

Alice API: migrating from V2 (legacy) to V4

Integrator transition guide · October 2026 · AlicePOS

The legacy V2 API (XML) is retired. All integrations must use the V4 API.

1. What changes at a glance

Topic	V2 (legacy)	V4 (new)
Base URL	Legacy endpoint (single URL for every request)	<code>https://ycoko9c7fg.execute-api.us-east-1.amazonaws.com/v4</code> , one URL per resource
Format	XML. Every call is a <code>POST</code> carrying a <code><request><type>...</type></code> envelope	JSON REST. The HTTP verb (<code>GET</code> , <code>POST</code> , <code>PUT</code> , <code>DELETE</code>) and the path define the operation
Authentication	<code><credentials></code> block: <code>apikey</code> , <code>timestamp</code> , HMAC-SHA256 signature	HTTP header <code>x-api-key: <V4 key></code> . No signature to compute. Your existing V2 credentials are reused (see below).
Filters	<code><filters><filter><field/><like/> <param/></code>	Standard query-string parameters (e.g. <code>?currency=CAD&name=shirt</code>)
Pagination	Full result set returned in one response	<code>offset</code> and <code>limit</code> on list endpoints. Loop until a page returns fewer rows than <code>limit</code> .
Incremental sync	<code>whenmod</code> / <code>when</code> UNIX timestamps	<code>start</code> / <code>end</code> dates (<code>YYYY-MM-DD</code>) where supported
Booleans	<code>t</code> / <code>f</code> strings	JSON <code>true</code> / <code>false</code>
Translations	Separate fields: <code>namefr</code> , <code>nameen</code> , <code>namesp</code> ...	Array of <code>{ "code": "fr", "value": "..." }</code> (e.g. <code>names[]</code> , <code>descriptions[]</code>)
Errors	<code><error></code> node in an HTTP 200 response	Real HTTP status codes: 401, 403, 404, 500...
Connectivity test	<code>Date</code> request	<code>GET /tools</code> returns the company ID linked to your key

Your API key: no new key needed. V4 accepts your existing V2 credentials, combined into a single key. Join your V2 access key and your V2 secret key, in that order, with no separator, and send the result in the `x-api-key` header.

V2 (old)	V4 (new)
Access: XXXXXXXX	Key: XXXXXXXXYYYY
Secret: YYYY	Header: x-api-key: XXXXXXXXYYYY

The secret is now sent with every request, so call V4 over HTTPS only and never write the key to logs or client-side code.

2. Endpoint mapping (1 to 1)

DIRECT V4 covers the V2 call. **PARTIAL** V4 covers it, but a filter or field differs (see notes). **NO EQUIVALENT** not in V4 yet. Contact AlicePOS if you depend on it.

V2 §	V2 transaction (type)	V4 endpoint	Status	Migration notes
3.1	Date	GET /tools (connectivity only)	NO EQUIVALENT	V4 does not return the server time. Use your own UTC clock. V4 date filters take calendar dates, so second-level sync is no longer needed.
3.2	Category	GET /categories?types=CATEGORY GET /categories/{categoryId}	PARTIAL	The <code>sku</code> filter becomes the path parameter <code>{categoryId}</code> (e.g. <code>class_442m</code>). No <code>whenmod</code> filter: re-read the full list (it is small).
3.3	Product	GET /products?currency=CAD GET /products/{productId}?currency=CAD GET /products/upcs/{upc}	PARTIAL	<code>currency</code> is required . <code>whenmod</code> becomes <code>start/end</code> . <code>upccode</code> becomes <code>/products/upcs/{upc}</code> or <code>upc[class_xxx]=value</code> . <code>name</code> stays <code>name</code> . <code>productfilter / online-only (isOnline)</code> is not yet implemented , so filter on your side for now. See section 3 for field renames.
3.4.1	Image	GET /products/{productId}/images	DIRECT	Take the first image (display order 0). Image metadata is also in the <code>images[]</code> array of the product response.
3.4.2	MultiImage	GET /products/{productId}/images GET /images?name=...	DIRECT	One call per product, or the global <code>/images</code> list (paginated, filter by <code>name</code>).
3.5	Inventory	GET /products/inventories GET /products/{productId}/inventories GET /products/upcs/{upc}/inventories	PARTIAL	Bulk list is paginated. <code>expand=true</code> adds serialized quantity details. The <code>when</code> filter is not available: poll the bulk list on a schedule.

V2 §	V2 transaction (type)	V4 endpoint	Status	Migration notes
3.6	Promotions	GET /promotions GET /promotions/{promotionId} GET /promotions/products?currency=CAD	DIRECT	status takes names (ONGOING , UPCOMING , EXPIRED , INACTIVE) instead of 0–3. /promotions/products replaces walking the promolinks tree. Per-product promos are also in products[].promotions[] .
3.7	Tax	GET /categories?types=TAX	DIRECT	Same tax class IDs (class_xx), so existing mappings stay valid.
3.8	SystemApplicationZone	(none)	NO EQUIVALENT	Store the system's zone ID as configuration on your side (one value per store). AlicePOS can provide it.
3.9	ApplicationZoneList	(none)	NO EQUIVALENT	Same approach as 3.8. The zone list rarely changes.
3.10	GLExportData	(none)	NO EQUIVALENT	The accounting posting report is not exposed in V4. Contact AlicePOS.
3.11	Services	GET /services GET /services/{serviceId} GET /categories?types=SERVICE_CATEGORY	DIRECT	Paginated. Filters: name , start / end . Service types and protection services are also available through types=SERVICE_REPAIR,SERVICE_PROTECTION .
3.12	ShippingAttributes	GET /categories?types=ATTRIBUTE GET /products/{productId}/attributes	PARTIAL	Find the size and weight attributes in the attribute list once, keep their IDs, then read each product's values from /products/{id}/attributes .
3.13	UPCList	GET /upcs GET /categories?types=UPC	DIRECT	UPC class IDs are the keys used by the upc[class_xxx] product filter.
3.14	EntityList	GET /contacts (mode 1) GET /companies (mode 2) GET /contacts/{contactsIds}	PARTIAL	Contacts and companies are separate endpoints, so call both for mode 0. search matches name, phone and email. No lastsync filter. Credit balance and credit limit fields need to be confirmed with AlicePOS.
3.15	PromotionEnded	GET /promotions/products? status=EXPIRED&startDate=...	PARTIAL	Returns products of expired promotions in the date window. Split new/used with the promotion state field.
4.1	Transaction_Add	POST /transactions GET /transactions/{transactionId} PUT /transactions/{transactionId}	DIRECT	JSON body, camelCase fields (mapping in section 3). The rules are unchanged: send to a store, never a master; a customer is created if missing. V4 adds read-back (GET) and update (PUT).
6.1	GLAccountList	(none)	NO EQUIVALENT	Not exposed in V4. Contact AlicePOS.

V2 §	V2 transaction (type)	V4 endpoint	Status	Migration notes
6.2	GLSingleInvoice	GET /invoices/{invoiceId}? include=items,payments	PARTIAL	{invoiceId} accepts the invoice ID, number (240115-1) or reference. Returns the invoice detail, not the GL debit/credit lines.
6.3	GLSummaryData	GET /invoices?start=...&end=...	PARTIAL	Same as 6.2: invoice list for the period, paginated, with no GL account breakdown.

3. Field mapping for the two most used calls

Product (V2 `Product` → V4 `/products`)

V2 field	V4 field
<code>sku</code> (<code>pric_xxx</code>)	<code>id</code>
<code>alicecode</code>	<code>aliceCode</code>
<code>name</code> , <code>namefr/en/sp</code>	<code>names[]</code> { <code>code</code> , <code>value</code> }
<code>desc</code> , <code>descfr/en/sp</code>	<code>descriptions[]</code> { <code>code</code> , <code>value</code> }
<code>enable</code> (t/f)	<code>isEnabled</code> (bool)
<code>taxable</code> (t/f)	<code>isTaxable</code> (bool)
<code>parentsku</code>	<code>parentId</code>
<code>buynew</code> / <code>buyused</code> / <code>sellnew</code> / <code>sellused</code>	<code>buyNew</code> / <code>buyUsed</code> / <code>sellNew</code> / <code>sellUsed</code>
<code>categoryid</code> , <code>category</code>	<code>categoryId</code> , <code>categoryName</code>
<code>upccodes</code> / <code>upc</code>	<code>upcs[]</code> (details: <code>/products/{id}/upcs</code>)
<code>promoprice</code> , <code>productspecificpromotion</code>	<code>promotions[]</code> { <code>id</code> , <code>linkId</code> , <code>price</code> , <code>state</code> , <code>trigger</code> , <code>action</code> , <code>start</code> , <code>end...</code> }
<code>picture</code> , <code>picturewhenmod</code>	<code>images[]</code> { <code>id</code> , <code>name</code> , <code>lastModified...</code> }
<code>inventoryqty</code>	Separate call: <code>/products/{id}/inventories</code>
<code>attributes</code> (incl. <code>sellable</code>)	Separate call: <code>/products/{id}/attributes</code>

Order (V2 `Transaction_Add` → V4 `POST /transactions`)

V2 field	V4 field
<code>order.id</code>	<code>id</code>
<code>order.currency</code>	<code>currency</code>
<code>discount_total</code> / <code>shipping_total</code> / <code>shipping_tax</code>	<code>discountTotal</code> / <code>shippingTotal</code> / <code>shippingTax</code>
<code>total</code> / <code>total_tax</code>	<code>total</code> / <code>totalTax</code>
<code>customer_id</code> / <code>shipto_id</code>	<code>customerId</code> / <code>shipToId</code>
<code>order.taxes.tax[]</code>	<code>orderTaxes[]</code> { <code>class</code> , <code>rate</code> , <code>amount</code> }
<code>alice_client_id</code> , <code>reference</code>	<code>aliceClientId</code> , <code>reference</code>
<code>invoicenote</code> / <code>internalnote</code>	<code>invoiceNote</code> / <code>internalNote</code>
<code>billing</code> / <code>shipping</code>	<code>billingInfo</code> / <code>shippingInfo</code>
<code>first_name</code> / <code>last_name</code>	<code>firstname</code> / <code>lastname</code>
<code>address_1</code> / <code>address_2</code> / <code>postcode</code>	<code>line1</code> / <code>line2</code> / <code>zip</code>
<code>products.product[].product_id</code>	<code>products[].id</code>
<code>sku</code> , <code>quantity</code> , <code>total</code> , <code>total_tax</code> , <code>price</code> , <code>used</code> , <code>note</code>	<code>sku</code> , <code>quantity</code> , <code>total</code> , <code>totalTax</code> , <code>price</code> , <code>used</code> , <code>note</code>
<code>promolinkid</code> / <code>promoid</code>	<code>promoLinkId</code> / <code>promoId</code>
(new)	<code>deposit</code> , <code>billingInfo.paymentMethod</code> , <code>products[].cost</code>

4. Migration steps

1. **Build your V4 key.** For each connected store, join the V2 access key and the V2 secret key (access + secret, no separator). See section 1.
2. **Test connectivity.** Call `GET /tools` with header `x-api-key`. It must return the expected company ID.
3. **Replace the transport layer.** Remove the XML envelope and HMAC signature code. Send JSON over HTTPS with the key header.
4. **Rewrite each call** using the table in section 2. Add pagination (`offset / limit`) to every list call.
5. **Remap fields** using section 3 (camelCase, booleans, translation arrays).
6. **Check every** **NO EQUIVALENT** **or** **PARTIAL** **item** you use, and tell AlicePOS right away if it blocks you.
7. **Validate on a test store**, especially `POST /transactions`, which writes to the database.
8. **Switch production** and let AlicePOS know once it is done.

Full interactive V4 reference (Swagger): <https://ycoko9c7fg.execute-api.us-east-1.amazonaws.com/v4/api-docs>

Technical contact: **[NAME / EMAIL]**