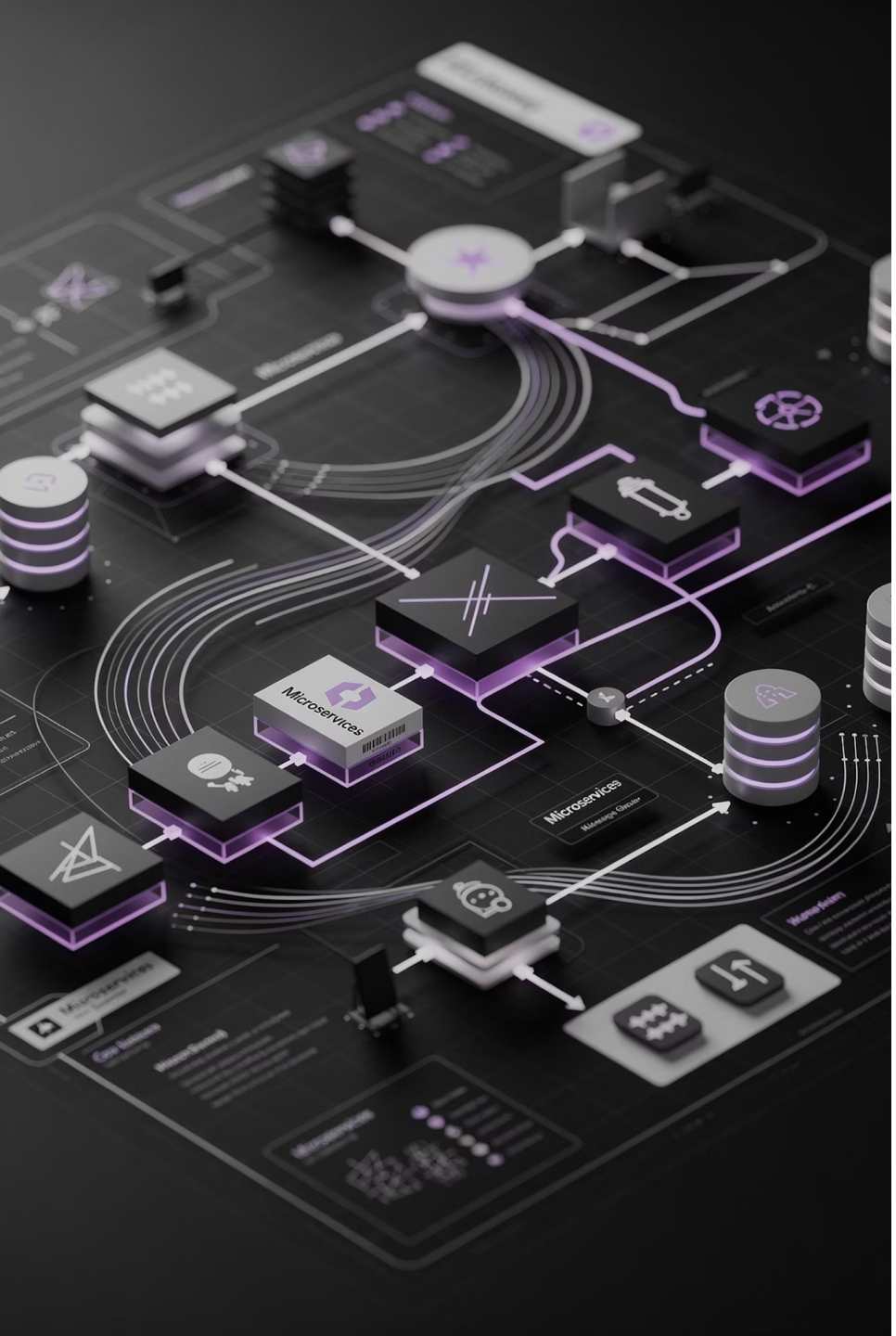


Integração API CWS Platform

Foco em Estoque, Contratos e Pedidos

Fluxo pensado para melhor otimização dos processos do portal

Fluxo recomendado para integração

Sequência sugerida para reduzir retrabalho operacional e facilitar a sustentação do processo.

01

Autenticação

Geração do token de segurança obrigatório para acesso às APIs, usando as credenciais oficiais do portal.

02

Estoque

Sincronização de saldo e disponibilidade com suporte a mapeamento entre SKU do ERP e SKU do portal.

03

Contratos

Gestão de preços por cliente, com consulta, criação e atualização em lote das regras comerciais.

04

Pedidos

Processamento das vendas com webhook e rotinas de fallback para validação de status.

Diretriz operacional

A ordem recomendada é autenticar, sincronizar estoque, validar contratos e então operar pedidos. Esse encadeamento reduz erros de precificação e evita reprocessamentos nas etapas posteriores.

Escopo atual: foco em estoque, contratos e pedidos, priorizando simplicidade de integração e clareza de governança.

Geração do token de acesso

Ponto de entrada obrigatório para todas as chamadas subsequentes da integração.

POST • 01 - Autenticação • Obter Token de Acesso

Requisição (cURL)

```
curl --location 'https://ws.sandbox.autorei.net/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Authorization: Basic YXBpLXdzOnBhc3N3b3JkLW1hc3Rlcg==' \
--data-urlencode 'scope=trust' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=teste.teste@teste.com.br' \
--data-urlencode 'password=@Teste01'
```

Resposta (JSON)

```
{
  "access_token": "56bb7fef-8a7c-4a97-be7b-05f44e423f08",
  "token_type": "bearer",
  "refresh_token": "b550ae00-8f43-4c86-9765-443039880be5",
  "expires_in": 43199,
  "scope": "trust"
}
```

Nota: As credenciais de acesso são fornecidas pela CWS no início da integração. O token deve ser reutilizado até a expiração para evitar chamadas desnecessárias de autenticação.

api-docs.cws.digital

Atualização em lote por atributo de SKU

Modelo indicado quando o produto já nasce com o atributo de integração definido no cadastro.

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

Requisição (cURL)

```
curl --location 'https://ws.sandbox.autorei.net/v1/stock/batch/' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "stockRequest": [
    {
      "skuAttributeName": "codigo-produto-erp",
      "skuAttributeValue": "5095111",
      "warehouseId": 6775,
      "quantity": 10,
      "price": 2297.56,
      "virtualQuantity": 0,
      "leadTime": 7,
      "leadTimeByRequest": false,
      "byRequest": false,
      "partnerPartCode": "802702",
      "priceRuleTags": [{"name": "teste"}]
    }
  ]
}'
```

Resposta (JSON)

```
[
  {
    "id": 63105512,
    "productId": 10294632,
    "warehouseId": 6775,
    "partCode": "802702",
    "price": 2297.56,
    "quantity": 10,
    "leadTime": 7,
    "leadTimeByRequest": false,
    "skuAttributeName": "codigo-produto-erp",
    "skuAttributeValue": "5095111",
    "status": "UPDATED",
    "priceRuleTags": [{"name": "teste"}]
  }
]
```

Nota: A identificação por skuAttributeName/skuAttributeValue exige a contratação do serviço de catálogo de produtos, que deve ser solicitado à CWS. O skuAttributeValue acompanha a criação do produto.
O partnerPartCode reaparece nos fluxos de contrato, esse valor deve ser padronizado e único

api-docs.cws.digital

Atualização por composição e fabricante

Alternativa útil quando a identificação do item parte dos metadados comerciais do produto.

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

Requisição (cURL)

```
curl --location --request PUT
'https://ws.sandbox.autorei.net/v1/stock/batch/' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "stockRequest": [
    {
      "brand": "teste",
      "composition": "UNITARY",
      "mfrPartCode": "PD001",
      "warehouseId": 6775,
      "quantity": 10,
      "price": 2297.56,
      "virtualQuantity": 0,
      "leadTime": 7,
      "leadTimeByRequest": false,
      "byRequest": false,
      "partnerPartCode": "802702",
      "priceRuleTags": [{"name": "teste"}]
    }
  ]
}'
```

Resposta (JSON)


```
[
  {
    "id": 56595234,
    "productId": 9503440,
    "warehouseId": 6775,
    "brand": "Mirador",
    "mfrPartCode": "PD001",
    "composition": "UNITARY",
    "partCode": "802702",
    "price": 2297.56,
    "quantity": 10,
    "leadTime": 7,
    "leadTimeByRequest": false,
    "priceRuleTags": [{"name": "teste"}],
    "status": "UPDATED"
  }
]
```


Nota: O partnerPartCode reaparece nos fluxos de contrato, esse valor deve ser padronizado e único

api-docs.cws.digital

Consulta inicial de contratos

Primeiro passo para localizar regras existentes antes de criar novos cadastros.

GET • 06 - Contratos • Listar Contratos

Requisição (cURL)

```
curl --location 'https://ws.sandbox.autorei.net/pricing/contract/list?
limit=1&offset=0' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json'
```

Resposta (JSON)

```
[
  {
    "id": 197395,
    "name": "contrato teste1",
    "active": true,
    "partner": {
      "id": 4136,
      "name": "",
      "document": ""
    },
    "allCustomers": false,
    "priority": 2,
    "dateStart": "2026-01-01T...",
    "contractType": "COUNTDOWN"
  }
]
```

Nota: Contrato é opcional: use apenas quando a operação tiver tabela de preços específica por cliente (avalie a necessidade). Se usar, prefira o código do cliente no ERP como nome do contrato.

Criação de contrato

Usar apenas quando o contrato do cliente não for localizado na etapa de consulta.

POST • 06 - Contratos • Criar Contrato

Requisição (cURL)

```
curl --location 'https://ws.sandbox.autorei.net/priceContract' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "name": "89002059",
  "startDate": "2026-01-01",
  "endDate": "2026-12-30",
  "allCustomers": false,
  "priority": 2,
  "contractType": "BASIC",
  "allPartners": false,
  "priceBeforeAfter": true,
  "customerGroupList": [
    { "externalId": "89002059" }
  ]
}'
```

Resposta (JSON)


```
{
  "id": 197419,
  "name": "89002059",
  "active": true,
  "partner": {"id": 4136, "name": "", "document": ""},
  "allCustomers": false,
  "allPartners": false,
  "priority": 2,
  "dateStart": "2026-01-01T00:00:00Z",
  "dateEnd": "2026-12-30T00:00:00Z",
  "dateCreated": "2026-04-20T15:26:38.675Z",
  "priceBeforeAfter": true,
  "contractType": "BASIC",
  "priceContractPartners": [],
  "priceContractCustomerGroups": [19915]
}
```


Nota: Recomendação: tratar esta rota como exceção operacional, acionada apenas quando a consulta inicial não retornar um contrato válido para o cliente.

Adição de produtos ao contrato

Etapa para publicar preços por item após a localização ou criação do contrato.

PUT • 06 - Contratos • Adicionar Produtos ao Contrato

Requisição (cURL)

```
curl --location --request PUT
'https://ws.sandbox.autorei.net/pricing/contract/193107/items' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "items": [
    { "partnerPartCode": "FO0000325397", "value": 10.09 },
    { "partnerPartCode": "32231A", "value": 10.09 }
  ]
}'
```

Resposta (JSON)

```
{
  "id": 193107,
  "items": [
    {
      "skuId": 1095124,
      "price": 10.09,
      "targetMargin": null,
      "totalCost": null,
      "status": "updated"
    }
  ],
  "notFound": {
    "items": [
      {
        "partnerPartCode": "32231A",
        "value": 10.09,
        "price": null,
        "targetMargin": null,
        "totalCost": null
      }
    ]
  }
}
```

Nota: partnerPartCode é o mesmo identificador enviado no vínculo do produto ao estoque. O serviço aceita até 100 itens por requisição.

api-docs.cws.digital

Webhook de pedidos

Mecanismo principal para receber eventos de criação e atualização de pedidos em tempo real.

13 - Callbacks e Webhooks • Webhooks de Pedidos

Configuração e autenticação

A CWS disponibiliza um webhook ativado por eventos. Sempre que um pedido é criado ou atualizado, uma notificação é enviada à API do cliente com dados básicos ou com o payload completo para gestão.

- **URL do domínio**
- **Endpoint de envio**
- **Endpoint de auth**
- **Headers de autenticação**

Exemplo de resposta Auth

```
{
  "access_token": "TOKEN",
  "expires_in": 999999
}
```

Resumo operacional

- 2 endpoints necessários
- evento em criação e atualização
- possibilidade de payload básico ou completo
- melhor canal para atualização em tempo real

documentação do webhook

Listagem de pedidos como fallback

Rotina de conferência para identificar pedidos que eventualmente não tenham sido integrados via evento.

GET • 03 - Pedidos • Listar Pedidos da Loja

Requisição (cURL)

```
curl --location 'https://ws.sandbox.autorei.net/partnerOrders/list/?
offset=0&limit=20&status=PAYMENT_APPROVED&createdFrom=2021-07-
16T00%3A00%3A00Z&createdTo=2021-07-15T00%3A00%3A00Z' \
--header 'Authorization: Bearer <TOKEN>'
```

Resposta (JSON)


```
[
  {
    "id": 1773537,
    "orderId": 1719962,
    "total": "R$90,00",
    "status": "PAYMENT_APPROVED",
    "dateCreated": "2025-08-15T10:25:05Z",
    "lastUpdated": "2025-08-15T10:26:49Z",
    "gateway": "STORE_PAYMENT",
    "partner": {
      "partnerName": "Loja teste",
      "partnerDocument": "111.222.333-75"
    },
    "customer": {
      "customerName": "cliente teste CWS",
      "customerDocument": "111.222.333-80"
    }
  }
]
```


Nota: Como fallback do webhook, este endpoint traz dados básicos suficientes para validar pedidos que não tenham sido capturados automaticamente pelo serviço de eventos.

api-docs.cws.digital

Atualização de status do pedido

A transição deve respeitar o ciclo de vida do pedido, sem salto entre etapas.

PUT • 03 - Pedidos • Atualizar Status do Pedido - Loja

Requisição (cURL)

```
curl --location --request PUT
'https://ws.sandbox.autorei.net/partnerOrders/149580' \
--header 'Authorization: Bearer <TOKEN>' \
--header 'Content-Type: application/json' \
--data '{
  "status": "READY",
  "supplierOrderId": "1212",
  "supplierInvoiceNumber": "5555",
  "supplierInvoiceSerie": "XPTO",
  "supplierInvoiceKey": "12345879789798798798798798",
  "supplierInvoiceDate": "2017-09-15"
}'
```

Resposta (JSON)

```
{
  "id": 1448889,
  "status": "PAYMENT_APPROVED",
  "total": 99.00,
  "customer": {
    "name": "Cliente Teste",
    "email": "cliente.teste@teste.com.br"
  },
  "order": {
    "originStoreName": "Loja teste"
  },
  "lastUpdated": "2025-01-13T17:50:07Z"
}
[Payload truncado para visualização]
```

Nota: A integração deve respeitar a ordem oficial dos status. Esse cuidado evita rejeições e inconsistências de acompanhamento no portal.

api-docs.cws.digital