

Integrações

Técnicas (Parceiros)

Documentação de API pra parceiros/integradores externos avaliando conectar com o Saara Online

- [API Saara Online — Guia técnico pra integradores \(Base\)](#)

API Saara Online — Guia técnico pra integradores (Base)

Documentação técnica preliminar pra avaliação de integração com a API do Saara Online (marketplace multi-vendedor baseado em Medusa/Mercur). Cobre os 3 grupos de endpoint necessários pra um conector de ERP/plataforma: **produtos**, **estoque** e **pedidos**. Credencial de teste no final da página.

URL base

```
https://slb.saaraonline.com.br
```

Autenticação

Autenticação por sessão via e-mail/senha (padrão do Medusa v2). Cada integrador autentica com as credenciais da loja e recebe um token Bearer:

```
POST /auth/member/emailpass
Content-Type: application/json

{
  "email": "integracao-base@saaraonline.com.br",
  "password": "<fornecida à parte>"
}
```

Resposta:

```
{ "token": "eyJhbGciOiI..." }
```

Todas as chamadas abaixo usam esse token, mais um header identificando a loja (necessário porque o sistema é multi-vendedor — um mesmo login pode ter acesso a mais de uma loja):

```
Authorization: Bearer <token>
x-seller-id: sel_xxxxxxxxxxx
```

O token expira periodicamente (padrão Medusa, ~24h) — o integrador deve re-autenticar quando receber 401, não é necessário pedir um token novo pra cada chamada.

1. Produtos

Listar produtos da loja

```
GET /vendor/products?limit=20&offset=0
```

Resposta (exemplo real, ambiente de teste):

```
{
  "products": [
    {
      "id": "prod_01M328MZ5G6D6N31NQY167CE74",
      "title": "Bolsa Tote Bag Grande Texturizada com Bolsos Frontais",
      "status": "published",
      "thumbnail": "https://slb.saaraonline.com.br/static/....jpeg"
    }
  ],
  "count": 2,
  "offset": 0,
  "limit": 20
}
```

Detalhe de um produto (com variantes e itens de estoque)

```
GET /vendor/products/:id?fields=id,title,status,thumbnail,*variants,*variants.inventory_items
```

```
{
  "product": {
    "id": "prod_01M328MZ5G6D6N31NQY167CE74",
    "title": "Bolsa Tote Bag Grande Texturizada com Bolsos Frontais",
    "status": "published",
    "variants": [
      {
        "id": "variant_01M328MZD80TTK5M5VVJFWHD98",
        "title": "Verde",

```

```
{
  "sku": "skubg109-Verde",
  "weight": 470,
  "inventory_items": [
    { "inventory_item_id": "iitem_01M328MZF5SZYGERAG13JK370" }
  ]
}
```

Criar / atualizar produto

```
POST /vendor/products
POST /vendor/products/:id (atualização)
DELETE /vendor/products/:id
```

Payload de criação segue o schema padrão do Medusa Admin/Vendor API (título, descrição, variantes, opções, preços, categorias, imagens) — a documentação completa de campos é a própria [documentação oficial do Medusa v2](#), já que o Mercur não altera esse contrato pra produtos.

2. Estoque

Consultar níveis de estoque de um item

```
GET /vendor/inventory-items/:inventory_item_id/location-levels
```

```
{
  "inventory_levels": [
    {
      "id": "ilev_01M328MZWFMMVY1QPP2AHESQYR",
      "inventory_item_id": "iitem_01M328MZF5SZYGERAG13JK370",
      "location_id": "sloc_01M2VWBE6C578KN67FZ1ZWFQFP",
      "stocked_quantity": 5,
      "reserved_quantity": 0,
      "available_quantity": 5
    }
  ]
}
```

Atualizar quantidade em estoque

```
POST /vendor/inventory-items/:inventory_item_id/location-levels/:location_id
```

```
{ "stocked_quantity": 10 }
```

Toda loja tem 1 local de estoque (`location_id`) por padrão — devolvido junto na consulta de níveis acima.

3. Pedidos

Listar pedidos da loja

```
GET /vendor/orders?limit=20&offset=0
```

```
{
  "orders": [
    {
      "id": "order_01M2WR5CKY6B0QCN9X781PJBBK",
      "display_id": 636,
      "status": "pending",
      "email": "cliente@exemplo.com",
      "currency_code": "brl",
      "items": [
        {
          "product_id": "prod_01M2WP6J72HQP5D2FZ0NKB3PPS",
          "variant_sku": "TESTE-NFE-001",
          "quantity": 1,
          "unit_price": 5
        }
      ]
    }
  ]
}
```

Detalhe de um pedido

```
GET /vendor/orders/:id
```

Marcar como enviado (fulfillment + rastreio)

```
POST /vendor/orders/:id/fulfillments
POST /vendor/orders/:id/fulfillments/:fulfillment_id/shipments (adiciona código de rastreio)
POST /vendor/orders/:id/fulfillments/:fulfillment_id/cancel
POST /vendor/orders/:id/fulfillments/:fulfillment_id/mark-as-delivered
```

Credencial de teste (temporária)

Login isolado numa loja de testes (sem dados reais de clientes/lojistas), pra avaliação. Pode ser revogado a qualquer momento.

- **Ambiente:** `https://slb.saaraonline.com.br`
- **E-mail:** `integracao-base@saaraonline.com.br`
- **Senha:** fornecida separadamente por canal seguro
- **x-seller-id:** `sel_01M2VWBDT5DMV5PB69333X5NXB`

Essa loja já tem 2 produtos de teste cadastrados (um simples, um com variação de cor + estoque) pra exercitar os exemplos acima.

Observações

- Toda rota `/vendor/*` exige os dois headers (Bearer token + `x-seller-id`) — sem `x-seller-id` a API responde 401 mesmo com token válido.
- Essa é uma documentação preliminar, montada pra destravar a avaliação técnica — se a integração avançar, será formalizada com escopos de API key dedicados por loja, rate limit e webhooks de pedido/estoque.