

Integração API NAPP

A integração NAPP consulta as vendas registradas no Illimitar em um período. O acesso é feito em duas etapas: primeiro a autenticação, que devolve um token; depois a consulta da lista de vendas, enviando esse token.

Os exemplos desta página usam o ambiente de demonstração

`https://demonstracao.pdv.moda` e o token de biblioteca `1b8e80a3-b465-44c4-8fb1-67758fd28d15`.

No ambiente do cliente, troque os dois pelos valores copiados nas telas abaixo. Toda chamada envia o cabeçalho `X-Requested-With: XMLHttpRequest`.

Fluxo da integração

Fluxo da integração NAPP: autenticar, conferir se o retorno chegou inteiro e só então consultar as vendas.

O que copiar antes de chamar a API

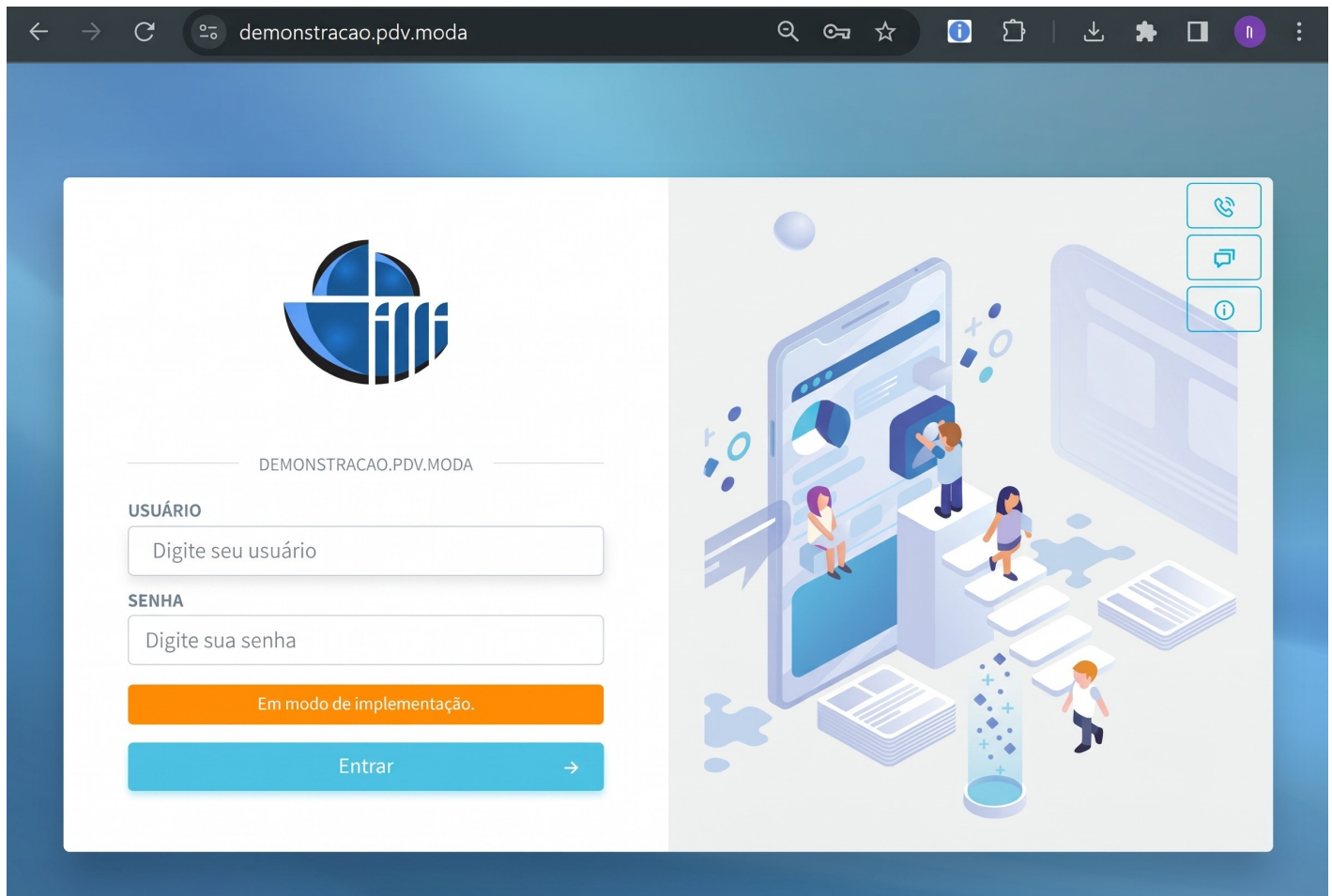
Dado	Onde copiar	Onde entra na chamada
Endereço do ambiente	Barra do navegador, na tela de entrada	Começo de todas as URLs
Usuário	Campo Usuário da tela de entrada	Campo <code>login</code> da autenticação
Senha	Senha cadastrada nesse usuário	Campo <code>senha</code> da autenticação
Token da biblioteca apiNapp	Sistema, Biblioteca, coluna Token	Trecho <code>/bibliotecas/{token}/</code> da consulta de vendas

1. Endereço, usuário e senha

Abra o Illimitar no navegador. O endereço base é o nome do site, antes da primeira barra. Não inclua `/LOGIN` e não deixe barra no final.

No exemplo da coleção, o endereço base é `https://demonstracao.pdv.moda`. No ambiente do cliente, use o endereço que aparecer na barra do navegador.

O usuário e a senha da tela de entrada são os mesmos enviados na autenticação da API. Na grade de usuários, a coluna Usuário é o `login`. A coluna Senha aparece mascarada. A senha da API é a que foi cadastrada nesse usuário.



Há dois caminhos para abrir a grade de usuários, se precisar conferir o login:

- Campo Acesso Rápido, no canto direito da barra superior: digite `Usuário` e selecione o item.
- Ícone Sistema: no grupo Usuário, clique em Usuário.

2. Chave (key) do usuário

A tela abaixo fica no cadastro do usuário de integração. O campo Chave (key), no bloco Acesso API, é a chave desse usuário. O valor que aparece na figura é só um exemplo do ambiente fotografado.

1. Na barra superior, clique no ícone Sistema (engrenagem).
2. Clique em Usuário.
3. Na grade, selecione o usuário da integração. No exemplo, Integração API NAPP.

4. Clique em Configuração.
5. Abra o bloco Acesso API e leia o campo Chave (key).

The screenshot shows the NAPP system configuration interface. The top navigation bar includes icons for various functions. The main menu on the left has 'Início' (1) and 'Usuário' (2). The 'Usuário' menu is expanded, showing 'Incluir', 'Duplicar', 'Configuração' (4), and 'Integração API NAPP' (3). The 'Configuração' option is selected, opening a modal titled 'Configuração do Usuário: Integração API NAPP'. The modal has tabs for 'Fiscal', 'Gestor', 'PDV', and 'Acesso API'. The 'Acesso API' tab is active, showing a 'Chave (key):' field with the value '4c48b339-1a27-4e69-bd78-a9f2746859d8' (5). Below the key field are buttons for 'Gerar novo PIN', 'Redefinir Chave API', 'Google Autenticador', 'Cartão', and 'Salvar'. A legend on the left side of the image provides the following steps:

- 1 - Menu → Sistema
- 2 - Usuário
- 3 - Selecione usuário
- 4 - Configuração do usuário
- 5 - Visualizar a chave API

At the bottom of the page, there is a pagination bar showing 'Página 1 de 1' and an 'Atualizar' button. The bottom right corner indicates '1 à 1 de 1 registro(s)'.

A coleção NAPP não envia a Chave (key). Quem libera a lista de vendas é o campo `bearer` devolvido na autenticação. Não cole essa chave no endereço da consulta e não a use no lugar do bearer.

3. Token da biblioteca apiNapp

O identificador do caminho `/bibliotecas/{token}/apiNapp/api/getVendas` é o Token da biblioteca apiNapp. Ele muda de um ambiente para outro.

1. No campo Acesso Rápido, digite `Biblioteca` e selecione o item. O outro caminho é o ícone Sistema, grupo Controles do Sistema, botão Biblioteca.
2. Na grade, filtre a coluna Nome por `napp`.
3. Use a linha em que a coluna Arquivo / Classe é `apiNapp` e a coluna Saída é `BIBLIOTECA`.
4. Copie o valor da coluna Token.

Junte as peças neste formato:

```
{endereço base}/bibliotecas/{Token}/apiNapp/api/getVendas
```

Exemplo montado com os dados da coleção de demonstração:

```
https://demonstracao.pdv.moda/bibliotecas/1b8e80a3-b465-44c4-8fb1-67758fd28d15/apiNapp/api/getVendas
```

O período vai no final do endereço, por exemplo `?data_inicio=2024-10-01&data_fim=2024-10-05`.

O trecho `1b8e80a3-b465-44c4-8fb1-67758fd28d15` é o Token da biblioteca apiNapp no ambiente da coleção. Em outro ambiente, troque somente esse trecho. O restante do caminho permanece `/bibliotecas/{token}/apiNapp/api/getVendas`.

Convenções

- Datas no formato `aaaa-mm-dd`. Exemplo: `2024-10-01`.
- A autenticação envia os dados como formulário (`application/x-www-form-urlencoded`), não como JSON.
- Todas as chamadas enviam `X-Requested-With: XMLHttpRequest`.
- A consulta de vendas exige o cabeçalho `Authorization: Bearer {token}`.
- Use o corpo só quando ele chegou inteiro. JSON utilizável abre com `{`, fecha com `}` e um leitor de JSON aceita o texto sem erro.
- As respostas de negócio vêm em JSON. Leia sempre o campo `success` antes de usar `data`.
- A autenticação não usa cookie. O token da resposta é o que libera a consulta seguinte.
- Não há paginação. O período inteiro volta na lista `data`.

4. Autenticar

Obtém o token de acesso. Guarde o campo `bearer` da resposta. Ele entra na consulta de vendas.

Método	POST
URL	<code>{endereço}/illi/autenticar</code>
Corpo	<code>application/x-www-form-urlencoded</code>
Cabeçalho	<code>X-Requested-With: XMLHttpRequest</code>

Campos enviados

Campo	Obrigatório	Descrição
login	Sim	Usuário da tela de entrada, o mesmo da coluna Usuário na grade.
senha	Sim	Senha cadastrada nesse usuário.

Exemplo

Troque `usuario` e `senha` pelos dados da tela de entrada. Troque o endereço se o ambiente não for o de demonstração.

```
curl -X POST "https://demonstracao.pdv.moda/illi/autenticar" \  
  -H "Content-Type: application/x-www-form-urlencoded" \  
  -H "X-Requested-With: XMLHttpRequest" \  
  --data "login=usuario&senha=senha"
```

```
POST /illi/autenticar HTTP/1.1  
Host: demonstracao.pdv.moda  
Content-Type: application/x-www-form-urlencoded  
X-Requested-With: XMLHttpRequest  
  
login=usuario&senha=senha
```

Sucesso — HTTP 200

Login aceito. O retorno completo traz `success`, `message`, `data` e `bearer`. Copie o `bearer` da aspa de abertura até a aspa de fechamento.

O exemplo gravado na coleção encurtou dois campos: `bearer` aparece como `eyJT` e `xss` como `bWQ.`. Esses pedaços não autenticam. No ambiente real o texto de cada um vem inteiro. `xss` só aparece quando o usuário tem esse dado. `controle` só aparece quando o usuário tem controle. A falta de `xss` ou de `controle` não torna o retorno incompleto. A falta de `bearer`, com `success` `true`, torna: descarte o corpo e autentique de novo.

JSON de sucesso

```
{
  "success": true,
  "message": "Autenticado com sucesso.",
  "data": false,
  "xss": "bWQ.",
  "bearer": "eyJ"
}
```

Campo	Descrição
success	true quando o login foi aceito.
message	Texto do resultado. No sucesso: Autenticado com sucesso.
data	Nesta resposta de autenticação o exemplo retorna false.
bearer	Token de acesso. Envie este valor na consulta de vendas.
xss	Valor devolvido junto com a autenticação. A consulta de vendas usa o campo bearer.

Guarde o conteúdo de `bearer`. Ele entra no cabeçalho da próxima chamada, no formato `Authorization: Bearer {bearer}`.

Credencial inválida — HTTP 200

Login ou senha incorretos também respondem HTTP 200. A diferença está no JSON. Não há token nesse retorno. Confira usuário e senha antes de consultar vendas.

JSON de credencial inválida

```
{
  "success": false,
  "message": "A credencial informada está incorreta ou é inválida.",
  "data": false,
  "errors": {
    "reason": "A credencial informada está incorreta ou é inválida."
  }
}
```

5. Lista de vendas

Retorna as vendas do período informado. Cada item traz filial, identificador da venda, data, valor e os indicadores de troca e cancelamento.

Método	GET
URL	{endereco}/bibliotecas/{token}/apiNapp/api/getVendas
Autenticação	Authorization: Bearer {bearer}
Cabeçalho	X-Requested-With: XMLHttpRequest

Parâmetros de consulta

Os dois parâmetros vão na URL (query string).

Campo	Tipo	Obrigatório	Descrição
data_inicio	texto aaaa-mm-dd	Sim	Data inicial do período.
data_fim	texto aaaa-mm-dd	Sim	Data final do período.

Exemplo

Substitua `COLE_O_BEARER_AQUI` pelo campo `bearer` da autenticação. Substitua o token da biblioteca e as datas pelo período que você quer consultar.

```
curl "https://demonstracao.pdv.moda/bibliotecas/1b8e80a3-b465-44c4-8fb1-67758fd28d15/apiNapp/api/getVendas?data_inicio=2024-10-01&data_fim=2024-10-05" \  
-H "Authorization: Bearer COLE_O_BEARER_AQUI" \  
-H "X-Requested-With: XMLHttpRequest"
```

```
GET /bibliotecas/1b8e80a3-b465-44c4-8fb1-67758fd28d15/apiNapp/api/getVendas?data_inicio=2024-10-01&data_fim=2024-10-05 HTTP/1.1  
Host: demonstracao.pdv.moda  
Authorization: Bearer COLE_O_BEARER_AQUI  
X-Requested-With: XMLHttpRequest
```

Sucesso — HTTP 200

`success` fica `true` quando existe ao menos uma venda no período. A mensagem é `Lista de vendas`. Leia o array `data`.

Nesta consulta o cabeçalho `Content-Type` pode vir `text/html; charset=UTF-8` mesmo com o corpo em JSON. Vale o corpo: se ele abre com `{` e fecha com `}`, leia como JSON.

JSON com vendas no período

```
{
  "success": true,
  "message": "Lista de vendas",
  "data": [
    {
      "CodigoFilial": "3",
      "idVenda": "12122",
      "DataVenda": "2024-10-01",
      "valor": "347",
      "troca": false,
      "cancelado": false
    },
    {
      "CodigoFilial": "3",
      "idVenda": "12125",
      "DataVenda": "2024-10-01",
      "valor": "597",
      "troca": false,
      "cancelado": false
    },
    {
      "CodigoFilial": "3",
      "idVenda": "12131",
      "DataVenda": "2024-10-01",
      "valor": "447",
      "troca": false,
      "cancelado": false
    },
    {
      "CodigoFilial": "3",
      "idVenda": "12132",
```

```
    "DataVenda": "2024-10-03",
    "valor": "647",
    "troca": false,
    "cancelado": false
  }
]
}
```

Campos de cada venda

Campo	Tipo no exemplo	Descrição
CodigoFilial	texto	Código da filial da venda.
idVenda	texto	Identificador da venda.
DataVenda	texto <code>aaaa-mm-dd</code>	Data da venda.
valor	texto numérico	Valor da venda. No exemplo vem como texto (<code>"347"</code>), não como número JSON.
troca	booleano	<code>true</code> quando a venda é troca.
cancelado	booleano	<code>true</code> quando a venda está cancelada.

Período sem vendas — HTTP 200

Período válido sem movimento também responde HTTP 200, com `success` igual a `false`. Trate esse caso como lista vazia. `data` vem `false`, não como lista vazia (`[]`). O texto de `message` e de `errors.reason` repete o período consultado.

JSON de período sem vendas

```
{
  "success": false,
  "message": "Nenhum registro encontrado no periodo  2024-10-01 - 2024-10-01",
  "data": false,
  "errors": {
    "reason": "Nenhum registro encontrado no periodo  2024-10-01 - 2024-10-01"
  }
}
```

```
}
```

Token ausente ou inválido — HTTP 500

Se o token não for enviado, ou se o usuário não estiver autenticado, a chamada devolve uma página HTML inteira, com HTTP 500. Não é um JSON cortado. O título da página é `Ops!`. No corpo visível há o título `SERVIÇO INDISPONÍVEL NO MOMENTO` e o parágrafo `Usuário não autenticado.`

Não interprete esse HTML como lista de vendas. Autentique de novo e repita a consulta com o `bearer` novo. Se o HTTP 500 continuar com token recém-obtido, o serviço está indisponível.

Corpo que não chegou inteiro

Descarte o corpo e repita a mesma chamada quando acontecer um destes casos:

- O texto parece JSON, mas não fecha com `}`, ou o leitor de JSON recusa o corpo.
- A autenticação traz `success` `true` e o campo `bearer` não veio, veio vazio, ou veio só o pedaço do exemplo (`eyJ`).
- A lista traz `success` `true` e o JSON fechou, mas um item não tem `CodigoFilial`, `idVenda`, `DataVenda`, `valor`, `troca` e `cancelado`. Não grave esse item. Se a lista inteira veio pela metade, repita a consulta do mesmo período.

Uma página que começa com `<!DOCTYPE` ou `<html` é o erro HTTP 500 descrito acima. Trate como falha de autenticação, não como JSON incompleto.

Ordem de chamada

1. `POST /illi/autenticar` com `login`, `senha` e `X-Requested-With: XMLHttpRequest`.
2. Confirme que o JSON fechou, que `success` é `true` e copie o `bearer` inteiro.
3. `GET .../apiNapp/api/getVendas` com `data_inicio`, `data_fim`, `Authorization: Bearer {bearer}` e `X-Requested-With: XMLHttpRequest`.
4. Se o JSON fechou e `success` for `true`, leia a lista em `data`. Cada item precisa dos seis campos.
5. Se `success` for `false` e a mensagem indicar que não há registro no período, encerre a consulta sem vendas.
6. Se a resposta for a página HTML com HTTP 500 e o parágrafo `Usuário não autenticado.`, renove o token e tente outra vez.

7. Se o JSON não fechar, ou se a autenticação vier sem `bearer`, descarte o corpo e repita a mesma chamada.

O que fazer em cada retorno

Situação	HTTP	success	O que fazer
Login aceito	200	true	Usar <code>bearer</code> na consulta de vendas.
Login ou senha inválidos	200	false	Corrigir as credenciais. Não há token.
Vendas encontradas	200	true	Ler o array <code>data</code> .
Nenhuma venda no período	200	false	Tratar como período vazio. <code>data</code> é <code>false</code> .
Token ausente ou inválido	500	não há JSON	Página HTML completa. Título <code>Ops!</code> , texto <code>SERVIÇO INDISPONÍVEL NO MOMENTO</code> e parágrafo <code>Usuário não autenticado.</code> Autenticar de novo.
Corpo cortado ou <code>bearer</code> ausente	o que chegar	JSON inválido, ou <code>true</code> sem <code>bearer</code>	Descartar e repetir a mesma chamada. Não usar token nem venda pela metade.
Item de venda sem os seis campos	200	true	Não gravar esse item. Se a lista veio pela metade, repetir a consulta do período.