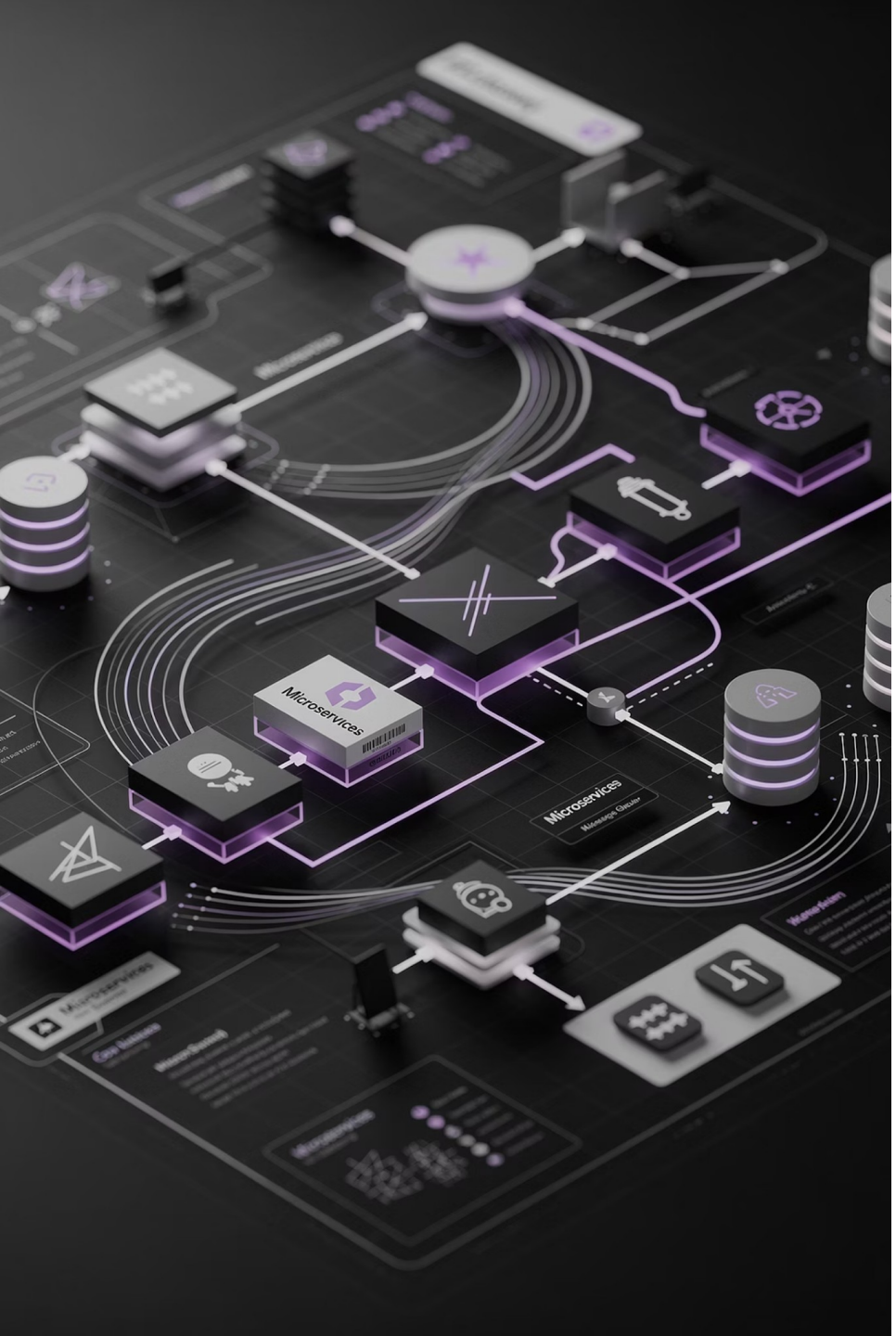

CWS Platform API Integration

Focus on Stock, Contracts and Orders

Flow designed to better optimize the portal's processes

Recommended integration flow

Suggested sequence to reduce operational rework and make the process easier to maintain.

01

Authentication

Generation of the security token required to access the APIs, using the portal's official credentials.

02

Stock

Synchronization of balance and availability, with support for mapping between the ERP SKU and the portal SKU.

03

Contracts

Price management per customer, with query, creation and batch update of the commercial rules.

04

Orders

Sales processing with webhook and fallback routines for status validation.

Operational guideline

The recommended order is to authenticate, synchronize stock, validate contracts and then operate orders. This sequence reduces pricing errors and avoids reprocessing in later steps.

Current scope: focus on stock, contracts and orders, prioritizing integration simplicity and governance clarity.

Access token generation

Mandatory entry point for all subsequent integration calls.

POST • 01 - Authentication • Get Access Token

Request (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'
```

Response (JSON)

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

Note: The access credentials are provided by CWS at the start of the integration. The token must be reused until it expires, to avoid unnecessary authentication calls.

api-docs.cws.digital

Batch update by SKU attribute

Recommended model when the product is created with the integration attribute already defined in its registration.

PUT • 02 - Stock and Price • Update or Create Stock in Batch

Request (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"}]
    }
  ]
}'
```

Response (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"}]
  }
]
```

Note: Identification by skuAttributeName/skuAttributeValue requires contracting the product catalog service, which must be requested from CWS. The skuAttributeValue comes with the product creation.
The partnerPartCode reappears in the contract flows; this value must be standardized and unique

api-docs.cws.digital

Update by composition and manufacturer

Useful alternative when the item's identification starts from the product's commercial metadata.

PUT • 02 - Stock and Price • Update or Create Stock in Batch

Request (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"}]
    }
  ]
}'
```

Response (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"
  }
]
```

Note: The partnerPartCode reappears in the contract flows; this value must be standardized and unique

Initial contract lookup

First step to locate existing rules before creating new records.

GET • 06 - Contracts • List Contracts

Request (cURL)

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

Response (JSON)


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


Note: The contract is optional: use it only when the operation has a customer-specific price table (assess the need). If you use it, prefer the customer's ERP code as the contract name.

api-docs.cws.digital

Contract creation

Use only when the customer's contract is not found in the lookup step.

POST • 06 - Contracts • Create Contract

Request (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" }
  ]
}'
```

Response (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]
}
```


Note: Recommendation: treat this route as an operational exception, triggered only when the initial lookup does not return a valid contract for the customer.

Adding products to the contract

Step to publish prices per item after the contract is found or created.

PUT • 06 - Contracts • Add Products to the Contract

Request (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 }
  ]
}'
```

Response (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
      }
    ]
  }
}
```

Note: partnerPartCode is the same identifier sent when linking the product to the stock. The service accepts up to 100 items per request.

api-docs.cws.digital

Orders webhook

Main mechanism to receive order creation and update events in real time.

13 - Callbacks and Webhooks • Orders Webhooks

Configuration and authentication

CWS provides an event-triggered webhook. Whenever an order is created or updated, a notification is sent to the customer's API with basic data or with the complete payload for management.

- **Domain URL**
- **Delivery endpoint**
- **Auth endpoint**
- **Authentication headers**

Auth response example

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

Operational summary

- 2 endpoints required
- event on creation and update
- option of a basic or complete payload
- best channel for real-time updates

webhook documentation

Order listing as fallback

Check routine to identify orders that may not have been integrated via event.

GET • 03 - Orders • List Store Orders

Request (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>'
```

Response (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"
    }
  }
]
```


Note: As a webhook fallback, this endpoint returns enough basic data to validate orders that were not captured automatically by the event service.

api-docs.cws.digital

Order status update

The transition must respect the order lifecycle, without skipping steps.

PUT • 03 - Orders • Update Order Status - Store

Request (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"
}'
```

Response (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 truncated for display]
```

Note: The integration must respect the official status order. This care avoids rejections and tracking inconsistencies on the portal.

api-docs.cws.digital