Documentação
Master API

2020

www.alternativasistemas.com.br
(14) 3879-0580
Rua Elza Felippini, 4-49
Jd. Auri Verde, 17047-170, Bauru-SP

Sumário

1 APRESENTAÇÃO 3
1.1 O que é API? 3
1.2 O que é padrão REST? 3
2 MASTER API 4
2.1 Autorizações 4
2.2 Headers e Filtros 5
2.3 Códigos de Retorno 6
2.4 Erros 6
2.5 Grupos 7
2.6 Requisições de Fila 7
2.7 Campos Obrigatórios 8
2.8 Requisições 13
2.8.1 Categoria 14
2.8.2 Cliente 17
2.8.3 Fabricante 23
2.8.4 Forma de Pagamento 25
2.8.5 Forma de Parcelamento 26
2.8.6 Fornecedor 27
2.8.7 Ping 30
2.8.8 Rastreamento 30
2.8.9 Status do Pedido 32
2.8.10 Status Venda 35
2.8.11 Transportadora 36
2.8.12 Produto 37
2.8.13 Venda 49
2.8.14 Transferência 55
3 CASOS E EXEMPLOS 57

1 APRESENTAÇÃO

O objetivo desta documentação visa ensinar e preparar os clientes que irão consumir da API do Master, através das requisições HTTPs fornecidas.

1.1 O que é API?

Resumidamente, o significado da sigla API em inglês é *"**Application Programming Interface**"*, ou seja, é a Interface de programação de aplicações, normalmente utilizada para integrar sistemas, possibilitando inúmeros benefícios, como segurança dos dados e troca de informações por meio de diferentes linguagens de programação.

1.2 O que é padrão REST?

Os serviços da API do Master seguem o padrão REST (*"Representational State Transfer"*), ou seja, um estilo arquitetural que busca utilizar de maneira mais eficiente e em sua plenitude as características do protocolo HTTP, principalmente com sua semântica.

Levando em consideração essa parte da semântica, o protocolo HTTP disponibiliza métodos (verbos) que normalmente vêm associados as requisições, indicando com isso o que deve ser feito no servidor.

Por mais que existam inúmeros métodos, apenas três são utilizados na API do Master:

- **GET:** Indica um recurso que será recuperado do servidor. Exemplo: Solicitar do servidor as informações dos produtos cadastrados.
- **POST:** Indica que um recurso será inserido ou criado dentro do servidor. Exemplo: Inserir um novo produto no servidor.
- **PUT:** Indica que um recurso será atualizado no servidor. Exemplo: Atualização de preço do produto.

Para que consiga ocorrer essa "conversa", a API do Master utiliza o JSON *("Javascript Object Notation")* como forma de representação. Ele é somente uma forma leve para realizar a troca de informações entre seus clientes. Veja abaixo um exemplo de JSON:

```
{
    "Nome": Alternativa
    "Endereço": [
                "Rua":"Rua Elza Felipinni, 4-49"
                "Bairro":"Jardim Auri Verde"
                ]
}
```

Informaões entre chaves "{ }", são chamadas de objetos e entre "[ ]" são os arrays do JSON.

2 MASTER API

Neste capítulo vamos dividir melhor como é feita a integração com o Master, explicando separadamente o funcionamento de cada método e o que é necessário para cada requisição. Além de explicar sobre as mensagens de erro e seus retornos.

A API do Master pode ser encontrada em: http://alternativa.net.br/api/v2/

2.1 Autorizações

A API do Master conta com um recurso de autorização para utilizar de seus métodos, ou seja, é possível controlar quem consegue realizar as requisições por meio de uma chave de segurança, chamada de Token.

Esse Token pode ser gerado pela aplicação do Master, através do menu ***"Token da API".*** Nesse menu ficam registrados os Tokens gerados e quais seus valores. Veja Imagem Abaixo:

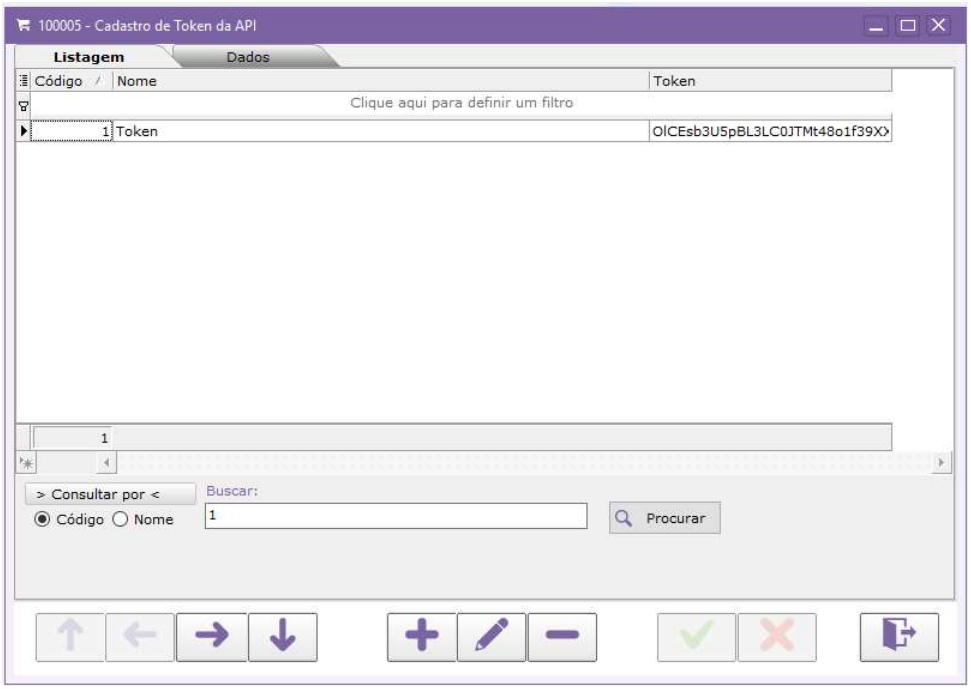


Figura: Tela de Tokens dentro do Master

Para realizar a requisição com esse Token, é necessário adicionar nos **Headers** o parâmetro **X-Token** que terá o Token que foi gerado no Master.

2.2 Headers e Filtros

Nas requisições, é possível adicionar headers para filtragem das informações, ou seja, direcionar principalmente as requisições que retornam grandes listas de objetos ou muita quantidade de dados.

Os possíveis filtros que podem ser colcoados:

- **Limit:** Tipo inteiro. É a quantidade de objetos que são retornados na fila
- **Offset:** Tipo inteiro. Índice de inicio da fila.
- **Count:** Tipo inteiro. Contador de totalidade de registros existentes
- **SearchBy:** Tipo string. Procura por determinado termo
- **DateOf:** Tipo Data. Data de Inicio da busca.
- **DateTo:** Tipo Data. Data limite da busca.
- **NaoPersistir:** Tipo booleano.

Requisições que possuem filtros mudam um pouco em sua resposta, ou seja, esses filtram são adicionados também no retorno da requisição, mostrando quais foram utilizados e quais seus valores. Veja exemplo de como um retorno comporta-se quando a requisição possui filtro:

```
"Filter": {
   "NaoPersistir": false,
   "Offset": 0,
   "Limit": 0,
   "Count": 0,
   "SearchBy": "",
   "DateOf": "Date",
   "DateTo": "Date"
}
```

2.3 Códigos de Retorno

Os códigos de retorno são os status da resposta que acompanham o protocolo HTTP e a API do Master padronizou esse status em suas requisições. Veja abaixo como ela foi definida:

Código	Status
200	OK – Busca, Inserção ou Update realizado
400 < ERRO < 500	ERRO – Estado inválido
500	ERRO – Erro Interno do Servidor

2.4 Erros

Os erros seguem um padrão de retorno, com duas informações básicas:

- **Código:** É o número de identificação do erro específico.
- **Mensagem:** Mensagem de erro associada ao código.

Exemplo de retorno de erro simples:

```
{
  "codigo": 0,
  "mensagem": "string"
}
```

Exemplo de retorno mais de um tipo de erro:

```
{
  "status": true,
  "erros": [
    {
      "codigo": 0,
      "mensagem": "string"
    }
  ]
}
```

2.5 Grupos

A API do Master está dividida em treze grupos que fazem requisições para determinada função dentro do servidor, esses grupos são:

- **Categoria**
- **Cliente**
- **Fabricante**
- **FormaPagamento**
- **FormaParcelamento**
- **Fornecedor**
- **Ping**
- **Produto**
- **Rastreamento**
- **StatusVenda**
- **Transportadora**
- **Venda**

Cada grupo desses possui um tipo de objeto único, que basicamente torna-se base das requisições, ou seja, são objetos específicos que determinam muitas vezes o corpo e as respostas das requisições.

2.6 Requisições de Fila

Os grupos: **categoria, cliente, fabricante, fornecedor, produto** e **venda** possuem métodos de fila de integração, ou seja, para serem inseridos ou alterados, são colocados primeiramente na fila, para depois serem consumidos pelo ERP, para isso existe algumas requisições padrões para enviar e retirar essas informações da fila. Veja como eles são montados abaixo:

1. **Método geral: POST /{grupo}/{código}/fila**

Descrição: Coloca na fila de integração.
Parâmetros: Token, grupo e código.
Response JSON Geral: **Status 200 – OK**

```
{
    Objeto especifico de cada grupo
}
```

Caso o status da requisição for 200, irá retornar o objeto especifico de cada grupo que foi adicionado à fila, ou seja, se for uma categoria, será um objeto do tipo categoria, caso seja uma venda, será um objeto do tipo venda e assim por diante.

2. **Método geral: POST /{grupo}/{código}/ok**

Descrição: Retira da fila de integração.
Parâmetros: Token, grupo e código.

Response JSON Geral: **Status 200 – OK**

```
{
    Objeto especifico de cada grupo
}
```

Caso o status da requisição for 200, irá retornar o objeto especifico de cada grupo que foi retirado da fila, ou seja, se for uma categoria, será um objeto do tipo categoria, caso seja uma venda, será um objeto do tipo venda e assim por diante.

2.7 Campos Obrigatórios

Algumas requisições precisam necessariamente de algumas informações para conseguirem executar sem problemas, caso algumas dessas informações não esteja disponível, a requisição retornará um código de erro. Para o melhor entendimento, organizamos tudo em uma tabela. Veja abaixo:

Tabela 1: Campos Obrigatórios/Erros

Categoria			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
String	*Nome*	1	Campo nome da categoria vazio
StatusVenda			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
String	*Nome*	1	Campo nome vazio
VendaPagamento			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
Double	*Valor*	1	Valor inválido
DateTime	*DataVencimento*	2	Data de vencimento inválida
Integer	*FormaPagamentoCodigo*	3	Forma de pagamento código inválida
VendaItem			
Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
String	*ItemNome*	1	Campo nome vazio
Double	*Quantidade*	2	Quantidade nula

Integer	*VendaCodigo*	3	Código da venda não pode ser igual a 0
Double	*PrecoUnitarioVenda*	4	Preço unitário da venda não pode ser igual a 0
Double	*ItemValorBruto*	6	Valor bruto do item não pode ser igual a 0
Double	*ItemValorLiquido*	7	Valor líquido do item não pode ser igual a 0
Integer	*ProdutoCodigo*	8	Código do produto não pode ser igual a 0
Venda			
Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
Campos em verde: caso não enviados, serão considerados pelo Código do Cliente			
Integer	*ClienteCodigo*	2	Código do cliente menor ou igual a 0
String	*ClienteNome*	3	Nome do cliente tem de ser preenchido
String(1)	*ClienteTipoPessoa*	4	Tipo do cliente tem de ser indicado. F: Físico, J: Jurídico, G: Governo, I: Imune ao ICMS.
String	*EntregaCEP*	9	CEP de entrega vazio
String	*EntregaUnidadeFederativa*	15	Unidade Federativa de entrega tem de ser preenchida
String	*EntregaNome*	11	Nome de entrega tem de ser preenchido
String	*EntregaLogradouro*	12	Logradouro de entrega tem de ser preenchido
String	*EntregaLogradouroNumero*	13	Número do logradouro de entrega tem de ser preenchido
String	*EntregaBairro*	14	Bairro de entrega tem de ser preenchido
Integer	*EntregaMunicipioCodigo*	16	Munícipio de entrega tem de ser preenchida
String	*EntregaTelefone*	17	Telefone de entrega tem de ser preenchida
String	*EntregaEmail*	18	Email de entrega tem de ser preenchida
Double	*ValorFrete*	19	Valor do frete inválido

Double	*ValorTotalBruto*	21	Valor total bruto inválido
Double	*ValorTotal*	22	Valor total inválido
DateTime	*DataVenda*	23	Data de venda inválida
Double	*ValorDesconto*	25	Valor de desconto da venda inválida
String(1)	*Origem*	26	Origem da venda inválido
String	*OrigemPedido*	27	Origem do pedido da venda inválido
String(1)	*TipoVenda*	28	Tipo da venda inválido
Boolean	*Reservada*	29	Reservada tem de ser preenchido.
ProdutoImagem			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
Integer	*ProdutoCodigo*	1	Campo código do produto vazio
Produto			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
String	*Nome*	1	Campo nome vazio
String	*EAN*	2	Campo EAN vazio
Double	*PrecoCusto*	3	Campo preço custo não pode ser 0
Double	*PrecoVenda*	4	Campo preço venda não pode ser 0
Integer	*TipoDisponibilidade*	5	Campo disponibilidade não pode ser 0
Integer	*QuantidadeMaximaPorCliente*	6	Campo quantidade máxima por cliente não pode ser 0
FormaParcelamento			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
String	*Nome*	1	Campo nome do parcelamento vazio
Fabricante			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
String	*Nome*	1	Campo nome vazio
String	*DocumentoFiscal*	2	CNPJ não pode ser vazio
String	*CEP*	3	CEP não pode ser vazio

String	*UnidadeFederativa*	4	Unidade Federativa não pode ser vazio
String	*Logradouro*	6	Logradouro não pode ser vazio
String	*Bairo*	7	Bairro não pode ser vazio
String	*Municipio*	8	Municipio não pode ser vazio
Parcela			
Tipo	Campos Obrigatórios	Código do Erro	Mensagem
Integer	*CodigoCliente*	1	Campo cliente código não pode ser menor ou igual a 0
Integer	*CodigoParcelamento*	2	Campo parcelamento código não pode ser menor ou igual a 0
Integer	*CodigoVenda*	3	Campo venda código não pode ser menor ou igual a 0
Integer	*CodigoFormaPagamento*	4	Campo forma pagamento código não pode ser menor ou igual a 0
Integer	*Numero*	5	Campo Numero não pode estar vazio
DateTime	*DataPagamento*	6	Campo Numero vencimento inválida ou vazia
Double	*Valor*	7	Campo valor não pode ser menor ou igual a 0
DateTime	*DataVencimento*	8	Campo data vencimento inválida ou vazia
Transportadora			
Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
String	*Nome*	1	Campo nome vazio
String	*DocumentoFiscal*	2	CNPJ não pode ser vazio
String	*CEP*	3	CEP não pode ser vazio
String	*UnidadeFederativa*	4	Unidade Federativa não pode ser vazio
String	*Logradouro*	6	Logradouro não pode ser vazio
String	*Bairro*	7	Bairro não pode ser vazio
String	*Municipio*	8	Municipio não pode ser vazio
ProdutoFornecedor			

Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
Integer	*FornecedorProdutoCodigo*	1	Campo código do fornecedor do produto vazio
Integer	*FornecedorCodigo*	2	Campo 12ornecedor código não pode ser igual a 0
Integer	*ProdutoCodigo*	3	Campo código do produto vazio
Cliente			
Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
String	*Nome*	2	Nome do cliente vazio
String(1)	*TipoPessoa*	3	Tipo pessoa do cliente diferente de F, J, G e I.
String	*DocumentoFiscal*	4	CNPJ do cliente não pode ser vazio
String	*CEP*	5	CEP não pode ser vazio
String	*UnidadeFederativa*	6	Unidade Federativa inválida
String	*Logradouro*	7	Logradouro não pode ser vazio
String	*Bairro*	8	Bairro não pode ser vazio
String	*Municipio*	9	Municipio não pode ser vazio
Fornecedor			
Tipo	Campos Obrigatórios	Codigo do Erro	Mensagem
String	*Nome*	1	Campo nome vazio
String	*DocumentoFiscal*	2	CNPJ não pode ser vazio
String	*CEP*	3	CEP não pode ser vazio
String	*UnidadeFederativa*	4	Unidade Federativa não pode ser vazio
String	*Logradouro*	6	Logradouro não pode ser vazio
String	*Bairro*	7	Bairro não pode ser vazio
String	*Municipio*	8	Municipio não pode ser vazio

2.8 Requisições e fluxo

Nesta área iremos abordar melhor como cada requisição funciona e exemplos de como elas são montadas, juntamente com os parâmetros necessários e o Status de retorno.

Antes de começar a dividir e explicar caso a caso vale salientar as **Unidades de Medidas** e os **Limites de caracteres** de alguns campos:

Unidades de medida principais:

Altura	Centímetros (cm)
Largura	Centímetros (cm)
Comprimento	Centímetros (cm)
Peso	Kilograma (kg)

Tabela 2: Unidades de Medidas usadas na API

Os **limites de caracteres** serão abordados dentro de cada método e quando não possuir nenhuma informação de limite é do tipo blob, ou seja, poderá enviar sem limite. Vale salientar que a API não irá retornar status de erro para tipos string em muitos casos, e irá cadastrar com o limite que lhe foi proposto, portanto apenas se o campo for obrigatório, o erro será visível.

2.8.1 Categoria

Esse grupo possui métodos para buscar, inserir e atualizar novas categorias de produtos dentro da API, colocando essas categorias em uma fila de integração.

1. **Método: GET /categoria/{código}**

Descrição: Busca Categoria no ERP e retorna-a se encontrá-la.
Parâmetros: Token e código.
Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "CodigoPai": 0,
  "Nome": "string(50)",
  "Descricao": "string(5000)",
  "Ordem": 0,
  "MetaKeywords": "string(500)",
  "Url": "string(100)",
  "Ativo": true,
  "VisualizarLogado": true,
  "MetaTitle": "string(200)",
  "MetaDescription": "string(200)"
}
```

Nesse caso é necessário informar qual o código da categoria que está sendo buscada na própria URL, o retorno é um Objeto JSON com as informações da categoria.

2. Método: GET /categoria

Descrição: Retorna Categorias da fila de integração

Parâmetros: Token apenas.

Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "CodigoPai": 0,
    "Nome": "string(50)",
    "Descricao": "string(5000)",
    "Ordem": 0,
    "MetaKeywords": "string(500)",
    "Url": "string(100)",
    "Ativo": true,
    "VisualizarLogado": true,
    "MetaTitle": "string(200)",
    "MetaDescription": "string(200)"
  }
]
```

Nesse caso só é necessário entrar com as informações do Token como autorização e o Retorno é um Array de Objetos JSON, com todas as categorias da fila de integrações.

3. **Método: PUT /categoria**

Descrição: Insere categoria

Parâmetros: Token e código (caso for atualizar categoria)

Body JSON:

```
 {
   "Codigo": 0,
   "CodigoPai": 0,
   "Nome": "string(50)",
   "Descricao": "string(500)",
   "Ordem": 0,
   "MetaKeywords": "string(500)",
   "Url": "string(100)",
   "Ativo": true,
   "VisualizarLogado": true,
   "MetaTitle": "string(200)",
   "MetaDescription": "string(200)"
}
```

Response JSON: **Status 200 – OK**

```
{
  "status": true,
  "erros": [
    {
      "codigo": 0,
      "mensagem": "string"
    }
  ]
}
```

Nesse caso irá retornar status 200 caso consiga inserir categoria, ou mensagem de erro caso não consiga.

2.8.2 Cliente

Esse grupo é o responsável por tratar as informações sobre os clientes, da mesma maneira que o grupo anterior, é possível consultar, inserir e atualizar informações de clientes.

1. **Método: GET /cliente**

Descrição: Retorna os clientes da fila da integração
Parâmetros: Token apenas
Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "NaoPersistir": true,
    "Nome": "string(60)",
    "TipoPessoa": "string(1)",
    "DataNascimento": "string",
    "Genero": "string(1)",
    "DocumentoFiscal": "string(15)",
    "Identidade": "string(20)",
    "RazaoSocial": "string(60)",
    "CEP": "string(9)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "Bairro": "string(60)",
    "LogradouroComplemento": "string(60)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Telefone": "string(9)",
    "TelefoneDDD": "string(3)",
    "Celular": "string(9)",
    "CelularDDD": "string(3)",
    "Email": "string(150)",
    "DataCadastro": "string",
    "ReceberEmail": true,
    "Observacoes": "string(70)",
    "PrecoValidade": 0,
    "MinimoVenda": 0,
    "Ativo": true,
    "TipoDestinatario": "string(1)",
    "DescontosPrazo": 0,
    "DescontoVista": 0,
    "MunicipioNome": "string(35)"
  }
]
```

Nesse caso será devolvido todos os clientes que estiverem na fila de integração, ou seja, um Array de Objetos JSON, necessitando apenas do Token de acesso para realizar a requisição.

2. Método: GET /cliente/{código}

Descrição: Retorna cliente através de seu código especifico.

Parâmetros: Token e código.

Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "Nome": "string(60)",
  "TipoPessoa": "string(1)",
  "DataNascimento": "string",
  "Genero": "string(1)",
  "DocumentoFiscal": "string(15)",
  "Identidade": "string(20)",
  "RazaoSocial": "string(60)",
  "CEP": "string(9)",
  "Logradouro": "string(185)",
  "LogradouroNumero": "string(60)",
  "Bairro": "string(60)",
  "LogradouroComplemento": "string(60)",
  "Municipio": "string(35)",
  "UnidadeFederativa": "string(2)",
  "Telefone": "string(9)",
  "TelefoneDDD": "string(3)",
  "Celular": "string(9)",
  "CelularDDD": "string(3)",
  "Email": "string(150)",
  "DataCadastro": "string",
  "ReceberEmail": true,
  "Observacoes": "string(70)",
  "PrecoValidade": 0,
  "MinimoVenda": 0,
  "Ativo": true,
  "TipoDestinatario": "string(1)",
  "DescontosPrazo": 0,
  "DescontoVista": 0,
  "MunicipioNome": "string(35)"
}
```

Nesse caso é retornado um cliente especifico e suas informações, sendo necessário apenas o Token e o código desse cliente.

3. **Método: PUT /cliente**

Descrição: Insere/Atualiza cliente

Parâmetros: Token e código caso for uma atualização.

Body JSON:

```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "Nome": "string(60)",
  "TipoPessoa": "string(1)",
  "DataNascimento": "string",
  "Genero": "string(1)",
  "DocumentoFiscal": "string(15)",
  "Identidade": "string(20)",
  "RazaoSocial": "string(60)",
  "CEP": "string(9)",
  "Logradouro": "string(185)",
  "LogradouroNumero": "string(60)",
  "Bairro": "string(60)",
  "LogradouroComplemento": "string(60)",
  "Municipio": "string(35)",
  "UnidadeFederativa": "string(2)",
  "Telefone": "string(9)",
  "TelefoneDDD": "string(3)",
  "Celular": "string(9)",
  "CelularDDD": "string(3)",
  "Email": "string(150)",
  "DataCadastro": "string",
  "ReceberEmail": true,
  "Observacoes": "string(70)",
  "PrecoValidade": 0,
  "MinimoVenda": 0,
  "Ativo": true,
  "TipoDestinatario": "string(1)",
  "DescontosPrazo": 0,
  "DescontoVista": 0,
  "MunicipioNome": "string(35)"
}
```

O JSON de resposta retornará status 200, com as novas informações do cliente inserido/atualizado.

Response JSON: **Status 200 – OK**


```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "Nome": "string(60)",
  "TipoPessoa": "string(1)",
  "DataNascimento": "string",
  "Genero": "string(1)",
  "DocumentoFiscal": "string(15)",
  "Identidade": "string(20)",
  "RazaoSocial": "string(60)",
  "CEP": "string(9)",
  "Logradouro": "string(185)",
  "LogradouroNumero": "string(60)",
  "Bairro": "string(60)",
  "LogradouroComplemento": "string(60)",
  "Municipio": "string(35)",
  "UnidadeFederativa": "string(2)",
  "Telefone": "string(9)",
  "TelefoneDDD": "string(3)",
  "Celular": "string(9)",
  "CelularDDD": "string(3)",
  "Email": "string(150)",
  "DataCadastro": "string",
  "ReceberEmail": true,
  "Observacoes": "string(70)",
  "PrecoValidade": 0,
  "MinimoVenda": 0,
  "Ativo": true,
  "TipoDestinatario": "string(1)",
  "DescontosPrazo": 0,
  "DescontoVista": 0,
  "MunicipioNome": "string(35)"
}
```


Sendo assim, é confirmado que ouve atualização/Inserção do cliente com sucesso no Master.

4. **Método: POST /getclientes**

Descrição: Gera lista de clientes

Parâmetros: Token e TipoPessoa

Body JSON:

```
{
  "TipoPessoa": {
    "Filtro": {
      "Like": true,
      "Length": 0,
      "Operator": "string"
    },
    "Valores": [
      "string"
    ]
  }
}
```

Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "Nome": "string(60)",
  "TipoPessoa": "string(1)",
  "DataNascimento": "string",
  "Genero": "string(1)",
  "DocumentoFiscal": "string(15)",
  "Identidade": "string(20)",
  "RazaoSocial": "string(60)",
  "CEP": "string(9)",
  "Logradouro": "string(185)",
  "LogradouroNumero": "string(60)",
  "Bairro": "string(60)",
  "LogradouroComplemento": "string(60)",
  "UnidadeFederativa": "string(2)",
  "Telefone": "string(9)",
  "TelefoneDDD": "string(3)",
  "Celular": "string(9)",
  "CelularDDD": "string(3)",
  "Email": "string(150)",
  "DataCadastro": "string",
  "ReceberEmail": true,
  "Observacoes": "string(70)",
  "Ativo": true,
```

```
  "TipoDestinatario": "string(1)",
  "DescontosPrazo": 0,
  "DescontoVista": 0,
  "MunicipioNome": "string(35)",
  "Enderecos": [
    {
      "Codigo": 0,
      "NaoPersistir": true,
      "ClienteCodigo": 0,
      "Municipio": "string(35)",
      "MunicipioCodigo": 0,
      "Logradouro": "string(185)",
      "LogradouroNumero": "string(60)",
      "Bairro": "string(60)",
      "CEP": 0,
      "UnidadeFederativa": 0,
      "Referencia": "string(60)",
      "Pais": "string(50)"
    }
  ],
  "Telefones": [
    {
      "Codigo": 0,
      "NaoPersistir": true,
      "ClienteCodigo": 0,
      "Tipo": "string(1)",
      "DDD": 0,
      "Telefone": 0,
      "Observacao": "string(70)",
      "PadraoNotaFiscal": true
    }
  ],
  "PrecoValidade": 0,
  "MinimoVenda": 0,
  "GrupoCodigo": 0,
  "Municipio": "string(35)"
}
```

Nesse caso irá gerar, portanto uma lista de clientes baseada no TipoPessoa e caso não encontre, retornará uma mensagem de erro.

2.8.3 Fabricante

Esse grupo aborda informações sobre os fabricantes relacionados aos produtos, sendo possível buscar, inserir e atualizar essas informações.

1. **Método: GET /fabricante**

Descrição: Retorna os fabricantes da fila de integração.
Parâmetros: Apenas o Token.
Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "Nome": "string(40)",
    "Ativo": true,
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Contato": "string(50)"
  }
]
```

Nesse caso é retornado um Array de Objetos JSON, cada objeto tem as informações dos fabricantes.

2. **Método: GET /fabricante/{código}**

Definição: Retorna o fabricante relacionado ao código.
Parâmetros: Token e código

Response JSON: **Status 200 – OK**

```
{
    "Codigo": 0,
    "Nome": "string(40)",
    "Ativo": true,
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Contato": "string(50)"
}
```

Nesse caso, portanto é retornado apenas um Objeto JSON, com as informações do fabricante escolhido através de seu código específico.

3. Método: PUT /fabricante

Descrição: Insere novo fabricante, ou atualiza um existente.

Parâmetros: Token e código caso for atualizar.

Body JSON:

```
{
    "Codigo": 0,
    "Nome": "string(40)",
    "Ativo": true,
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Contato": "string(50)"
}
```

Response JSON: **Status 200 – OK**

```
{
    "Codigo": 0,
    "Nome": "string(40)",
    "Ativo": true,
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Contato": "string(50)"
}
```

Nesse caso, ao retornar status 200, na response da requisição é mostrado o fabricante que foi inserido ou atualizado. Caso não retorne status 200, irá devolver um JSON de erro.

2.8.4 Forma de Pagamento

Esse grupo é responsável por retornar as formas de pagamentos possíveis e cadastradas no sistema.

1. **Método: GET /formapagamento**

Descrição: Retorna todas as formas de pagamento existentes no ERP.

Parâmetros: Token.

Response JSON: **Status 200 – OK**

```
{
      "formas_pgto": [
            {
                  "Codigo": 0,
                  "Nome": "string(40)"
            }
      ]
}
```

Nesse caso é retornado um Array de Objetos JSON, com o código e o nome das formas de pagamento possíveis.

2.8.5 Forma de Parcelamento

Esse grupo é responsável por tratar e buscar as formas de parcelamento cadastradas no sistema.

1. **Método: GET /formaparcelamento**

Descrição: Retorna todas as formas de parcelamento cadastradas no ERP.
Parâmetros: Apenas o Token.
Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "Nome": "string(40)",
    "DiasInicial": 0,
    "Intervalo": 0,
    "Quantidade": 0,
    "Compra": true,
    "Padrao": true
  }
]
```

Nesse caso é retornado um Array de objetos, onde cada objeto é uma forma de parcelamento diferente.

2. **Método GET /formaparcelamento/{código}**

Descrição: Retorna forma de parcelamento especifica pelo seu código.
Parâmetros: Token e código.
Response JSON: **Status 200 - OK**

```
  {
    "Codigo": 0,
    "Nome": "string(40)",
    "DiasInicial": 0,
    "Intervalo": 0,
    "Quantidade": 0,
    "Compra": true,
    "Padrao": true
  }
```

Nesse caso, o retorno será um objeto, indicando qual foi o parcelamento requisitado.

2.8.6 Fornecedor

Esse grupo contém dados de fornecedores, sendo possível realizar a busca, inserção e alteração dessas informações.

1. **Método: GET /fornecedor**

Descrição: Retorna fornecedores da fila de integração.
Parâmetros: Token.
Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "Nome": "string(60)",
    "TipoPessoa": "string(1)",
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Observacao": "string(70)",
    "Contato": "string(50)"
  }
]
```

É retornado um array de objetos, onde cada objeto representa o fornecedor e suas informações dentro da fila de integração.

2. **Método: GET /fornecedor/{código}**

Descrição: Retorna um fornecedor específico, dependendo do seu código.
Parâmetros: Token e código.

Response JSON: **Status 200 – OK**

```
{
    "Codigo": 0,
    "Nome": "string(60)",
    "TipoPessoa": "string(1)",
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Observacao": "string(70)",
    "Contato": "string(50)"
}
```

Nesse retorno é devolvido apenas um objeto com as informações do fornecedor especificado pelo código.

3. **Método: PUT /fornecedor**

Descrição: Insere novo fornecedor, ou atualiza um existente.

Parâmetros: Token e código caso for atualização.

Body JSON:

```
{
    "Codigo": 0,
    "Nome": "string(60)",
    "TipoPessoa": "string(1)",
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Observacao": "string(70)",
    "Contato": "string(50)"
}
```

Response JSON: **Status 200 – OK**

```
{
    "Codigo": 0,
    "Nome": "string(60)",
    "TipoPessoa": "string(1)",
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "Bairro": "string(60)",
    "Telefone": "string(15)",
    "Observacao": "string(70)",
    "Contato": "string(50)"
}
```

Nessa requisição, ao retornar o status 200, é devolvido o fornecedor que foi atualizado ou inserido no sistema.

2.8.7 Ping

O grupo do Ping é responsável pela obtenção dos status do Ping da conexão com o servidor.

1. **Método: GET /ping**

Descrição: Retorna data e hora atual, junto com a data e hora do banco de dados.
Parâmetros: Token.
Response JSON: **Status 200 – OK**

```
{
  "status": true,
  "datalocal": "2018-07-13T11:07:57.521Z",
  "databanco": "2018-07-13T11:07:57.521Z",
  "conexaobanco": true
}
```

Tanto a data atual quanto a data do banco são retornadas dentro de um objeto, juntamente com status de conexão com o banco.

2.8.8 Rastreamento

O grupo de rastreamento é o responsável buscar e alterar dados de rastreio dentro do Master, com apenas duas requisições possíveis.

1. **Método: GET /tracking/{códigoVenda}**

Descrição: Retorna status de rastreio.
Parâmetros: Token e código da venda.

Response JSON: **Status 200 – OK**


```
{
  "Tracking": [
    {
      "NaoPersistir": true,
      "CodigoVenda": 0,
      "CodigoStatusTransportadora": 0,
      "Data": "2018-07-13T11:07:57.612Z",
      "Hora": "2018-07-13T11:07:57.612Z",
      "Descricao": "string(200)"
    }
  ]
}
```


Nesse objeto será retornado um array de rastreio, com informações de rastreamento.

2. Método PUT /tracking

Descrição: Altera entrada de rastreio.

Parâmetros: Token.

Body JSON:


```
{
  "NaoPersistir": true,
  "CodigoVenda": 0,
  "CodigoStatusTransportadora": 0,
  "Data": "2018-07-13T11:07:57.378Z",
  "Hora": "2018-07-13T11:07:57.378Z",
  "Descricao": "string(200)"
}
```


Response JSON: **Status 200 – OK**


```
{
  "Tracking": [
    {
      "NaoPersistir": true,
      "CodigoVenda": 0,
      "CodigoStatusTransportadora": 0,
      "Data": "2018-07-13T11:07:57.612Z",
      "Hora": "2018-07-13T11:07:57.612Z",
      "Descricao": "string(200)"
    }
  ]
}
```

2.8.9 Status do Pedido

O grupo status do pedido é responsável por buscar ou alterar o status do pedido de uma venda no sistema.

1. **Método: GET /statuspedido**

Descrição: Retorna o status de todos os pedidos pendentes criados pela API
Parâmetros: Token.
Response JSON: **Status 200 – OK**


```
[
  {
    "Codigo": 0,
    "NaoPersisitir": true,
    "VendaCodigo": 0,
    "PedidoID": 0,
    "StatusID": 0,
    "NotificarCliente": true,
    "NumeroObjeto": "string(20)",
    "Enviado": true,
    "NotaFiscalNumero": 0,
    "NotaFiscalSerie": 0,
    "NotaFiscalSituacao": "string(500)",
    "NotaFiscalEletronica": "string(50)",
    "Data": "2018-07-13T11:07:57.629Z",
    "Hora": "2018-07-13T11:07:57.629Z",
    "PedidoExterno": "string(100)",
    "Hash": "string(36)",
    "LinkNfeConsulta": "string(150)",
    "DataHoraEmissao": "2018-07-13T11:07:57.629Z",
    "TemNotaFiscal": true,
    "NotaFiscalCodigo": 0
  }
]
```


Nesse request é retornado um array de objetos, contendo em cada objeto informações dos pedidos pendentes criados pela API.

2. **Método: GET /statuspedido/{códigoVenda}**

Descrição: Retorna o status do pedido pendente especifico.

Parâmetros: Token, código da venda.

Response JSON: **Status 200 – OK**


```
{
  "Codigo": 0,
  "NaoPersisitir": true,
  "VendaCodigo": 0,
  "PedidoID": 0,
  "StatusID": 0,
  "NotificarCliente": true,
  "NumeroObjeto": "string(20)",
  "Enviado": true,
  "NotaFiscalNumero": 0,
  "NotaFiscalSerie": 0,
  "NotaFiscalSituacao": "string(500)",
  "NotaFiscalEletronica": "string(50)",
  "Data": "2018-07-13T11:07:57.629Z",
  "Hora": "2018-07-13T11:07:57.629Z",
  "PedidoExterno": "string(100)",
  "Hash": "string(36)",
  "LinkNfeConsulta": "string(150)",
  "DataHoraEmissao": "2018-07-13T11:07:57.629Z",
  "TemNotaFiscal": true,
  "NotaFiscalCodigo": 0
}
```


Essa requisição retornará apenas um objeto, com o status do pedido especificado pelo código da venda.

3. **Método: PUT /statuspedido**

Descrição: Altera status do pedido.

Parâmetros: Token e código para atualização.

Body JSON:

```
{
    "Codigo": 0,
    "NaoPersisitir": true,
    "VendaCodigo": 0,
    "PedidoID": 0,
    "StatusID": 0,
    "NotificarCliente": true,
    "NumeroObjeto": "string(20)",
    "Enviado": true,
    "NotaFiscalNumero": 0,
    "NotaFiscalSerie": 0,
    "NotaFiscalSituacao": "string(500)",
    "NotaFiscalEletronica": "string(50)",
    "Data": "2018-07-13T11:07:57.629Z",
    "Hora": "2018-07-13T11:07:57.629Z",
    "PedidoExterno": "string(100)",
    "Hash": "string(36)",
    "LinkNfeConsulta": "string(150)",
    "DataHoraEmissao": "2018-07-13T11:07:57.629Z",
    "TemNotaFiscal": true,
    "NotaFiscalCodigo": 0
}
```

Nessa requisição é enviado no seu corpo, os dados necessários para a alteração do status do pedido. Nessa requisição é necessário enviar o código para conseguir atualizar o pedido corretamente.

Response JSON: **Status 200 – OK**


```
{
    "Codigo": 0,
    "NaoPersisitir": true,
    "VendaCodigo": 0,
    "PedidoID": 0,
    "StatusID": 0,
    "NotificarCliente": true,
    "NumeroObjeto": "string(20)",
    "Enviado": true,
    "NotaFiscalNumero": 0,
    "NotaFiscalSerie": 0,
    "NotaFiscalSituacao": "string(500)",
    "NotaFiscalEletronica": "string(50)",
    "Data": "2018-07-13T11:07:57.629Z",
    "Hora": "2018-07-13T11:07:57.629Z",
    "PedidoExterno": "string(100)",
    "Hash": "string(36)",
    "LinkNfeConsulta": "string(150)",
    "DataHoraEmissao": "2018-07-13T11:07:57.629Z",
    "TemNotaFiscal": true,
    "NotaFiscalCodigo": 0
}
```


O objeto de retorno será o pedido, com o status já alterado e seguindo o mesmo modelo do corpo da requisição.

2.8.10 Status Venda

O grupo de status da venda é o responsável pela requisição que busca no sistema do status atual da venda, possui apenas uma requisição que apenas devolve informação do servidor para o cliente.

1. **Método: GET /statusvenda**

Descrição: Retorna o status da venda que está no ERP.
Parâmetros: Token.

Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "Nome": "string(100)"
  }
]
```

É retornado nessa requisição, um array de objetos, sendo que esses objetos são todos os status possíveis cadastrados no sistema.

2.8.11 Transportadora

Esse grupo é o responsável pela busca das transportadoras no sistema, possui apenas uma requisição com o método GET.

1. **Método: GET /transportadora**

Descrição: Retorna todas as transportadoras cadastradas no ERP.
Parâmetros: Token.
Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "Nome": "string(60)",
    "CEP": "string(9)",
    "Municipio": "string(35)",
    "UnidadeFederativa": "string(2)",
    "Bairro": "string(60)",
    "Logradouro": "string(185)",
    "LogradouroNumero": "string(60)",
    "LogradouroComplemento": "string(60)",
    "TipoPessoa": "string(1)",
    "DocumentoFiscal": "string(20)",
    "Identidade": "string(15)",
    "Email": "string(150)",
    "Site": "string(60)",
    "Telefone": "string(15)"
  }
]
```

É retornado um array de objetos, onde cada objeto representa uma transportadora cadastrada no sistema.

2.8.12 Produto

Esse grupo está associado as informações sobre os produtos cadastrados, estoque, preço e suas respectivas grades sendo possível a busca, inserção e alteração.

1. **Método: GET /produto/{código}**

Descrição: Retorno produto associado ao seu código.
Parâmetros: Token e código.
Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "Nome": "string(200)",
  "EAN": "string(20)",
  "Fornecedores": [
    {
      "Codigo": 0,
      "FornecedorCodigo": 0,
      "ProdutoCodigo": 0,
      "FornecedorProdutoCodigo": "string",
      "FornecedorPadrao": true
    }
  ],
  "FabricanteCodigo": 0,
  "Referencia": "string(25)",
  "Classificacao": "string(1)",
  "Modelo": "string(100)",
  "Conteudo": "string(50)",
  "DescricaoCurta": "string(300)",
  "DescricaoLonga": "string",
  "PrecoCusto": 0,
  "PrecoVenda": 0,
  "Promocao": true,
  "PrecoPromocao": 0,
  "PromocaoInicio": "string",
  "PromocaoFim": "string",
  "PrevalecerPrecoPai": true,
  "Taxa": 0,
  "NumeroMaximoParcelas": 0,
  "EstoqueMaximo": 0,
  "EstoqueAtual": 0,
```

```
    "TipoDisponibilidade": 0,
    "PreVenda": true,
    "PreVendaData": "string",
    "PreVendaLimite": 0,
    "VendaSemEstoque": true,
    "VendaSemEstoqueData": "string",
    "VendaSemEstoqueLimite": 0,
    "ExibirDisponibilidade": true,
    "DisponibilidadeEmEstoque": 0,
    "DisponibilidadeSemEstoque": 0,
    "TipoReposicao": "string(1)",
    "Altura": 0,
    "Largura": 0,
    "Profundidade": 0,
    "Peso": 0,
    "PesoCubico": 0,
    "QuantidadeMaximaPorCliente": 0,
    "EstoqueMinimoFrase": "string(100)",
    "FreteGratis": true,
    "Ativo": true,
    "TituloVariacao": "string(100)",
    "TituloSubVariacao": "string(100)",
    "DataEntrada": "string",
    "DataSaida": "string",
    "MetaTitle": "string(300)",
    "MetaDescription": "string(3000)",
    "MetaKeywords": "string(3000)",
    "PalavrasParaPesquisa": "string(200)",
    "Ordem": 0,
    "CategoriaPrincipalCodigo": 0,
    "CategoriaAdicionalCodigos": [
      0
    ],
    "VisualizarBusca": true,
    "VisualizarLogado": true,
    "VisualizarPreco": true,
    "VisualizarUrlDireto": true,
    "TemOpcaoPresente": true,
    "PresenteValor": 0,
    "Video": "string",
    "Unidade": "string(2)",
    "NCM": "string(10)",
    "Foto": "string"
  }
```


É retornado na requisição um objeto JSON com todas as informações do produto.

2. **Método: GET /produto**

Descrição: Retorna todos produtos da fila de integração.

Parâmetros: Token.

Response JSON: **Status 200 – OK**


```
[
  {
    "Codigo": 0,
    "Nome": "string(200)",
    "EAN": "string(20)",
    "Fornecedores": [
      {
        "Codigo": 0,
        "FornecedorCodigo": 0,
        "ProdutoCodigo": 0,
        "FornecedorProdutoCodigo": "string",
        "FornecedorPadrao": true
      }
    ],
    "FabricanteCodigo": 0,
    "Referencia": "string(25)",
    "Classificacao": "string(1)",
    "Modelo": "string(100)",
    "Conteudo": "string(50)",
    "DescricaoCurta": "string(300)",
    "PrecoCusto": 0,
    "PrecoVenda": 0,
    "Promocao": true,
    "PrecoPromocao": 0,
    "PromocaoInicio": "string",
    "PromocaoFim": "string",
    "PrevalecerPrecoPai": true,
    "Taxa": 0,
    "NumeroMaximoParcelas": 0,
    "EstoqueMaximo": 0,
    "EstoqueAtual": 0,
    "TipoDisponibilidade": 0,
    "PreVenda": true,
    "PreVendaData": "string",
    "PreVendaLimite": 0,
    "VendaSemEstoque": true,
    "VendaSemEstoqueData": "string",
    "VendaSemEstoqueLimite": 0,
    "ExibirDisponibilidade": true,
    "DisponibilidadeEmEstoque": 0,
```

```
    "DisponibilidadeEmEstoque": 0,
    "DisponibilidadeSemEstoque": 0,
    "TipoReposicao": "string(1)",
    "Altura": 0,
    "Largura": 0,
    "Profundidade": 0,
    "Peso": 0,
    "PesoCubico": 0,
    "QuantidadeMaximaPorCliente": 0,
    "EstoqueMinimoFrase": "string(100)",
    "FreteGratis": true,
    "Ativo": true,
    "TituloVariacao": "string(100)",
    "TituloSubVariacao": "string(100)",
    "DataEntrada": "string",
    "DataSaida": "string",
    "MetaTitle": "string(300)",
    "MetaDescription": "string(3000)",
    "MetaKeywords": "string(3000)",
    "PalavrasParaPesquisa": "string(200)",
    "Ordem": 0,
    "CategoriaPrincipalCodigo": 0,
    "CategoriaAdicionalCodigos": [
      0
    ],
    "VisualizarBusca": true,
    "VisualizarLogado": true,
    "VisualizarPreco": true,
    "VisualizarUrlDireto": true,
    "TemOpcaoPresente": true,
    "PresenteValor": 0,
    "Video": "string",
    "Unidade": "string(2)",
    "NCM": "string(10)",
    "Foto": "string"
  }
]
```


Essa requisição devolve um array de objetos do tipo produto, com todos os produtos existentes na fila de integração, ou um erro caso não encontre.

3. **Método: PUT /produto**

Descrição: Insere um novo produto, ou atualiza existente.

Parâmetros: Token e código caso for atualização.

Body JSON:

```
{
  "Codigo": 0,
  "Nome": "string(200)",
  "EAN": "string(20)",
  "Fornecedores": [
    {
      "Codigo": 0,
      "FornecedorCodigo": 0,
      "ProdutoCodigo": 0,
      "FornecedorProdutoCodigo": "string",
      "FornecedorPadrao": true
    }
  ],
  "FabricanteCodigo": 0,
  "Referencia": "string(25)",
  "Classificacao": "string(1)",
  "Modelo": "string(100)",
  "Conteudo": "string(50)",
  "DescricaoCurta": "string(300)",
  "DescricaoLonga": "string",
  "PrecoCusto": 0,
  "PrecoVenda": 0,
  "Promocao": true,
  "PrecoPromocao": 0,
  "PromocaoInicio": "string",
  "PromocaoFim": "string",
  "PrevalecerPrecoPai": true,
  "Taxa": 0,
  "NumeroMaximoParcelas": 0,
  "EstoqueMaximo": 0,
  "EstoqueAtual": 0,
  "TipoDisponibilidade": 0,
  "PreVenda": true,
  "PreVendaData": "string",
  "PreVendaLimite": 0,
  "VendaSemEstoque": true,
  "VendaSemEstoqueData": "string",
  "VendaSemEstoqueLimite": 0,
  "ExibirDisponibilidade": true,
  "DisponibilidadeEmEstoque": 0,
```

```
    "DisponibilidadeSemEstoque": 0,
    "TipoReposicao": "string(1)",
    "Altura": 0,
    "Largura": 0,
    "Profundidade": 0,
    "Peso": 0,
    "PesoCubico": 0,
    "QuantidadeMaximaPorCliente": 0,
    "EstoqueMinimoFrase": "string(100)",
    "FreteGratis": true,
    "Ativo": true,
    "TituloVariacao": "string(100)",
    "TituloSubVariacao": "string(100)",
    "DataEntrada": "string",
    "DataSaida": "string",
    "MetaTitle": "string(300)",
    "MetaDescription": "string(3000)",
    "MetaKeywords": "string(3000)",
    "PalavrasParaPesquisa": "string(200)",
    "Ordem": 0,
    "CategoriaPrincipalCodigo": 0,
    "CategoriaAdicionalCodigos": [
      0
    ],
    "VisualizarBusca": true,
    "VisualizarLogado": true,
    "VisualizarPreco": true,
    "VisualizarUrlDireto": true,
    "TemOpcaoPresente": true,
    "PresenteValor": 0,
    "Video": "string",
    "Unidade": "string(2)",
    "NCM": "string(10)",
    "Foto": "string"
  }
```


Response JSON: **Status 200 – OK**

A requisição devolve apenas o status 200 se for bem-sucedida, ou um JSON de erro com o código e mensagem específica.

Essa requisição irá, portanto, tentar criar um produto caso não seja enviado nenhum código, caso contrário irá atualizar algum produto existente no ERP.

4. **Método: GET /estoque**

Descrição: Retorna fila de estoques.

Parâmetros: Token.

Response JSON: **Status 200 – OK**

```
{
    "status": true,
    "Erros": [],
    "estoque": [
        {
            "Codigo": 0,
            "NaoPersistir": false,
            "Empresa": 0,
            "CodigoPai": 0,
            "Estoque": "string",
            "DisponibilidadeEmEstoque": 0,
            "DisponibilidadeSemEstoque": 0,
            "EstoqueSite": 0,
            "QuantidadeTerceiros": "string",
            "TipoDisponibilidade": "string",
            "EstoqueMinimo": 0,
            "MultiEstoque": [...],
            "EAN": "string(20)"
        }
    ]
}
```

Nessa requisição é retornado um array de objetos do tipo estoque, contendo as informações de estoque para determinado produto (delimitado pelo código). Esse array representa a fila de estoques e qual seu produto associado a ele.

5. **Método: GET /estoque/{código}**

Descrição: Retorna estoque do produto no ERP.

Parâmetros: Token e código do produto.

Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "CodigoPai": 0,
  "EstoqueAtual": "0"
}
```

6. **Método: GET /estoquereservado/{código}**

Descrição: Retorna estoque reservado do produto **somado** ao estoque atual

Parâmetros: Token e código do produto.

Response JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "CodigoPai": 0,
  "EstoqueAtual": "0"
}
```

7. Método: PUT /estoque

Descrição: Atualiza estoque do produto no ERP.

Parâmetros: Token e código do produto

Body JSON:

```
{
  "Codigo": 0,
  "Estoque": 0,
  "Empresa": 0,
}
```

Response JSON: **Status 200 – OK**

```
{
    "status": true,
    "Erros": [],
    "Model": {
        "Codigo": 237,
        "CodigoPai": 0,
        "Estoque": "2.00",
        "Empresa": 1
    }
}
```

8. **Método: GET /preco/{código}**
Descrição: Retorna preço do produto.
Parâmetros: Token e código do produto.
Response JSON: **Status 200 – OK**


```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "CodigoPai": 0,
  "PrecoCusto": 0,
  "PrecoVenda": 0,
  "PrecoFicticio": 0,
  "PrecoSite": 0,
  "PrecoFicticioSite": 0,
  "PromocaoInicio": "2018-07-16T12:06:09.762Z",
  "PromocaoFim": "2018-07-16T12:06:09.762Z",
  "PrecoFilhos": [
    0
  ]
}
```


Essa requisição retorna um objeto com todos os preços existentes de determinado produto cadastrado no ERP.

9. **Método: PUT /preco/{codigo}**
Descrição: Grava preço do produto no ERP caso o produto exista.
Parâmetros: Token e código do produto.
Body JSON: **Status 200 – OK**


```
{
  "Codigo": 0,
  "NaoPersistir": true,
  "CodigoPai": 0,
  "PrecoCusto": 0,
  "PrecoVenda": 0,
  "PrecoFicticio": 0,
  "PrecoSite": 0,
  "PrecoFicticioSite": 0,
  "PromocaoInicio": "2018-07-16T12:06:09.557Z",
  "PromocaoFim": "2018-07-16T12:06:09.557Z",
  "PrecoFilhos": [
    0
  ]
}
```

Response JSON: **Status 200 – OK**

A requisição irá devolver status 200 caso tenha sido bem-sucedida ou retornará um objeto de erro, com código e mensagem de erro.

É possível por meio dessa requisição, atualizar os preços do produto especificado e salvar essas informações no ERP.

10. Método: GET /produtograde

Descrição: Retorna todas as grades do ERP.

Parâmetros: Token.

Response JSON: **Status 200 – OK**

```
[
  {
    "GradeXCodigo": 0,
    "GradeYCodigo": 0,
    "GradeX": "string",
    "GradeY": "string"
  }
]
```

11. Método PUT /produtograde

Descrição: Atualiza a grade ou insere novas grades no ERP

Parâmetros: Token, código, GradeXCodigo, GradeYCodigo para alterar.

Body JSON:

```
  {
    "Codigo": 0,
    "GradeXCodigo": 0,
    "GradeYCodigo": 0,
    "GradeX": "string",
    "GradeY": "string"
  }
```

Response JSON: **Status 200 – OK**

A requisição irá devolver status 200 caso seja bem-sucedida, ou retornará um objeto do tipo erro com o código e a mensagem.

Para inserir uma grade nova, os campos da requisição devem ser mandados vazios, caso contrário será atualização.

12. Método: PUT /produtoimagem

Descrição: Adiciona Imagens do produto.

Parâmetros: Token, código do produto e URL da imagem do produto.

Body JSON:

```
{
    "ProdutoCodigo": 0,
    "Imagem": "string",
    "Descricao": "string",
    "Data": "string",
    "Hora": "string"
}
```

Response JSON: **Status 200 – OK**

```
  {
    "Codigo": 0,
    "ProdutoCodigo": 0,
    "Imagem": "string",
    "Descricao": "string",
    "Data": "string",
    "Hora": "string"
  }
```

Em caso de sucesso a requisição devolve a própria imagem

13. Método: GET /produtoimagem/{código}

Descrição: Retorna Imagens do produto.

Parâmetros: Token e código do produto.

Response JSON: **Status 200 – OK**

```
[
  {
    "Codigo": 0,
    "ProdutoCodigo": 0,
    "Imagem": "string",
    "Descricao": "string",
    "Data": "2018-07-16",
    "Hora": "string"
  }
]
```

A requisição devolve um array do tipo *"produtoimagem",* onde tem as imagens do produto cadastradas no ERP, em formato string.

14. Método: POST /getprodutos
Descrição: Gera fila de produtos.
Parâmetros: Token.
Body JSON:

```
{
  "TipoPessoa": {
    "Filtro": {
      "Like": true,
      "Length": 0,
      "Operator": "string"
    },
    "Valores": [
      "string"
    ]
  }
}
```

Response JSON: **Status 200 – OK**
Array com objetos do tipo produto que estão na fila de integração caso seja bem-sucedida ou objeto do tipo erro, com array dos produtos com erro.

2.8.13 Venda

Esse grupo é responsável por controlar as vendas que ocorrem no ERP, sendo possível buscar, inserir e alterar uma venda, além de algumas outras funcionalidades relacionadas com a venda.

1. **Método GET /venda/{código}**
Descrição: Retorna venda associada ao código.
Parâmetros: Token e código.

```
    "ValorFrete": 0,
    "ValorEncargos": 0,
    "ValorDesconto": 0,
    "ValorTotalBruto": 0,
    "DataVenda": "string",
    "EntregaNome": "string",
    "EntregaEmail": "string",
    "NumeroObjeto": "string",
    "EntregaTelefone": "string",
    "EntregaLogradouro": "string",
    "EntregaLogradouroNumero": "string",
    "EntregaLogradouroComplemento": "string",
    "EntregaBairro": "string",
    "EntregaMunicipio": "string",
    "EntregaUnidadeFederativa": "string",
    "EntregaCEP": "string",
    "CupomDescontoCodigo": "string",
    "CupomDescontoValor": 0,
    "VendaPagamentos": [
      {
        "Valor": 0,
        "DataVencimento": "string",
        "DataCaptura": "string",
        "NumeroAutenticacao": "string",
        "NumeroAutorizacao": "string",
        "FormaPagamentoCodigo": 0,
        "VendaCodigo": 0
      }
    ],
    "Itens": [
      {
        "Codigo": 0,
        "VendaCodigo": 0,
        "ProdutoCodigo": 0,
        "PrecoUnitarioVenda": 0,
        "PrecoUnitarioCusto": 0,
        "EmbaladoParaPresente": true,
        "Quantidade": 0,
        "AtributosEspeciais": "string",
        "ItemNome": "string",
        "ItemDescontoUnitario": 0
      }
```


Response JSON: **Status 200 – OK**

```
    ],
    "Observacoes": "string",
    "ObservacoesLoja": "string",
    "CodigoStatus": 0,
    "DescricaoStatus": "string",
    "PrevisaoEntrega": "string",
    "Origem": "string",
    "OrigemPedido": "string",
    "CodigoPedidoExterno": "string",
    "TipoVenda": "string",
    "Reservada": true
  }
```

A requisição devolve um objeto do tipo venda, com as informações da venda salva no ERP. Caso a requisição não consiga fazer, irá devolver um objeto do tipo erro.

2. Método GET /venda

Descrição: Retorna vendas da fila de integração.

Parâmetros: Token.

Response JSON: **Status 200 – OK**

A requisição devolve um array de objetos do tipo venda (citado acima), com todas as vendas na fila de integração. Caso não encontre, retorna um objeto do tipo erro com código e mensagem.

3. Método: PUT /venda

Descrição: Insere venda.

Parâmetros: Token.

Body JSON: **Status 200 – OK**

```
{
  "Codigo": 0,
  "ClienteCodigo": 0,
  "ClienteTipoPessoa": "string(1)",
  "DocumentoFiscal": "string(20)",
  "Identidade": "string(15)",
  "TransportadoraCodigo": 0,
  "ValorTotal": 0,
  "ValorFrete": 0,
  "ValorEncargos": 0,
  "ValorDesconto": 0,
  "ValorTotalBruto": 0,
  "DataVenda": "string",
  "EntregaNome": "string(60)",
```

```
  "EntregaEmail": "string(150)",
  "NumeroObjeto": "string(20)",
  "EntregaTelefone": "string(20)",
  "EntregaLogradouro": "string(60)",
  "EntregaLogradouroNumero": "string(60)",
  "EntregaLogradouroComplemento": "string(60)",
  "EntregaBairro": "string(60)",
  "EntregaMunicipio": "string(35)",
  "EntregaUnidadeFederativa": "string(2)",
  "EntregaCEP": "string(9)",
  "CupomDescontoCodigo": "string(20)",
  "CupomDescontoValor": 0,
  "VendaPagamentos": [
    {
      "Valor": 0,
      "DataVencimento": "string",
      "DataCaptura": "string",
      "NumeroAutenticacao": "string(50)",
      "NumeroAutorizacao": "string(50)",
      "FormaPagamentoCodigo": 0,
      "VendaCodigo": 0
    }
  ],
  "Itens": [
    {
      "Codigo": 0,
      "VendaCodigo": 0,
      "ProdutoCodigo": 0,
      "PrecoUnitarioVenda": 0,
      "PrecoUnitarioCusto": 0,
      "EmbaladoParaPresente": true,
      "Quantidade": 0,
      "AtributosEspeciais": "string(100)",
      "ItemNome": "string(200)",
      "ItemDescontoUnitario": 0
    }
  ],
  "Observacoes": "string",
  "ObservacoesLoja": "strin",
  "CodigoStatus": 0,
  "DescricaoStatus": "string(200)",
  "PrevisaoEntrega": "string",
  "Origem": "string(1)",
  "OrigemPedido": "string(100)",
  "CodigoPedidoExterno": "string(100)",
```

```
  "TipoVenda": "string(1)",
  "Reservada": true
}
```

A requisição é responsável por fazer a inserção da venda no ERP, não sendo possível alterá-la.

Response JSON: **Status 200 – OK**

Se a requisição for bem sucedida, retornará apenas o status 200, caso contrário, retornará um objeto do tipo erro, com código e mensagem.

4. Método: GET /notafiscal/{vendacodigo}

Descrição: Retorna nota fiscal relacionada a venda.

Parâmetros: Token e código da venda.

Response JSON: **Status 200 – OK**

```
{
  "CodigoCliente": 0,
  "CodigoVenda": 0,
  "NotaFiscalNumero": 0,
  "NumeroObjeto": "string",
  "NumeroPedido": 0,
  "NotaFiscalSerie": 0,
  "NotaFiscalEletronica": "string",
  "NotaFiscalXML": "string",
  "LoteEspecie": "string",
  "DataEmissao": "2018-07-16T12:06:09.844Z",
  "DestinatarioNome": "string",
  "DestinatarioTipoPessoa": "string",
  "DestinatarioDocumentoFiscal": "string",
  "DestinatarioIdentidade": "string",
  "DestinatarioLogradouro": "string",
  "DestinatarioLogradouroNumero": "string",
  "DestinatarioComplemento": "string",
  "DestinatarioBairro": "string",
  "DestinatarioMunicipioCodigo": 0,
  "DestinatarioMunicipio": "string",
  "DestinatarioUF": "string",
  "DestinatarioCEP": "string",
  "DestinatarioTelefone": "string",
  "ValorTotal": 0,
  "Peso": 0, ...
```

```
    "PesoCubico": 0,
    "Volumes": 0,
    "TransportadoraCodigo": 0,
    "TransportadoraNome": "string"
  }
```

A requisição irá retornar um objeto com as informações da venda, juntamente com valor da nota fiscal e seu respectivo XML.

5. Método: POST /venda/{código}/aprovar
Descrição: Aprova a venda.
Parâmetros: Token e código da venda.
Response JSON: **Status 200 – OK**
Se a venda for aprovada, irá retornar apenas o status 200, caso contrário será retornado um objeto do tipo erro, com código e mensagem.

6. Método: POST /venda/{código}/cancelar
Descrição: Cancela a venda.
Parâmetros: Token e código da venda.
Response JSON: **Status 200 – OK**
Assim como o método acima, se o cancelamento for bem-sucedido irá retornar status 200, caso contrário, será retornado um objeto do tipo erro, com código e mensagem.

7. Método: GET /vendafiscal
Descrição: Retorna lista de vendas na fila, com informação de nota e empresa.
Parâmetros: Token
Response JSON: **Status 200 – OK**

```
[
      "Venda": {...}, //Mesmo modelo do get de Venda
      "NotaFiscal": {...}, // Mesmo modelo do get de Nota
      "Empresa": {...} //Modelo de Empresa
]
```

2.8.14 Transferência

Endpoint responsável por administrar as transferências que ocorrem no ERP, sendo possível realizar busca ou inserção através do código de transferência informado.

1. **Método: GET /transferencia/{código de transferencia}**

Descrição: Busca transferência a partir do seu código.
Parâmetros: Token e código de transferência.

```
            "Hora": "1899-12-30T16:26:44.993-03:00",
            "TipoTransferencia": "string",
            "Destino": "string",
            "Observacao": "string",
            "Empresa": integer,
            "Status": "",
            "CodigoCliente": integer,
            "CodigoFornecedor": integer,
            "Itens": [
                {
                    "Codigo": integer,
                    "NaoPersistir": boolean,
                    "CodigoTransferencia": integer,
                    "CodigoProduto": integer,
                    "ValorUnitario": "string",
                    "Quantidade": "string",
                    "Total": "string",
                    "Tipo": "string"
                }
            ]
        }
    ]
}
```

Response JSON: **Status 200 - OK**

A requisição retorna a data e hora, o tipo de transferência realizada, o seu destino, qual a empresa, o código do cliente e do fornecedor e os itens da transferência.

2. **Método: PUT /transferencia/**

Descrição: Insere uma nova transferência a partir dos dados específicos enviados.
Parâmetros: Token

Body JSON: **Status 200 - OK**


```
{
    "CodigoUsuario": 0,
    "CodigoUsuario2": 0,
    "Data": "2018-09-24T00:00:00.000-03:00",
    "Hora": "1899-12-30T16:32:00.000-03:00",
    "TipoTransferencia": "S",
    "Destino": "testao",
    "Observacao": "",
    "Empresa": 1,
    "Status": "",
    "CodigoCliente": 0,
    "CodigoFornecedor": 0,
    "Itens": [
        {
            "CodigoProduto": 1,
            "ValorUnitario": "2.00",
            "Quantidade": "2.00",
            "Total": "1.00",
            "Tipo": "S"
        }
    ]
```


Requisição será enviada e com isso gera gerada a transferência. O objeto de retorno é do mesmo tipo que o primeiro get dessa sessão, vejamos:

Response JSON: **Status 200 - OK**


```
{
    "status": boolean,
    "Erros": [],
    "Model": {
        "Codigo": integer,
        "NaoPersistir": boolean,
        "CodigoUsuario": integer,
        "CodigoUsuario2": integer,
        "Data": "2018-09-24T00:00:00.000-03:00",
        "Hora": "1899-12-30T16:32:00.000-03:00",
        "TipoTransferencia": "S",
        "Destino": "string",
        "Observacao": "string",
        "Empresa": integer,
        "Status": "string",
        "CodigoCliente": integer,
        "CodigoFornecedor": integer,
        "Itens": [
            {
                "Codigo": 0,
                "NaoPersistir": false,
                "CodigoTransferencia": integer,
                "CodigoProduto": integer,
                "ValorUnitario": "string",
                "Quantidade": "string",
                "Total": "string",
                "Tipo": "string"
            }
        ]
    }
}
```


3 CASOS E EXEMPLOS

Afim de facilitar o entendimento geral da documentação, será disponibilizado alguns casos e exemplos de requisições, demonstrando como a API se comporta em alguns cenários de desenvolvimento. Esses exemplos serão apenas de chamadas, mostrando algumas peculiaridades que não foram totalmente esmiuçadas no decorrer do capítulo de requisições e que pode estar gerando dúvida.

- Diferença nas requisições GET com o código (objeto) e sem o código (array de objeto).
  Grupo: Fornecedor
  Token: OlCEsb3U5pBL3LC0JTMt48o1f39XXTMNyXsx3fTF6J67708
  Banco: MasterSite

GET /fornecedor:


```
{
    //para os outros grupos, basta trocar o nome do array
    "fornecedor": [
        {
            "Codigo": 10000052,
            "NaoPersistir": false,
            "Nome": "Amatools Coml. e Importadora Ltda",
            "TipoPessoa": "J",
            "DocumentoFiscal": "07228424000157",
            "Identidade": "535371914110",
            "Email": "",
            "Site": "",
            "CEP": "13420-835",
            "MunicipioCodigo": 3717,
            "UnidadeFederativa": "SP",
            "Logradouro": "Av: Dois Corregos",
            "LogradouroNumero": "2.200",
            "Bairro": "Dois Corregos",
            "Telefone": "1934290050",
            "Observacao": "",
            "Contato": ""
        },
        {
            "Codigo": 10000059,
            "NaoPersistir": false,
            "Nome": "MEGA DAN IMPORT. E DISTRIB. LTDA",
            "TipoPessoa": "J",
            "DocumentoFiscal": "09090023000109",
```

```
            "Email": "",
            "Site": "",
            "CEP": "15057-430",
            "MunicipioCodigo": 3838,
            "UnidadeFederativa": "SP",
            "Logradouro": "AVENIDA TARRAF",
            "LogradouroNumero": "3210-F",
            "Bairro": "JARDIM ALTO ALEGRE",
            "Telefone": "1721389840",
            "Observacao": "",
            "Contato": ""
        },
        {
            "Codigo": 10000060,
            "NaoPersistir": false,
            "Nome": "FERT - VALE IND DE FERTILIZANTES LTDA",
            "TipoPessoa": "J",
            "DocumentoFiscal": "06186392000101",
            "Identidade": "394079053118",
            "Email": "",
            "Site": "",
            "CEP": "11940-000",
            "MunicipioCodigo": 3560,
            "UnidadeFederativa": "SP",
            "Logradouro": "ROD. REGIS BITTENCOURT",
            "LogradouroNumero": "474",
            "Bairro": "PQ INDUSTRIAL",
            "Telefone": "1338642928",
            "Observacao": "",
            "Contato": ""
        },
        ... // n-fornecedores
    ]
}
```

GET /fornecedor/10000060

```
{
    //para os outros grupos, basta trocar o nome do array
    "fornecedor": [
        {
            "Codigo": 10000060,
            "NaoPersistir": false,
            "Nome": "FERT - VALE IND DE FERTILIZANTES LTDA",
            "TipoPessoa": "J",
            "DocumentoFiscal": "06186392000101",
            "Identidade": "394079053118",
            "Email": "",
            "Site": "",
            "CEP": "11940-000",
            "MunicipioCodigo": 3560,
            "UnidadeFederativa": "SP",
            "Logradouro": "ROD. REGIS BITTENCOURT",
            "LogradouroNumero": "474",
            "Bairro": "PQ INDUSTRIAL",
            "Telefone": "1338642928",
            "Observacao": "",
            "Contato": ""
        } //apenas 1 objeto dentro do array
    ]
```

- Requisições POST que enviam filtros de busca de dados.
  Token: OlCEsb3U5pBL3LC0JTMt48o1f39XXTMNyXsx3fTF6J67708
  Banco: MasterSite

  POST /getclientes
  Body:

```
{
  "DocumentoFiscal": {
    "Filtro": {
      //não obrigatório, apenas para filtragem
    },
    "Valores": [
      "16399851823" //valores para serem buscados do grupo

    ]
  }
}
```

Response: **Status 200 – OK**


```
{
    "status": true,
    "Erros": [],
    "clientes": [
        {
            "Codigo": 2,
            "NaoPersistir": false,
            "Nome": "ADRIANO MUFALO",
            "TipoPessoa": "F",
            "DataNascimento": "1980-02-01T00:00:00.000-03:00",
            "Genero": "M",
            "DocumentoFiscal": "16399851823",
            "Identidade": "29.468.181-4",
            "RazaoSocial": "",
            "CEP": "38402288",
            "Logradouro": "Avenida Comendador Alexandrino Garcia
- de 1343/1344 ao fim",
            "LogradouroNumero": "14-55",
            "Bairro": "Minas Gerais",
            "LogradouroComplemento": "Casa 84",
            "UnidadeFederativa": "MG",
            "Telefone": "149968680",
            "TelefoneDDD": "14",
            "Celular": "996868025",
            "CelularDDD": "14",
            "Email": "asm@alternativa.net.br",
            "DataCadastro": "2014-04-01T00:00:00.000-03:00",
            "ReceberEmail": false,
            "Observacoes": "",
            "Ativo": true,
            "TipoDestinatario": "",
            "DescontoPrazo": "0.00",
            "DescontoVista": "0.00",
            "MunicipioNome": "",
            "Enderecos": [...],
            "Telefones": [...],
            "PrecoValidade": 0,
            "MinimoVenda": "0.00",
            "GrupoCodigo": 0,
            "Municipio": "3170206"
        },
```

```
        {
            "Codigo": 6,
            "NaoPersistir": false,
            "Nome": "ADRIANO MUFALO",
            "TipoPessoa": "F",
            "DataNascimento": "1980-02-01T00:00:00.000-03:00",
            "Genero": "M",
            "DocumentoFiscal": "16399851823",
            "Identidade": "294681814",
            "RazaoSocial": "",
            "CEP": "17047170",
            "Logradouro": "Rua Elza Felippini",
            "LogradouroNumero": "449",
            "Bairro": "Jardim Auri Verde",
            "LogradouroComplemento": "",
            "UnidadeFederativa": "SP",
            "Telefone": "996868025",
            "TelefoneDDD": "14",
            "Celular": "",
            "CelularDDD": "",
            "Email": "adriano@mufalo.com.br",
            "DataCadastro": "2015-05-28T00:00:00.000-03:00",
            "ReceberEmail": false,
            "Observacoes": "",
            "Ativo": true,
            "TipoDestinatario": "",
            "DescontoPrazo": "0.00",
            "DescontoVista": "0.00",
            "MunicipioNome": "",
            "Enderecos": [...],
            "Telefones": [...],
            "PrecoValidade": 0,
            "MinimoVenda": "0.00",
            "GrupoCodigo": 0,
            "Municipio": "3506003"
        },
        ... //n-clientes com esse valor
      ]
}
```

- Requisições GET de preço e estoque
  Grupo: Fornecedor
  Token: OlCEsb3U5pBL3LC0JTMt48o1f39XXTMNyXsx3fTF6J67708
  Banco: MasterSite

GET /estoque/3
Response: **Status 200 – OK**

```
{
    "status": true,
    "Erros": [],
    "Model": {
        "Estoque": {
            "Codigo": 3,
            "CodigoPai": 0,
            "EstoqueAtual": "10.00"
        }
    }
}

//código refetente ao código do produto.
```

GET /preco/3
Response: **Status 200 – OK**

```
{
    //array de preço, com todos os preços do produto 3
    "preco": [
        {
            "Codigo": 3,
            "NaoPersistir": false,
            "CodigoPai": 0,
            "PrecoCusto": "167.10",
            "PrecoVenda": "13225.00",
            "PrecoFicticio": "300.00",
            "PrecoSite": "0.00",
            "PrecoFicticioSite": "0.00",
            "PromocaoInicio": "1899-12-30T00:00:00.000-03:00",
            "PromocaoFim": "1899-12-30T00:00:00.000-03:00",
            "PrecoFilhos": []
        }
    ]
}
```

- Exemplificação de erro sendo gerado nas requisições
  Grupo: Venda
  Token: OlCEsb3U5pBL3LC0JTMt48o1f39XXTMNyXsx3fTF6J67708
  Banco: MasterSite

  POST /venda/101/cancelar
  Nesse caso a mensagem de erro ao tentar cancelar a venda de número 101, que já se encontra cancelada no sistema.
  Response:

```
{
    "status": false,
    "Erros": [
        {
            "Codigo": 1,
            "Mensagem": "Esta venda já foi CANCELADA."
        }
    ]
}
```

- Exemplo de PUT para inserção e alteração
  Grupo: Categoria
  Token: OlCEsb3U5pBL3LC0JTMt48o1f39XXTMNyXsx3fTF6J67708
  Banco: MasterSite

  PUT /categoria – Inserção
  Body:

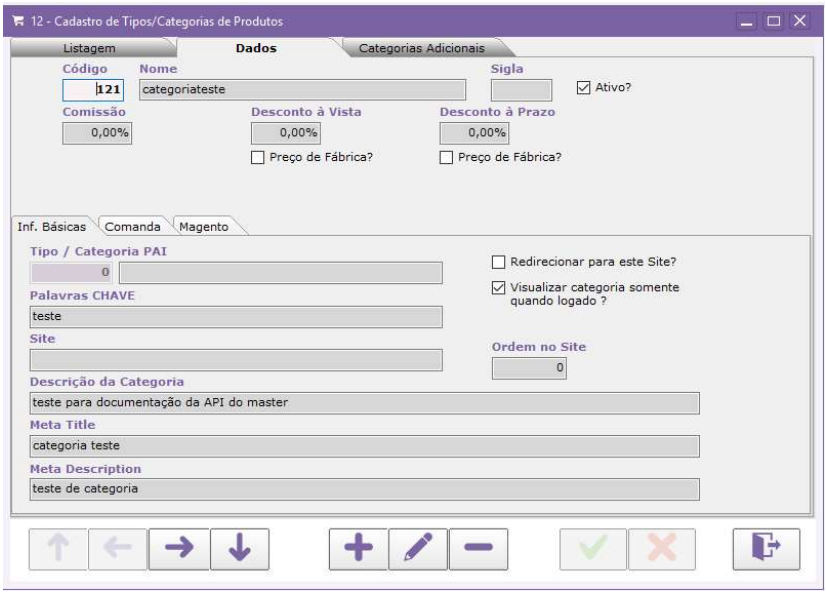
```
{
  "Nome": "categoriateste",
  "Descricao": "teste para documentação da API do master",
  "Ordem": 0,
  "MetaKeywords": "teste",
  "Url": "",
  "Ativo": true,
  "VisualizarLogado": true,
  "MetaTitle": "categoria teste",
  "MetaDescription": "teste de categoria"
}
//por ser uma inserção, não colocar o código
```

Response: **Status 200 – OK**


```
{
    "status": true,
    "Erros": [],
    "Model": {
        "Codigo": 121,
        "NaoPersistir": false,
        "CodigoPai": 0,
        "Nome": "categoriateste",
        "Descricao": "teste para documentação da API do
master",
        "Ordem": 0,
        "MetaKeywords": "teste",
        "Url": "",
        "Ativo": true,
        "VisualizarLogado": true,
        "MetaTitle": "categoria teste",
        "MetaDescription": "teste de categoria",
        "Nivel": 0,
        "Arvore": ""
    }
}
```


A partir desse momento ela já está criada no Master, podendo ser acessada pelo seu devido menu, veja imagem abaixo:

Figura 2 – Categoria criada no ERP.



PUT /categoria – Alteração

Body:

```
{
  "Codigo": 121,
  "Nome": "categoriateste",
  "Descricao": "Alteração da descrição",
  "Ordem": 0,
  "MetaKeywords": "teste",
  "Url": "",
  "Ativo": true,
  "VisualizarLogado": true,
  "MetaTitle": "categoria teste",
  "MetaDescription": "teste de categoria"
}
//colocado o código para alteração de categoria
```

Response: **Status 200 - OK**


```
{
    "status": true,
    "Erros": [],
    "Model": {
        "Codigo": 121,
        "NaoPersistir": false,
        "CodigoPai": 0,
        "Nome": "categoriateste",
        "Descricao": "Alteração da descrição",
        "Ordem": 0,
        "MetaKeywords": "teste",
        "Url": "",
        "Ativo": true,
        "VisualizarLogado": true,
        "MetaTitle": "categoria teste",
        "MetaDescription": "teste de categoria",
        "Nivel": 0,
        "Arvore": ""
    }
}
```


A resposta da requisição é um objeto com outro objeto *"**Model**"* dentro. Esse model é do tipo categoria e tem todas as informações atualizadas da categoria específica criada.