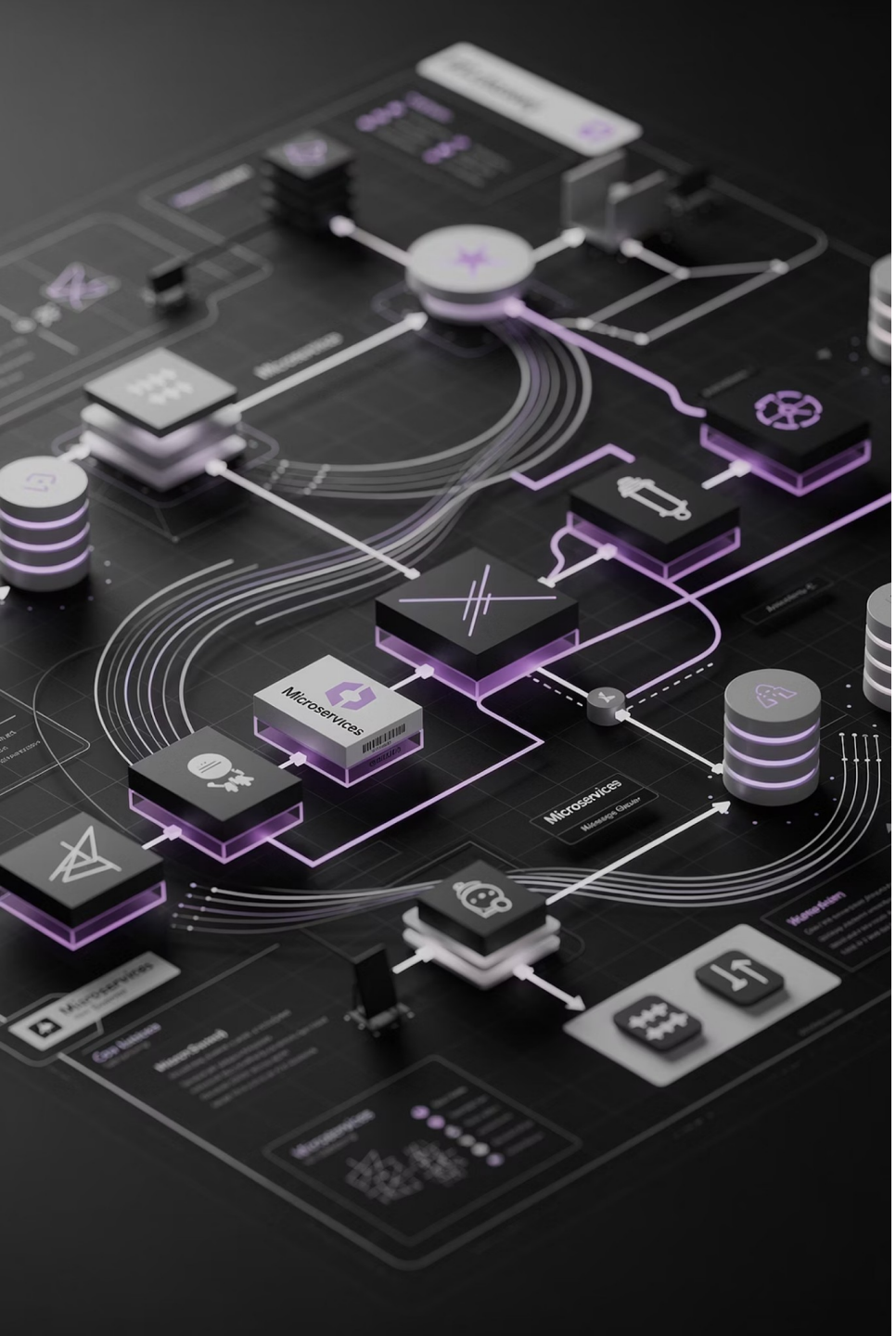

Integration Best Practices

CWS Platform API

Recommendations for a stable, high-performance integration that is ready for homologation

OVERVIEW

Why follow these practices

Correct use of the endpoints, call control and response handling ensure that the integration scales.

01

Authentication

Reuse the token until it expires and renew it only when necessary.

02

Logs and errors

Log the full response body and handle each HTTP code.

03

Stock and price

Planned initial load, batch submission and updates only to what changed.

04

Orders

Webhook as the main channel, listing as fallback and the status flow respected.

Before going to production

- Develop and test all flows in the sandbox environment (ws.sandbox.autorei.net).
- Document the flows in the homologation documentation template.
- Schedule the homologation meeting with CWS to validate the processes before the cutover.

Note: Each endpoint's contract, business rules and errors are in the official documentation. Questions: suporte-api@cws.digital. api-docs.cws.digital

When to request the token

One authentication per process, reused in all calls until it expires.

POST • 01 - Authentication • Get Access Token

Rules

- The access_token is valid for 12 hours (expires_in = 43199 seconds).
- Reuse the same token in all calls of the process.
- Authenticate again only when you receive a 401, with a single attempt per run, to avoid loops.
- Always send Authorization: Bearer <access_token>. Without the Bearer prefix, the API responds 401.

Example: new product in the ERP

- Use the token already obtained to send the product's stock.
- Do not authenticate again between calls of the same process.
- If the response is 401, generate a new token once and repeat the call.
- If the 401 persists, stop and log the error: likely a credential problem.

Note: Excessive authentications hurt the integration's performance and can be identified during homologation.

Log everything and handle each response

Save the request and the full response body: this speeds up the analysis of support tickets.

Code	When it happens	Recommended action
`200 with ERROR`	Batch submission: a rejected item comes with status ERROR and the reason in error	Always go through the list, even with HTTP 200, and handle each item
`400`	Invalid JSON or failure of the whole batch (e.g., mixed identification methods)	Fix the payload. The batch uses code/message/errorDescription instead of error
`401`	Token missing, without Bearer, malformed or expired	Renew the token once and repeat the call
`403`	Parent Account route with an unlinked branch (Invalid partnership)	Check the parent account–branch link and the partnerId
`404`	Resource does not exist or is outside the store's scope	Check the id used in the route
`422`	Business rule: status transition, required field, item limit	Read the reason in error or message and adjust the request

Note: Error formats vary by endpoint (error or code/message). Handle both and keep the original body in the log. api-docs.cws.digital

First load in production

The batch submission creates or updates the stock: there is no need to query the stock before sending.

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

01

Define the identification

productId, brand + mfrPartCode + composition or skuAttribute (the latter requires the product catalog service).

02

Send in batches

Up to 50 items per request, all with the same identification method.

03

Keep the mapping

Save the returned productId alongside the ERP code for the next submissions.

04

Handle the rejections

Items with status ERROR come back with the reason; fix and resend only those.

Store that already sells on the platform

Run the first load outside business hours and check beforehand which ERP products already have stock registered, to avoid quantity and price discrepancies during the cutover.

Note: The old sequence of product search → addition → update by stock id was replaced by the batch submission, which creates or updates in a single call.

api-docs.cws.digital

Batch submission rules

Behaviors that most often cause stock discrepancies or rejections.

Always a full submission

Optional fields that are not resent are erased (leadTimeDate, leadTimeByRequest, profitMargin, totalCost, targetMargin). Resend the whole item.

One method per batch

Mixing identification methods rejects the whole batch with 400 (mixed request types in list).

No repeated items

Items with the same warehouse and identifier are deduplicated; if the values differ, the item is not processed.

Date with time zone

leadTimeDate requires the full date and time with the -03:00 offset. The date alone rejects the whole request.

Lead time x virtual stock

leadTimeByRequest true does not accept leadTime > 0; virtual stock > 0 requires leadTime when leadTimeByRequest is false.

Unique code in the warehouse

partnerPartCodeUnique: true zeroes the warehouse stocks that already use the same partnerPartCode. Use it only when reusing codes.

Note: The warehouse (warehouseId) must belong to the token's store, or to the branch given in the Parent Account route; otherwise the whole batch is rejected.

api-docs.cws.digital

Updates in daily operation

A high call volume during business hours calls for small, targeted submissions.

During business hours

- Send only the products with an actual change in quantity or price.
- Group changes made close together into a single batch.

Full ERP update

- Schedule it outside business hours (early morning or night).
- Split the stock into batches of up to 50 items.

Batch identified by 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"
    }
  ]
}'
```

Note: productId has the highest precedence in identification and skips resolution by manufacturer or attribute, making the submission faster. api-docs.cws.digital

New products in the ERP

Stock is only created for products that exist in the catalog available to the store.

Recommended flow

- Send the new product through the same batch submission, with the identification method already adopted.
- With status CREATED, save the productId in the mapping.
- With status ERROR for product not found, confirm the product registration with CWS.

Products created by the store

- Create your own product only when it does not exist in the CWS catalog.
- Your own product is available only for your store's stock, not for other partners.
- In marketplaces, check the rules for displaying products created by third parties, so that the item is available for sale.

Note: Identification by skuAttributeName/skuAttributeValue requires the product catalog service, requested from CWS. api-docs.cws.digital

Receiving and updating orders

Real-time event as the main channel and periodic query as a safeguard.

01

Webhook

Receive order creation and updates. With the simplified payload, use the id to fetch the details.

02

Fallback

List orders by updatedFrom/updatedTo to capture what the event did not deliver. Up to 25 per page.

03

Status

Follow the official status flow. To approve payment, send APPROVED.

04

ERP

In READY, send supplierOrderId with the order number in your ERP.

Order ids

- orderId is the customer order (parent); id is the partner order, one per supplier.
- All detail and status routes use the partner order id.

Note: Details on transitions, required fields and cancellation reasons are in the Order Update Flow document. api-docs.cws.digital

Price contracts and Parent Account management

Optional features, used according to the operation's business model.

Contracts (optional)

- Use only when the operation has a customer-specific price table.
- List the contracts before creating a new one, to avoid duplicates.
- Add up to 100 products per request.
- A product repeated in the same submission is not processed: send each one once.

Parent Account Management

- Stock and contracts have Parent Account routes, with the branch's partnerId in the route.
- The branch must be linked to the token's parent account; without a link, 403.
- Orders follow the marketplace option: orders placed on the portal, without depending on a link.

Note: The complete examples of each route are in the CWS API Integration (Store) and CWS API Integration (Parent Account) materials. api-docs.cws.digital

Checklist for the homologation meeting

Items validated with CWS before the release to production.

✓ **Flows tested in the sandbox**

Authentication, stock, orders and, if used, contracts and webhook.

✓ **Documentation filled in**

Homologation documentation template with flows, endpoints and rules.

✓ **Token reused**

One authentication per process and a single renewal on 401.

✓ **Complete logs**

Request and response body saved for each error.

✓ **Initial load plan**

Defined strategy for the first stock load in production.

✓ **Fallback routines**

Periodic order query to cover webhook failures.

Contact: To schedule the homologation or ask questions, contact suporte-api@cws.digital.