Método de pagamento do pedido (consulte as formas de pagamento disponíveis).  |
| payment[installments] | Número de parcelas do pagamento.<br>Padrão: 1   |
| products              | Produtos do pedido.   |

## Optional Parameters

---

|             |   |
|-------------|---|
| affiliateId | ID de cadastro do parceiro.<br>OU valores predefinidos: <ul style="list-style-type: none"><li>• mobile</li><li>• store</li><li>• marketplace</li><li>• callcenter</li></ul> |
|-------------|---|

---

|                 |  |
|-----------------|--|
| ipAddress       | Endereço IP de realização do pedido.               |
| datePurchased   | Data de realização do pedido.                      |
| note            | Alguma observação adicional sobre o pedido.        |
| shipping[title] | Descrição opcional do método de entrega do pedido. |

## PUT

PUT <http://www.seudominio.com.br/ws/v1/order/insert>

## Request

### ATTRIBUTES

affiliateId

number

ID de parceiro ou origem do pedido.

123

status

required

string

ID de status do pedido OU Tag pré-definida.

pending

ipAddress

string

Endereço de IP de origem do pedido.

127.0.0.1

datePurchased

string

Data de realização do pedido.

2016-06-20 12:11:10

note

string

Alguma observação adicional sobre o pedido.

enviar nota fiscal para outronome@gmail.com

orderTotal

required

number

Valor total do pedido.

123.45

customerId



required

number

Código do cliente previamente cadastrado.

789

shipping

required

object

payment

required

object

nfe

object

products

required

array

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

import javax.ws.rs.client.Client;

import javax.ws.rs.client.ClientBuilder;

import javax.ws.rs.client.Entity;

import javax.ws.rs.core.Response;

import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();

Entity<String> payload = Entity.text("{ \"affiliateId\": 0, \"status\": 1, \"ipAddress\":

\"127.0.0.1\", \"datePurchased\": \"2016-06-20 12:11:10\", \"orderTotal\": 123.45,

\"customer\_id\": 789, \"shipping\": { \"method\": \"sedex\", \"scheduleDate\": \"\",

\"cost\": 12.78, \"address\": { \"firstname\": \"primeiro nome\", \"lastname\": \"segundo

nome\", \"postcode\": \"12345-678\", \"street\": \"endereço 1\", \"number\": 123,

\"complement\": \"apto 3030\", \"suburb\": \"bairro\", \"city\": \"cidade\", \"state\":

\"MG\" } }, \"payment\": { \"method\": \"boleto\", \"installments\": 1 }, \"products\": [ {

\"id\": 123, \"quantity\": 1, \"unitPrice\": 123.45, \"finalPrice\": 123.45, \"giftWrap\":

false, \"isFree\": false }, { \"id\": 456, \"quantity\": 2, \"unitPrice\": 100.3,

\"finalPrice\": 200.6, \"giftWrap\": false, \"isFree\": false } ]}");

Response response = client.target("http://www.seudominio.com.br/ws/v1/order/insert")

```
.request(MediaType.TEXT_PLAIN_TYPE)
.header("access-token", "1a1b2c3d4e5f6g7h...")
.put(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "orderId": 4561
}
```

## Pesquisa/Listagem

POST

POSThttp://www.seudominio.com.br/ws/v1/order/list

## Request

### ATTRIBUTES

```
query
string
Termo de busca para filtro do resultado.
cliente tal
date[from]
string
Data inicio de realização dos pedidos.
2016-06-01
date[to]
string
Data final de realização dos pedidos.
null
status
number
Status dos pedidos para filtro.
1
```

shipping  
string  
Forma de entrega.  
null  
payment  
string  
Forma de pagamento.  
null  
origin  
string  
Origem do pedido.  
null  
complete  
boolean  
Pedido finalizado (concluido).  
true  
paid  
boolean  
Pedido pago.  
true  
order  
string  
Coluna para ordenação do resultado.  
orders\_id  
order\_type  
string  
Tipo de ordenação da coluna de ordem definida.  
desc  
limit  
number  
Limite do resultado da consulta.  
2  
Production  
Java  
Try  
// Maven : Add these dependencies to your pom.xml (java6+)  
// <dependency>  
// <groupId>org.glassfish.jersey.core</groupId>  
// <artifactId>jersey-client</artifactId>  
// <version>2.8</version>  
// </dependency>  
// <dependency>  
// <groupId>org.glassfish.jersey.media</groupId>  
// <artifactId>jersey-media-json-jackson</artifactId>  
// <version>2.8</version>  
// </dependency>  
  
import javax.ws.rs.client.Client;

```

import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"date\": { \"from\": \"2017-01-01\" }, \"order_type\": \"desc\", \"complete\": true, \"limit\": 2}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/order/list")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));

```

## Response

200

HEADERS

Content-Type:application/json

BODY

```

{
  "status": 200,
  "limit": "2",
  "ordersFound": 30,
  "ordersTotal": 6474.89,
  "orders": [
    {
      "orderId": 31,
      "orderTotal": 377.15,
      "orderTotalPaid": 0,
      "statusId": 1,
      "orderIsComplete": true,
      "datePurchased": "2016-06-13 17:06:42",
      "lastModified": "2016-06-14 16:47:58",
      "datePaid": null,
      "currency": "BR",
      "currencyValue": 1,
      "affiliateId": 0,
      "clientIP": "127.0.0.1",
      "note": "nota interna sobre o pedido",
      "shipping": {
        "method": "sedex",
        "title": "",
        "date": "2016-06-13",
        "time": "11:00 até 12:00",

```

```

        "track_id": null
    },
    "payment": {
        "method": "mercadopago_boleto",
        "title": "",
        "installments": 1
    },
    "nfe":{
        "number": 00000090,
        "key": "",
        "series": 001,
        "date": null,
        "link": ""
    },
    "customer": {
        "id": 11,
        "company": "",
        "name": "Cliente com nome",
        "first_name": "Primero nome",
        "last_name": "Sobrenome",
        "email": "emailcliente@dominio.com.br"
    }
},
{
    "orderId": 30,
    "orderTotal": 365.95,
    "orderTotalPaid": 365.95,
    "statusId": 1,
    "orderIsComplete": true,
    "datePurchased": "2019-06-19 15:51:11",
    "lastModified": null,
    'datePaid': "2019-06-20 13:20:10",
    "currency": "BR",
    "currencyValue": 1,
    "affiliateId": 0,
    "clientIP": "127.0.0.1",
    "note": null,
    "shipping": {
        "method": "terrestre",
        "title": null,
        "date": null,
        "time": null,
        "track_id": null
    },
    "payment": {
        "method": "boleto",
        "title": null,
        "installments": 1
    }
}

```

```

    },
    "nfe":{
      "number": null,
      "key": null,
      "series": null,
      "date": null,
      "link": null
    },
    "customer": {
      "id": 11,
      "company": "",
      "name": "Cliente com nome",
      "first_name": "Primero nome",
      "last_name": "Sobrenome",
      "email": "emailcliente@dominio.com.br"
    }
  }
]
}

```

Os pedidos podem ser paginados definindo 10,20 para o parâmetro limit obtendo assim os registros da posição 10 até a posição 20 conforme busca.

O limite padrão de retorno é de 20 pedidos. O total de pedidos encontrados é retornado na coluna ordersFound e a totalização dos mesmo na coluna ordersTotal.

Você pode utilizar os seguintes parâmetros no post da requisição:

#### Required Parameters

É requerido que pelo menos 1 parametro de filtro seja enviado no corpo da requisição.

#### Optional Parameters

query

Termo de busca para filtro da listagem.  
 Pode ser enviado: número do pedido, nome de cliente, email, empresa, cidade, cep, telefone...  
 Para pesquisar vários códigos de pedidos, separe-os por virgula: 1234,5678,9012...

|                     |   |
|---------------------|---|
| date[from]          | Data inicio do range de busca.<br>Pode ser usado para busca de pedidos a partir da data YYYY-MM-DD.<br>Formato: YYYY-MM-DD  |
| date[to]            | Data final do range de busca.<br>Quando utilizado, se faz necessário o envio também da data de inicio.<br>Formato: YYYY-MM-DD   |
| date_modified[from] | Data inicio do range de busca pela data de modificação do pedido.<br>Pode ser usado para busca de pedidos modificados a partir da data YYYY-MM-DD.<br>Formato: YYYY-MM-DD |
| date_modified[to]   | Data final do range de busca pela data de modificação do pedido.<br>Quando utilizado, se faz necessário o envio também da data de inicio.<br>Formato: YYYY-MM-DD          |
| status              | Status de pedido.<br>Para obter a lista de código de cada status, verifique o método <a href="#">/order/status/list</a>   |
| shipping            | Forma de entrega.   |
| payment             | Forma de pagamento.   |
| origin              | Origem do pedido.   |

|            |  |
|------------|--|
| complete   | Pedido finalizado (concluído).   |
| paid       | Pedido pago (definido como pago).<br>Valores aceitos: true   false   |
| order      | Coluna de ordenação da listagem.<br>Valor padrão: orders_id<br>Colunas aceitas: orders_id, customers_name, customers_email |
| order_type | Tipo de ordenação da coluna definida.<br>Valor padrão: asc<br>Valores aceitos: asc (crescente) ou desc (decrescente).      |
| limit      | Limite do resultado da consulta.<br>Padrão: 50<br>Exemplo: 10 ou "10,50" para obter registros da posição 10 até a 50.      |

## Por ID

GET

GET <http://www.seudominio.com.br/ws/v1/order/id>

### Parameters

id  
ID do pedido a ser consultado. Example: 12345.  
Number

### Request

Production



Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.core</groupId>
```

```
//   <artifactId>jersey-client</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
// <dependency>
```

```
//   <groupId>org.glassfish.jersey.media</groupId>
```

```
//   <artifactId>jersey-media-json-jackson</artifactId>
```

```
//   <version>2.8</version>
```

```
// </dependency>
```

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.client.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Response response = client.target("http://www.seudominio.com.br/ws/v1/order/{id}")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .get();
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
```

```
  "status": 200,
```

```
  "orderId": 31,
```

```
  "orderIsComplete": true,
```

```
  "totalValue": 377.15,
```

```
  "order": {
```

```
    "statusId": 1,
```

```
    "statusDescription": "Pendente",
```

```
    "datePurchased": "2016-06-17 17:06:42",
```

```
    "lastModified": "2016-06-17 16:47:58",
```

```
    "currency": "BR",
```

```
    "currencyValue": 1,
```

```
"affiliateId": 0,
"note": "nota de aviso interno sobre o pedido",
"shipping": {
  "method": "sedex",
  "title": "SEDEX",
  "date": "2016-06-13",
  "time": "11:00 até 12:00",
  "cost": 24.4,
  "track_id": null,
  "address": {
    "name": "Jota De Paula",
    "firstname": "Jota",
    "lastname": "De Paula",
    "company": "",
    "street_address": "Rua Monteiro Lobato",
    "street_number": 237,
    "street_complemento": "",
    "suburb": "Ouro Preto",
    "city": "Belo Horizonte",
    "postcode": "31310-530",
    "state": "Minas Gerais",
    "state_code": "MG",
    "country": "Brasil",
    "format_id": "1"
  }
},
"payment": {
  "method": "mercadopago_boleto",
  "title": "",
  "installments": 1
},
"nfe": {
  "number": null,
  "key": null,
  "series": null,
  "date": null,
  "link": ""
},
"customer": {
  "id": 12,
  "email": "tarcisio@iset.com.br",
  "type": "F",
  "company": "",
  "cpf_cnpj": "213.731.351-07",
  "rg": "",
  "name": "Jota De Paula",
  "first_name": "Jota",
  "last_name": "De Paula",
```

```
"dob": null,
"address": {
  "postcode": "31310-530",
  "street": "Rua Monteiro Lobato",
  "number": 237,
  "complement": "",
  "suburb": "Ouro Preto",
  "city": "Belo Horizonte",
  "state_code": "MG",
  "state": "Minas Gerais",
  "country": "Brasil"
},
"phones": {
  "primary": {
    "ddd": 31,
    "number": "84010710"
  },
  "cellphone": {
    "ddd": 0,
    "number": ""
  }
}
},
"totals": [
  {
    "code": "ot_subtotal",
    "title": "Sub-total",
    "text": "R$ 383,76",
    "value": 383.76,
    "value_action": 383.76,
    "action": "",
    "sort_order": 1
  },
  {
    "code": "ot_rules",
    "title": "Regra 3",
    "text": "- R$ 1,00",
    "value": 1,
    "value_action": 1,
    "action": "rem_v",
    "sort_order": 5
  },
  {
    "code": "ot_operacoes",
    "title": "Desconto em % de",
    "text": "-&nbsp;10,00%",
    "value": 10,
    "value_action": 10,
```

```
"action": "rem_p",
"sort_order": 7
},
{
  "code": "ot_shipping",
  "title": "SEDEX",
  "text": "+&nbsp;R$ 24,40",
  "value": 24.4,
  "value_action": 24.4,
  "action": "add_v",
  "sort_order": 8
},
{
  "code": "ot_total",
  "title": "<b>Valor total do pedido</b>",
  "text": "<span class='txt_price'>R$ 377,15</span><div class='forma_pagto'><span id='max_p_s_juros'>em até <b>5x</b> de</span> <span id='vl_parcela_s_juros'><b>R$ 77,12 sem juros</b>&nbsp;ou</span>\r\n          <span>em até <b>24x</b></span> de <span><b>R$ 21,37",
  "value": 377.15,
  "value_action": 377.15,
  "action": "",
  "sort_order": 11
}
],
"products": [
  {
    "id": 15084,
    "name": "Boné Starter Snapback Los Angeles City Black",
    "cod_ref": "",
    "model": "Los Angeles City",
    "quantity": 2,
    "price_unit": 101.94,
    "price_final": 101.94,
    "weight": 1.2,
    "dimensions": {
      "width": 11,
      "height": 2,
      "length": 16
    },
    "configuration": "",
    "attributes": null,
    "mercadolive_id": "MLB724754022",
    "related_categories": [
      174,
      279,
      471,
      479
    ]
  }
]
```

```
]
},
{
  "id": 16083,
  "name": "Boné Starter Snapback São Paulo Blue/White/Navy",
  "cod_ref": "5055663506036",
  "model": "São Paulo",
  "quantity": 1,
  "price_unit": 89.94,
  "price_final": 89.94,
  "weight": 1,
  "dimensions": {
    "width": 11,
    "height": 2,
    "length": 16
  },
  "configuration": "",
  "attributes": null,
  "mercadolive_id": "MLB724758982",
  "related_categories": [
    174,
    279,
    471,
    479
  ]
},
{
  "id": 16090,
  "name": "Boné Starter Snapback Spow Grey/White/Black",
  "cod_ref": "5055663506043",
  "model": "Spow",
  "quantity": 1,
  "price_unit": 89.94,
  "price_final": 89.94,
  "weight": 1,
  "dimensions": {
    "width": 11,
    "height": 2,
    "length": 16
  },
  "configuration": "",
  "attributes": null,
  "mercadolive_id": "MLB724759052",
  "related_categories": [
    174,
    279,
    471,
    479
  ]
}
```

```
]
}
```

Obtem todas as informações relacionados a um pedido de acordo com o ID informado.

## Andamentos

Gerenciamento e consulta de andamentos de um pedido.

PUT

PUT<http://www.seudominio.com.br/ws/v1/order/id/status>

## Request

### ATTRIBUTES

status\_id

required

string

ID de cadastro do status a ser definido para o pedido.

123

track\_code

string

Código de rastreamento, caso tenha.

SS123456789BR

comments

string

Comentários/Informações a respeito deste novo status.

Seu pedido foi enviado!

isViewable

boolean

O cliente poderá visualizar o status.

true

nfe

object

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

```
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{ \"status_id\": 123, \"track_code\": \"SS123456789BR\", \"comments\": \"Seu pedido foi enviado!\", \"isViewable\": true}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/order/id/status")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .put(payload);

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
        Content-Type:application/json
BODY
{
  "status": 200,
  "orderId": 4561
}
```

---

### Required Parameters

---

|           |   |
|-----------|---|
| status_id | ID de cadastro do status a ser definido para o pedido.<br>OU valores predefinidos: <ul style="list-style-type: none"> <li>• pending - Pendente/Aguardando</li> <li>• paid - Pago</li> <li>• invoice - Faturado</li> <li>• shipped - Enviado</li> <li>• finished - Finalizado</li> <li>• canceled - Cancelado</li> </ul> |
|-----------|---|

#### Optional Parameters

|            |   |
|------------|---|
| track_code | Código de rastreamento, caso tenha.                   |
| nfe        | Dados da Nota Fiscal do pedido.                       |
| comments   | Comentários/Informações a respeito deste novo status. |
| isViewable | O cliente poderá visualizar o status.                 |

A atualização de status do pedido através da API não efetua (por enquanto) a notificação para o cliente, não contabiliza os pontos pelo Programa de Pontos caso a loja tenha o recurso ativo e não realiza a atualização da lista personalizada, caso o pedido esteja relacionado.

## Status

### Status de Pedidos

Lista de status da loja para gerenciamento de pedidos.

GET

GET <http://www.seudominio.com.br/ws/v1/order/status/list>

### Request

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)



```
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Response response = client.target("http://www.seudominio.com.br/ws/v1/order/status/list")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .get();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
    Content-Type:application/json
BODY
{
  "status": 200,
  "result": [
    {
      "id": 1,
      "title": "Pendente",
      "color": "#FFFFCC"
    },
    {
      "id": 2,
      "title": "Processando",
      "color": "#0099FF"
    }
  ]
}
```

O status são totalmente gerenciados pelo Lojista, será necessário obter esta lista para utilização dos IDs de cadastro nos gerenciamentos de pedidos.

## Webhooks

### Utilizando Webhooks

Webhooks permitem que sistemas externos recebam notificações de todos os eventos que ocorrem no sistema. Quando um evento acontece, o sistema envia uma requisição HTTP POST para a URL configurada no webhook com as informações relativas ao evento. Ao receber a notificação, o sistema pode executar diversas tarefas, dependendo do objetivo da integração.

Os webhooks também são chamados de Callbacks ou Reverse API.

### Configurando Webhooks

É recomendável que o endereço que receberá a requisição HTTP POST de notificação do evento seja um endereço que use SSL, ou seja, que inicie com `https://`.

Você pode configurar os webhooks em Configurações > API Rest > Webhooks.

### Pedidos

Serão enviadas todas as alterações e criação de novos pedidos, seja pelo painel administrativo (manual) ou pelo catálogo.

- Response 200 (application/json)

```
{
  "type": "order-changed",
  "orderId": 1112,
  "domain": "dominio.com",
  "statusId": 10,
  "status": "Aprovado"
}
```

- 

### Pedidos removidos

Será enviado notificações quando um pedido for removido pelo painel administrativo.

- Response 200 (application/json)

```
{
  "type": "order-removed",
  "orderId": 1112,
  "domain": "dominio.com"
}
```

-

## Produtos

Será enviado notificações quando um produto for criado ou alterado pelo painel administrativo.

- Response 200 (application/json)

```
{
  "type": "product-changed",
  "product_id": 2321,
  "sku": "ABC12345",
  "quantity": 10,
  "domain": "dominio.com"
}
```

•

## Produtos removidos

Será enviado notificações quando um produto for removido pelo painel administrativo.

- Response 200 (application/json)

```
{
  "type": "product-removed",
  "product_id": 4412,
  "domain": "dominio.com"
}
```

•

## Estoque

Serão enviadas todas alterações de estoque. Caso seja um atributo que tenha alterações de estoque, será enviado a coluna attr\_id com o ID do atributo alterado.

- Response 200 (application/json)

```
{
  "type": "product-stock-changed",
  "product_id": 4412,
  "sku": "ABC12345",
  "old_quantity": 10,
  "new_quantity": 8,
  "domain": "dominio.com"
}
```

•

## Categorias

Será enviado notificações quando uma categoria for criada ou alterada pelo painel administrativo.

- Response 200 (application/json)

```
{
  "type": "category-changed",
  "categoryId": 12,
  "domain": "dominio.com"
}
```

- 

## Categorias removidas

Será enviado notificações quando um pedido for removido pelo painel administrativo.

- Response 200 (application/json)

```
{
  "type": "category-removed",
  "categoryId": 13,
  "domain": "dominio.com"
}
```

## Gerenciando WebHook

Você poderá adicionar e remover webhooks via API utilizando os métodos abaixo.

Caso já exista a URL a ser cadastrada, nenhuma ação será feita e o retorno será sempre de sucesso.

### Required Parameters

|        |   |
|--------|---|
| type   | Tipo de webhook a ser cadastrado: <ul style="list-style-type: none"> <li>• orders - Pedidos</li> <li>• products - Produtos</li> <li>• categories - Categorias</li> </ul>  |
| action | Tipo de ação conforme o tipo definido: <ul style="list-style-type: none"> <li>• change - Ao criar/alterar determinado registro.</li> <li>• remove - Ao remover um registro.</li> <li>• stock - Ao alterar estoque de determinado produto. (Válido apenas para o tipo "product").</li> </ul> |
| url    | URL de destino da notificação a ser enviada.  |

Inserir

POST <http://www.seudominio.com.br/ws/v1/webhook>

## Request

### ATTRIBUTES

type

required

string

Tipo de webhook

"orders"

```
action
required
string
Ação do webhook
"change"
url
required
string
URL de notificação do webhook
"https://www.dominio..."
Production
Java
Try
// Maven : Add these dependencies to your pom.xml (java6+)
// <dependency>
//   <groupId>org.glassfish.jersey.core</groupId>
//   <artifactId>jersey-client</artifactId>
//   <version>2.8</version>
// </dependency>
// <dependency>
//   <groupId>org.glassfish.jersey.media</groupId>
//   <artifactId>jersey-media-json-jackson</artifactId>
//   <version>2.8</version>
// </dependency>
```

```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
Entity<String> payload = Entity.text("{
    \"type\": \"orders\",
    \"action\": \"change\",
    \"url\": \"https://www.dominio.com.br/erp/acao\",
}");
Response response = client.target("http://www.seudominio.com.br/ws/v1/webhook")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .post(payload);
```

```
System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
  "status": 200,
}
```

## Remover

DELETEhttp://www.seudominio.com.br/ws/v1/webhook

## Request

### ATTRIBUTES

type

required

string

Tipo de webhook

"orders"

action

required

string

Ação do webhook

"change"

url

required

string

URL de notificação do webhook

"https://www.dominio..."

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.MediaType;

Client client = ClientBuilder.newClient();
Response response = client.target("http://www.seudominio.com.br/ws/v1/webhook")
    .request(MediaType.TEXT_PLAIN_TYPE)
    .header("access-token", "1a1b2c3d4e5f6g7h...")
    .delete();

System.out.println("status: " + response.getStatus());
System.out.println("headers: " + response.getHeaders());
System.out.println("body:" + response.readEntity(String.class));
```

## Response

```
200
    HEADERS
        Content-Type:application/json
    BODY
    {
        "status": 200,
    }
```

# Marketing

## Mailing

Obtem todos os e-mails e/ou informações básicas de clientes.

POST

POSThttp://www.seudominio.com.br/ws/v1/mailling/{newsletter|customers})

## Request

### ATTRIBUTES

page  
number  
Número da página a ser obtida.  
0

perPage

number

Número máximo de registros a ser obtido por página (Valor máximo: 500).

100

Production

Java

Try

// Maven : Add these dependencies to your pom.xml (java6+)

// <dependency>

// <groupId>org.glassfish.jersey.core</groupId>

// <artifactId>jersey-client</artifactId>

// <version>2.8</version>

// </dependency>

// <dependency>

// <groupId>org.glassfish.jersey.media</groupId>

// <artifactId>jersey-media-json-jackson</artifactId>

// <version>2.8</version>

// </dependency>

```
import javax.ws.rs.client.Client;
```

```
import javax.ws.rs.client.ClientBuilder;
```

```
import javax.ws.rs.core.Entity;
```

```
import javax.ws.rs.core.Response;
```

```
import javax.ws.rs.core.MediaType;
```

```
Client client = ClientBuilder.newClient();
```

```
Entity<String> payload = Entity.text("{
```

```
    \"page\": 0,
```

```
    \"perPage\": 100,
```

```
}");
```

```
Response response =
```

```
client.target("http://www.seudominio.com.br/ws/v1/mailling/{newsletter|customers}")
```

```
    .request(MediaType.TEXT_PLAIN_TYPE)
```

```
    .header("access-token", "1a1b2c3d4e5f6g7h...")
```

```
    .post(payload);
```

```
System.out.println("status: " + response.getStatus());
```

```
System.out.println("headers: " + response.getHeaders());
```

```
System.out.println("body:" + response.readEntity(String.class));
```

## Response

200

HEADERS

Content-Type:application/json

BODY

```
{
```

```
    \"status\": 200,
```



```
"page": 0,  
"perPage": 100,  
"pages": 1,  
"totalFound": 2,  
"totalCount": 2,  
"result": [  
  {  
    "name": "Nome1",  
    "email": "email@site.com"  
  },  
  {  
    "name": "Nome2",  
    "email": "email@dominio.com"  
  },  
]  
}
```