

[Primeiros passos com a API da Nuvemshop](#)

[Controle de versões](#)

[Autenticação](#)

[Recursos da API](#) >

[Guias](#) >

[Utilitários](#) >

[Crie seu aplicativo](#) ↗

[REGISTRO DE ALTERAÇÕES](#)

Home > Primeiros passos com a API da Nuvemshop

Versão: 2025-03

## Primeiros passos com a API da Nuvemshop

Esta é uma API no estilo REST que usa JSON para serialização e OAuth 2 para autenticação, seguindo este modelo de [versionamento](#).

### Por onde eu começo

Quer começar a integrar APIs? Aqui está uma lista de verificação rápida:

- 1. Cadastre-se como parceiro na [Tiendanube](#) ou [na Nuvemshop](#).
- 2. Após acessar o painel de administração do seu parceiro, vá para a seção "Aplicativos" e crie seu aplicativo.
- 3. Saiba [como autenticar](#) seu aplicativo conosco.
- 4. Leia a documentação da API para entender o que você pode fazer com seu aplicativo.

[Por onde eu começo?](#)

[Fazer um pedido](#)

[Autenticação](#)

[Identifique seu aplicativo](#)

[Apenas JSON](#)

[Erros do cliente](#)

[Erros do servidor](#)

[Limitação de taxa](#)

[Paginação](#)

[Verbo HTTP](#)

[Línguas e Internacionalização](#)

[Integração de front-end](#)

[Suspensão do acesso à API devido à falta de pagamento](#)

[Recursos da API](#)

[Ajude-nos a melhorar](#)

### Fazer uma

Todos os URLs começam com [https://api.tiendanube.com/2025-03/{store\\_id}](https://api.tiendanube.com/2025-03/{store_id}) ou [https://api.nuvemshop.com.br/2025-03/{store\\_id}](https://api.nuvemshop.com.br/2025-03/{store_id}). Somente SSL.

O caminho é prefixado com o ID da loja e a versão da API. Se alterarmos a API de forma incompatível com versões anteriores, incrementaremos o marcador de versão e manteremos o suporte estável para os URLs抗igos. Para entender melhor,

Portanto, se você quiser acessar a loja com o ID 123456 por meio da API, a URL será uma das seguintes:

- <https://api.tiendanube.com/2025-03/123456>

ou

- <https://api.nuvemshop.com.br/2025-03/123456>.

Para solicitar todos os produtos da loja, você faria o seguinte no curl:

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'User-Agent: MyApp (name@mail.com)' \
https://api.tiendanube.com/2025-03/123456/products
```

Onde ACCESS\_TOKEN é o token de acesso da loja para seu aplicativo (consulte Autenticação)?

Para criar algo, você precisa incluir o Content-Type cabeçalho e os dados JSON:

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: MyApp (name@mail.com)' \
-d '{ "name": "My new product" }' \
https://api.tiendanube.com/2025-03/123456/products
```

Seguimos o framework OAuth 2 para permitir que os usuários autorizem seu aplicativo a usar o Tiendanube/Nuvemshop em nome deles. Resumidamente, quando um usuário instala o aplicativo, você obtém um token de acesso (uma sequência secreta que indica seus direitos sobre a loja dele), o qual deve ser incluído no cabeçalho de cada requisição (como mostrado no exemplo acima).

Leia o [guia de autenticação](#) para começar.

### Identifique seu

Em cada requisição à API, você deve incluir um User-Agent cabeçalho com o nome do seu aplicativo e um link para ele ou seu endereço de e-mail para que possamos entrar em contato com você. Aqui estão alguns exemplos:

User-Agent: Super app (<http://superapp.com/contact>) ou User-Agent: Awesome app ([awesome@app.com](mailto:awesome@app.com))

Se você não fornecer esse cabeçalho, receberá uma 400 Bad Request resposta.

### Apenas

Todos os dados são enviados e recebidos em formato JSON. Nossa formação não possui elemento raiz e utilizamos snake\_case para descrever as chaves dos atributos. Isso significa que você precisa enviar o parâmetro --name Content-Type: application/json; charset=UTF-8 ao realizar envios POST ou PUT de dados para o Tiendanube/Nuvemshop.

Você receberá um 415 Unsupported Media Type código de resposta se omitir o Content-Type cabeçalho.

### do cliente

Estes são os possíveis tipos de erros do cliente em chamadas de API que recebem corpos de requisição:

- O envio de um JSON inválido resultará em uma 400 Bad Request resposta.

```
HTTP/1.1 400 Bad Request
Content-Length: 34
```

```
{"error": "Problems parsing JSON"}
```

- O envio de campos inválidos resultará em uma 422 Unprocessable Entity resposta.

```
HTTP/1.1 422 Unprocessable Entity
Content-Length: 47
```

```
{
  "src": [
    "can't be blank"
  ]
}
```

### do servidor

Se o Tiendanube/Nuvemshop estiver com problemas, você poderá ver um erro 5xx. Isto significa que o aplicativo está completamente fora de ar, mas você também pode ver erros como "502 Bad Gateway Error: Not Forward", "Error: Not Forward" ou "Error: Not Forward". Em todos esses casos, é sua responsabilidade tentar novamente mais tarde. 503 Service Unavailable 504 Gateway Timeout.

### de taxa

Para controlar o tráfego recebido da API, você pode realizar um número limitado de requisições em um determinado período.

Atualmente, a abordagem utilizada para limitar o uso da API baseia-se na implementação do algoritmo Leaky Bucket, onde o tamanho padrão do bucket é de 40 requisições com uma taxa de vazamento de 2 requisições por segundo. Isso significa que você pode realizar até 2 requisições por segundo, com picos de 40 requisições, sem receber um erro 429 Too Many Requests. Utilizamos os seguintes cabeçalhos para fornecer informações sobre o limite de uso:

- x-rate-limit-limit Total de solicitações que podem ser feitas em um determinado período, neste caso, o tamanho do bucket.
- x-rate-limit-remaining Pedidos restantes para preencher completamente o balde.
- x-rate-limit-reset Milissegundos restantes para esvaziar completamente o balde.

Importante: O limite de taxa funciona entre cada loja e aplicativo.

Se a loja for Next ou Evolution (planos superiores), o limite de taxa é multiplicado por 10.

Exemplos:

1. Queue 20 requests at a time Elas não serão processadas simultaneamente. Serão processadas de acordo com o limite de taxa: 2 solicitações/seg. Todas as 20 solicitações levarão 10 segundos para serem processadas.
2. Sends 4 requests/sec A cada segundo, 4 novas solicitações chegam e 2 são processadas. Portanto, a cada segundo, 2 solicitações ficam em fila. Após 20 segundos, teremos 40 solicitações em fila (capacidade máxima do bucket). No segundo seguinte, duas serão processadas normalmente, porém as outras 2 serão perdidas (pois receberão um erro da nossa API: status 429).
3. Queues 50 requests at a time Como o bucket suporta apenas 40 requisições em fila, 10 serão perdidas (pois receberão um erro da nossa API: status 429). As outras 40 serão processadas de acordo com o limite de taxa: 2 requisições/seg.

Por padrão, as requisições que retornam múltiplos itens não serão paginadas. Você deve especificar páginas adicionais com o `page` parâmetro. Você também pode definir um tamanho de página personalizado de até 200 com o `per_page` parâmetro.

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'User-Agent: MyApp (name@email.com)' \
'https://api.tiendanube.com/2025-03/123456/products?page=2&per_page=100'
```

Note que a numeração das páginas começa em 1 e que omitir o `page` parâmetro retornará a primeira página.

Para verificar a contagem total de resultados, você pode usar o `x-total-count` cabeçalho:

```
x-total-count: 156
```

Para verificar os links seguinte e anterior na paginação, você pode usar o `Link` cabeçalho:

```
Link: <https://api.tiendanube.com/2025-03/123456/products?page=3&per_page=100>; rel="next",
<https://api.tiendanube.com/2025-03/123456/products?page=50&per_page=100>; rel="last"
```

Os valores possíveis são:

- `next` Exibe a URL da próxima página de resultados.
- `last` Exibe a URL da última página de resultados.
- `first` Exibe a URL da primeira página de resultados.
- `prev` Exibe a URL da página de resultados imediatamente anterior.

Você **deve** usar os `Link` URLs em vez de criar os seus próprios.

## HTTP

Sempre que possível, a API 2025-03 se esforça para usar os verbos HTTP apropriados para cada ação.

- `HEAD` Pode ser executado em qualquer recurso para obter apenas as informações do cabeçalho HTTP.
- `GET` Utilizado para recuperar recursos.
- `POST` Utilizado para criar recursos.
- `PUT` Utilizado para substituir recursos ou coleções. Para solicitações PUT sem atributo de corpo, certifique-se de definir o cabeçalho `Content-Length` como zero.
- `DELETE` Utilizado para excluir recursos.

## Línguas e

As lojas podem potencialmente ter vários idiomas. Isso significa que algumas propriedades dos endpoints de Produto e Categoria serão objetos que detalham o valor para cada idioma.

Por exemplo,

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'User-Agent: MyApp (name@email.com)' \
-D '' \
'https://api.tiendanube.com/2025-03/123456/categories/1234'
```

```
HTTP/1.1 200 OK
Date: Thu, 05 Sep 2013 18:18:22 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 376
x-main-language: es
```

```
{
  "id": 1234,
  "name": {
    "es": "Nombre de la Categoría",
    "pt": "Nome da Categoria"
  },
  "description": {
    "es": "",
    "pt": ""
  },
  "handle": {
    "es": "nombre-de-la-categoría",
    "pt": "nome-da-categoria"
  },
  "parent": null,
  "subcategories": [
  ],
  "created_at": "2013-09-05T17:59:19+0000",
  "updated_at": "2013-09-05T17:59:19+0000"
}
```

As propriedades afetadas pela internacionalização serão sempre um objeto, mesmo que haja apenas um idioma ativo. Se você se interessa apenas pelo idioma principal, pode obter o código dele lendo o `X-Main-Language` cabeçalho.

Ao enviar dados via POST ou PUT, você pode fornecer um objeto com um valor para cada idioma:

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: MyApp (name@email.com)' \
-d '{ "name": {"es": "Ejemplo", "pt": "Exemplo"} }' \
'https://api.tiendanube.com/2025-03/123456/products'
```

Ou você pode simplesmente fornecer um valor de string, que será aplicado a todos os idiomas:

```
curl -H 'Authentication: bearer ACCESS_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: MyApp (name@email.com)' \
-d '{ "name": "Example" }' \
'https://api.tiendanube.com/2025-03/123456/products'
```

## de front-end

Ao usar `scripts` para integrar seu aplicativo com a loja virtual, você pode querer se vincular a determinados eventos ou acessar objetos relacionados.

Embora o suporte para isso ainda esteja sendo aprimorado, existe um único evento ao qual você pode se vincular: '\$(product\_variant).LS.registerOnChangeVariant(callback)' onde '\$(product\_variant)' é uma função que recebe um único argumento, '\$(product\_variant)', que contém informações sobre a variante de produto escolhida. Em particular, '\$(product\_variant)' contém uma string detalhando um seletor 'Query' que você pode usar para obter um container para o formulário da variante escolhida; isso é importante ao trabalhar com temas que possuem um recurso de "compra rápida", como LinkedMan ou LinkedWoman, pois deve haver mais de um formulário de produto presente em uma página.variant.variant.element.

## Suspensão do acesso à API devido à falta de...

Por se tratar de um serviço de assinatura mensal, é possível que uma loja não renove o serviço. Nesse caso, a loja ficará offline e a API ficará inacessível.

Em ambos os casos, todas as chamadas à API retornarão uma `402 Payment Required` resposta, os scripts não serão incluídos e os webhooks não serão acionados. Certifique-se de tratar esse código de erro para notificar o usuário de que ele precisa refazer o pagamento, em vez de exibir um erro genérico do servidor.

Assim que o pagamento necessário for efetuado, a API volta a ficar acessível.

Se seu aplicativo precisar saber quando o acesso à API for suspenso ou retomado (por exemplo, porque você pode ter perdido um webhook e deseja fazer uma sincronização completa), você pode se registrar nos `app/suspended` e `app/resumed` usando um Webhook.

Observação: esses webhooks não são acionados quando o aplicativo fica sem "dias gratuitos". Nesses casos, a API também ficará inacessível, mas nenhum webhook será acionado.

## da API

- Caixa abandonado
- Categoria
- Campos personalizados de categoria
- Cupons
- Cliente
- Campo personalizado do cliente
- Descontos
- Ordem de projeto
- Localização
- Metacampos
- Ordem
- Campos personalizados do pedido
- Provedor de Pagamento
- Produto
- Campos personalizados do produto
- Imagem do produto
- Variante do produto
- Campos personalizados de variantes de produto
- Roteiro
- Transportadora
- Loja
- Transação
- Webhook

## Ajude-nos a

Por favor, diga-nos como podemos melhorar a API. Se você tiver uma solicitação de recurso específico ou se encontrou um bug, use o GitHub Issues. Sinta-se à vontade para criar um fork desta documentação e enviar um pull request com melhorias.

Para conversar conosco sobre a API, fique à vontade para escrever.

- <<mailto:parceiros@nuvemshop.com.br>> para BR.
- <<mailto:socios@tiendanube.com>> Para AR, MX e outros.

Última atualização em 22 de Janeiro de 2026

Próximo  
Controle de versões »

### Documentos

Designers Web   
Nimbus 

### Comunidade

Instagram   
LinkedIn   
Twitter   
YouTube   
Facebook 

### Mais

Parceiros   
Termos e Condições   
Política de Privacidade   
Política de Qualidade   
Apolar

Copyright © 2026 Nuvemshop. Desenvolvido com Docusaurus.