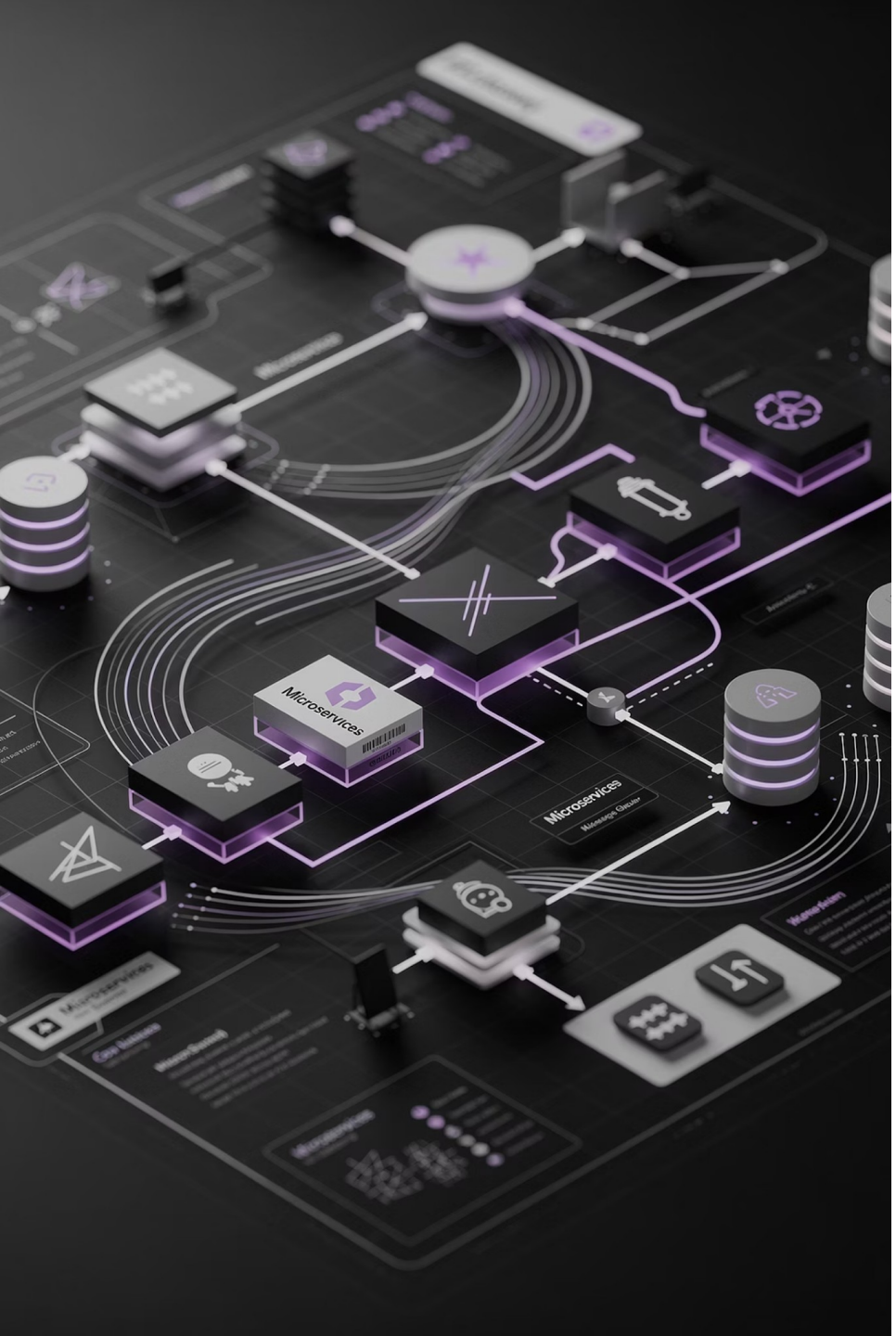



# Boas Práticas de Integração

## API CWS Platform

Recomendações para uma integração estável, performática e pronta para homologação

# Por que seguir estas práticas

Uso correto dos endpoints, controle de chamadas e tratamento de respostas garantem a escala da integração.

01

## Autenticação

Reutilizar o token até expirar e renovar apenas quando necessário.

02

## Logs e erros

Registrar o corpo completo das respostas e tratar cada código HTTP.

03

## Estoque e preço

Carga inicial planejada, envio em lote e atualização só do que mudou.

04

## Pedidos

Webhook como canal principal, listagem como fallback e fluxo de status respeitado.

**Antes de ir para produção**

- Desenvolva e teste todos os fluxos no ambiente sandbox (ws.sandbox.autorei.net).
- Documente os fluxos no modelo de documentação para homologação.
- Agende a reunião de homologação com a CWS para validar os processos antes da virada.

**Nota:** Contratos de cada endpoint, regras de negócio e erros estão na documentação oficial. Dúvidas: suporte-api@cws.digital. api-docs.cws.digital

# Quando solicitar o token

Uma autenticação por processo, reutilizada em todas as chamadas até a expiração.

**POST • 01 - Autenticação • Obter Token de Acesso**

**Regras**

- O access_token vale 12 horas (expires_in = 43199 segundos).
- Reutilize o mesmo token em todas as chamadas do processo.
- Autentique novamente só ao receber 401, com uma única tentativa por execução, para evitar loops.
- Envie sempre Authorization: Bearer <access_token>. Sem o prefixo Bearer, a API responde 401.

**Exemplo: novo produto no ERP**

- Use o token já obtido para enviar o estoque do produto.
- Não autentique de novo entre uma chamada e outra do mesmo processo.
- Se a resposta for 401, gere um novo token uma vez e repita a chamada.
- Se o 401 persistir, interrompa e registre o erro: provável problema de credencial.

**Nota:** Autenticações em excesso prejudicam o desempenho da integração e podem ser identificadas na homologação.

api-docs.cws.digital

# Registre tudo e trate cada resposta

Salve requisição e corpo completo da resposta: isso acelera a análise de chamados.

| Código | Quando acontece | Ação recomendada |
|---|---|---|
| `200 com ERROR` | Envio em lote: item recusado vem com status ERROR e motivo em error | Percorra a lista sempre, mesmo com HTTP 200, e trate item a item |
| `400` | JSON inválido ou falha no lote inteiro (ex.: formas de identificação misturadas) | Corrija o payload. O lote usa code/message/errorDescription em vez de error |
| `401` | Token ausente, sem Bearer, malformado ou expirado | Renove o token uma vez e repita a chamada |
| `403` | Rota por matriz com filial não vinculada (Invalid partnership) | Verifique o vínculo matriz–filial e o partnerId |
| `404` | Recurso inexistente ou fora do escopo da loja | Confira o id usado na rota |
| `422` | Regra de negócio: transição de status, campo obrigatório, limite de itens | Leia o motivo em error ou message e ajuste o envio |

**Nota:** Os formatos de erro variam por endpoint (error ou code/message). Trate os dois e guarde o corpo original no log. api-docs.cws.digital

# Primeira carga em produção

O envio em lote cria ou atualiza o estoque: não é preciso consultar o estoque antes de enviar.

**PUT • 02 - Estoque e Preço • Atualizar ou Criar Estoque em Lote**

**01**

### Defina a identificação

productId, brand + mfrPartCode + composition ou skuAttribute (este exige o serviço de catálogo de produtos).

**02**

### Envie em lotes

Até 50 itens por requisição, todos com a mesma forma de identificação.

**03**

### Guarde o DE/PARA

Salve o productId retornado junto ao código do ERP para os próximos envios.

**04**

### Trate as recusas

Itens com status ERROR voltam com o motivo; corrija e reenvie só esses.

**Loja que já vende na plataforma**

Faça a primeira carga fora do horário comercial e valide antes quais produtos do ERP já têm estoque cadastrado, para evitar divergência de quantidade e preço durante a virada.

**Nota:** A antiga sequência busca de produto → adição → atualização pelo id do estoque foi substituída pelo envio em lote, que cria ou atualiza em uma única chamada.

api-docs.cws.digital

# Regras do envio em lote

Comportamentos que mais causam divergência ou recusa no estoque.

**Envio sempre completo**

Campos opcionais não reenviados são apagados (leadTimeDate, leadTimeByRequest, profitMargin, totalCost, targetMargin). Reenvie o item inteiro.

**Uma forma por lote**

Misturar formas de identificação recusa o lote inteiro com 400 (mixed request types in list).

**Sem itens repetidos**

Mesmo depósito e identificador são deduplicados; se os valores divergirem, o item não é processado.

**Data com fuso**

leadTimeDate exige data e hora completas com offset -03:00. Só a data recusa a requisição inteira.

**Lead time x estoque virtual**

leadTimeByRequest true não aceita leadTime > 0; estoque virtual > 0 exige leadTime quando leadTimeByRequest é false.

**Código único no depósito**

partnerPartCodeUnique: true zera os estoques do depósito que já usam o mesmo partnerPartCode. Use só ao reaproveitar códigos.

**Nota:** O depósito (warehouseId) precisa ser da loja do token, ou da filial informada na rota por matriz; caso contrário o lote inteiro é recusado. api-docs.cws.digital

# Atualização na operação diária

Volume de chamadas alto no horário comercial pede envios pequenos e pontuais.

### Durante o horário comercial

- Envie apenas os produtos com alteração real de quantidade ou preço.
- Agrupe as alterações próximas em um único lote.

### Atualização completa do ERP

- Programe para fora do horário comercial (início da manhã ou noite).
- Divida o estoque em lotes de até 50 itens.

### Lote identificado por productId

```
curl --location --request PUT
'https://ws.sandbox.autorei.net/v1/stock/batch' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "stockRequest": [
    {
      "productId": 500101,
      "warehouseId": 1001,
      "price": 58.62,
      "quantity": 1123,
      "virtualQuantity": 0,
      "leadTime": 2,
      "partnerPartCode": "SKU-A-001"
    }
  ]
}'
```

**Nota:** productId tem a maior precedência na identificação e dispensa a resolução por fabricante ou atributo, tornando o envio mais rápido. api-docs.cws.digital

# Novos produtos no ERP

O estoque só é criado para produtos que existem no catálogo disponível para a loja.

**Fluxo recomendado**

- Envie o novo produto pelo mesmo envio em lote, com a forma de identificação já adotada.
- Com status CREATED, salve o productId no DE/PARA.
- Com status ERROR de produto não encontrado, confirme o cadastro do produto com a CWS.

**Produtos criados pela loja**

- Crie um produto próprio apenas quando ele não existir no catálogo da CWS.
- O produto próprio fica disponível somente para o estoque da sua loja, não para outros parceiros.
- Em marketplaces, verifique as regras para exibição de produtos criados por terceiros, para que o item fique à venda.

**Nota:** Identificação por skuAttributeName/skuAttributeValue exige o serviço de catálogo de produtos, solicitado à CWS. api-docs.cws.digital

PEDIDOS

# Recebimento e atualização de pedidos

Evento em tempo real como canal principal e consulta periódica como garantia.

01

### Webhook

Receba criação e atualização de pedidos. Com payload simplificado, use o id para buscar os detalhes.

02

### Fallback

Liste pedidos por updatedFrom/updatedTo para capturar o que o evento não entregou. Até 25 por página.

03

### Status

Siga o fluxo oficial de status. Para aprovar pagamento, envie APPROVED.

04

### ERP

Em READY, envie supplierOrderId com o número do pedido no seu ERP.

**Ids do pedido**

- orderId é o pedido do cliente (pai); id é o pedido do parceiro, um por fornecedor.
- Todas as rotas de detalhe e de status usam o id do pedido do parceiro.

**Nota:** Detalhes de transições, campos obrigatórios e motivos de cancelamento estão no documento Fluxo de Atualização de Pedidos. api-docs.cws.digital

# Contratos de preço e gestão por matriz

Recursos opcionais, usados conforme o modelo comercial da operação.

**Contratos (opcional)**

- Use apenas quando a operação tiver tabela de preços específica por cliente.
- Liste os contratos antes de criar um novo, evitando duplicidade.
- Adicione até 100 produtos por requisição.
- Produto repetido no mesmo envio não é processado: envie cada um uma vez.

**Gestão por matriz**

- Estoque e contratos têm rotas por matriz, com o partnerId da filial na rota.
- A filial precisa estar vinculada à matriz do token; sem vínculo, 403.
- Pedidos seguem a opção marketplace: pedidos feitos no portal, sem depender de vínculo.

**Nota:** Os exemplos completos de cada rota estão nos materiais Integração API CWS (Loja) e Integração API CWS (Matriz). api-docs.cws.digital

# Checklist para a reunião de homologação

Itens validados com a CWS antes da liberação em produção.

✓ **Fluxos testados no sandbox**

Autenticação, estoque, pedidos e, se usados, contratos e webhook.

✓ **Documentação preenchida**

Modelo de documentação para homologação com fluxos, endpoints e regras.

✓ **Token reutilizado**

Uma autenticação por processo e renovação única em 401.

✓ **Logs completos**

Requisição e corpo da resposta salvos para cada erro.

✓ **Plano de carga inicial**

Estratégia definida para a primeira carga de estoque em produção.

✓ **Rotinas de fallback**

Consulta periódica de pedidos para cobrir falhas do webhook.

**Contato:** Para agendar a homologação ou tirar dúvidas, fale com suporte-api@cws.digital.