

.cart



The .cart File Format

An open, interoperable, vendor-neutral standard for ecommerce shopping carts

Version	0.1.0 — Draft / Request for Comments
Cart version	0.1
File extension	.cart
Encoding	JSON (UTF-8)
Media type	application/vnd.cart+json
Author	Daniel Rosehill
License	MIT
Canonical source	github.com/danielrosehill/Dot-Cart

1. Purpose

.cart is an open, vendor-neutral file format for describing the contents of an ecommerce shopping cart. It exists so that a cart can be treated as a **portable document** rather than ephemeral session state locked inside a single website's cookies.

The format is designed from the ground up for **agentic AI interoperability**: an AI agent (browser-based or API-based, e.g. via an MCP server) should be able to export a cart from any vendor into a .cart file, and then import, compare, merge, re-price, or transform that file — for example into a purchase order or request for quotation.

Design goals

1. **Human-readable and machine-parseable** — plain JSON, no binary blobs.
2. **Vendor-neutral** — no field assumes a specific platform (Shopify, Magento, a custom industrial-supplier storefront, etc.).
3. **Tax-aware** — prices are expressed both before and after VAT/sales tax, because B2B and B2C contexts need different views of the same cart.
4. **Lossless where possible, graceful where not** — unknown vendor-specific data is preserved under `extensions` rather than discarded.
5. **Self-describing** — a .cart file carries enough context (merchant, currency, capture time, source URL) to be meaningful on its own, on any device, at any later date.

2. File conventions

- A .cart file **MUST** contain a single JSON object (the **cart document**).
- Field names are `snake_case`.
- Timestamps are ISO 8601 strings with timezone (e.g. `2026-07-24T09:30:00+03:00`).
- Currency codes are ISO 4217 (e.g. `ILS`, `USD`, `EUR`).
- Country codes are ISO 3166-1 alpha-2 (e.g. `IL`, `US`).
- Monetary amounts are JSON numbers in major units of the cart's currency (e.g. `118.00` means 118 shekels when currency is `ILS`). Producers **SHOULD** emit at most the number of decimal places conventional for the currency.
- VAT/tax rates are decimal fractions (`0.18` = 18%).
- Consumers **MUST** ignore fields they do not recognize (forward compatibility).

2.1 Conformance profiles — Core vs Expanded

A .cart file is designed to be useful at two levels of richness. Both are the same format and the same schema; a profile is just a statement of *how much* of the format a producer filled in or a consumer relies on.

Core profile — the minimum for a file to mean “here is a cart and what it costs”. A Core file **MUST** contain:

- `cart_version`, `id`, `created_at`
- `merchant.name`
- `currency`
- `items[]`, each with `name`, `quantity`, `unit_price_incl_tax`, `line_total_incl_tax`
- `totals.grand_total` and `totals.amount_pending_payment`

Any conforming consumer (a viewer, a price tracker, a “reopen my cart” tool) can rely on the Core fields being present.

Expanded profile — everything else the format defines: full provenance (`source`), merchant detail, the tax breakdown (`*_excl_tax` / `tax`), shipping methods, payment (including card and method), and the richer item fields (`sku`, `gtin`, `mpn`, `price_breaks`, `min_order_quantity`, `lead_time_days`, etc.). Expanded fields are what make a cart useful for **procurement** — cross-vendor comparison, RFQs, and purchase orders.

Producers **SHOULD** emit as much of the Expanded profile as they can capture, and **MUST** at least satisfy Core. Consumers **MUST** degrade gracefully when only Core fields are present. A future `profile` hint field may be added to let a producer declare its intent explicitly; until then, presence of fields is authoritative.

2.2 Observed vs. computed values, and multi-currency

Some figures in a `.cart` file are **observed** — read directly from the merchant — and some are **computed** by the producing tool (a currency conversion, for example). Where a value is computed rather than observed, the producer SHOULD flag it so a consumer knows not to treat it as authoritative. The current mechanism is a sibling boolean such as `exchange_rate_calculated`; a general computed marker may be standardized in a later version.

By default a cart is **single-currency**: every amount is in the top-level currency, and `multi_currency` is `false`. Only when `multi_currency` is `true` may individual items carry their own currency, `exchange_rate`, and `exchange_rate_calculated`. Even then, totals remain expressed in the cart's top-level currency.

3. Top-level structure

Field	Type	Required	Description
<code>cart_version</code>	string	✓	Version of this spec the file conforms to ("0.1").
<code>id</code>	string	✓	Unique identifier for this cart snapshot (UUID recommended).
<code>created_at</code>	string (ISO 8601)	✓	When the cart was captured/exported.
<code>updated_at</code>	string (ISO 8601)	—	Last modification of this file.
<code>expires_at</code>	string (ISO 8601)	—	When the captured prices/quote stop being valid, if the merchant states it. Signals that a comparison or PO built from this file may be stale.
<code>locale</code>	string	—	BCP 47 locale the cart was presented in (e.g. "he-IL", "en-us"). Helps interpret names and formatting.
<code>source</code>	object	—	Where and how the cart was captured. See §3.1.
<code>transaction_type</code>	string	—	"b2c" (default) or "b2b". When "b2b", the buyer object and both parties' VAT details become relevant.
<code>merchant</code>	object	✓	The seller the cart belongs to. See §3.2.
<code>buyer</code>	object	—	The purchasing party (chiefly for B2B). See §3.9.
<code>currency</code>	string	✓	ISO 4217 currency for all amounts in the file.
<code>multi_currency</code>	boolean	—	<code>false</code> by default. When <code>true</code> , individual items may carry their own currency and exchange-rate fields (see §3.4). Totals stay in the top-level currency.
<code>tax</code>	object	—	Cart-level tax context. See §3.3.
<code>items</code>	array	✓	Line items. See §3.4.
<code>shipping</code>	object	—	Shipping methods and destination. See §3.5.
<code>totals</code>	object	✓	Cart-level monetary totals. See §3.6.
<code>payment</code>	object	—	Payment status/expectations. See §3.7.
<code>notes</code>	string	—	Free-text notes (human or agent authored).
<code>extensions</code>	object	—	Namespaced vendor/tool-specific data. See §3.8.

3.1 source

Provenance of the snapshot — important for agents that need to re-open or re-validate the cart later.

Field	Type	Description
<code>platform</code>	string	Ecommerce backend / storefront platform powering the merchant, if known or detected (e.g. "shopify", "woocommerce", "magento", "bigcommerce", "custom").
<code>platform_detected</code>	boolean	<code>true</code> if <code>platform</code> was auto-detected by the capturing agent (e.g. from markup, headers, or JS globals) rather than supplied by a human.
<code>platform_confidence</code>	number	Detector confidence in <code>platform</code> , 0–1, when <code>platform_detected</code> is <code>true</code> .
<code>cart_url</code>	string	URL of the live cart, if one exists.
<code>captured_by</code>	string	Tool/agent that produced the file (e.g. "cart-exporter-mcp/0.3").
<code>capture_method</code>	string	One of "browser_agent", "api", "manual", "ocr", "other".

3.2 merchant

Field	Type	Required	Description
name	string	✓	Merchant/vendor display name.
legal_name	string	—	Registered company name, if different.
website	string	—	Merchant homepage URL.
merchant_id	string	—	Merchant identifier (company/VAT registration number, etc.).
vat_id	string	—	Seller VAT/tax registration number (the seller side of a B2B transaction).
vat_issuing_authority	string	—	Authority/country that issued the seller's VAT ID (e.g. "IL", "DE").
country	string	—	Seller's country (ISO 3166-1 alpha-2); defaults to <code>address.country</code> when present.
compliance	array of strings	—	Data-protection / security standards the merchant attests to — useful for (government) procurement, e.g. ["soc2", "iso_27001", "gdpr"].
global_merchant_id	string	—	Reserved / placeholder. A future universal merchant identifier; no ecosystem registry exists yet. Populate only against an agreed scheme.
address	object	—	Postal address: street, city, region, postal_code, country.
contact	object	—	email, phone.

3.3 tax

Field	Type	Description
scheme	string	"vat", "sales_tax", OR "none".
default_rate	number	Default rate applied to items unless overridden per item.
prices_include_tax	boolean	Whether the merchant's displayed prices include tax.

3.4 items[]

Each element describes one line item.

Field	Type	Required	Description
sku	string	✓*	Merchant SKU / catalog number. (*Required if the merchant exposes one.)
name	string	✓	Short product name.
description	string	—	Longer description.
variant	string	—	Human-readable variant label (e.g. "Blue / XL").
options	array	—	Structured variant selections: [{ "name": "Color", "value": "Blue" }, ...].
quantity	number	✓	Quantity in the cart.
unit	string	—	Unit of measure ("each", "box", "m", "kg" ...). Defaults to "each".
unit_price_excl_tax	number	✓	Price per unit before VAT/tax.
unit_price_incl_tax	number	✓	Price per unit after VAT/tax.
rrp_excl_tax	number	—	Merchant's regular/reference price (RRP) per unit before tax, when the item is discounted from a higher list price.
rrp_incl_tax	number	—	Regular/reference price (RRP) per unit after tax. Discount percentage is measured against this.
currency	string	—	Item currency — only when cart multi_currency is true and it differs from the cart currency.
exchange_rate	number	—	Rate used to convert this item's currency to the cart currency.
exchange_rate_calculated	boolean	—	true if the rate was computed by the producing tool rather than quoted by the merchant (see §2.2).
tax_rate	number	—	Item-level rate if it differs from tax.default_rate.
line_total_excl_tax	number	✓	quantity × unit_price_excl_tax (after line discounts).
line_total_incl_tax	number	✓	Line total after VAT/tax.
discount	object	—	Discount applied to this line (already reflected in line totals). See below.
product_url	string	—	Canonical product page URL.
image_url	string	—	Product image URL.
manufacturer	string	—	Brand/manufacturer name.
mpn	string	—	Manufacturer part number (distinct from merchant SKU).
gtin	string	—	Barcode (EAN/UPC), when known — key for cross-vendor comparison.
availability	string	—	"in_stock", "backorder", "preorder", "unknown".
min_order_quantity	number	—	Minimum orderable quantity (MOQ), common in B2B.
lead_time_days	number	—	Quoted lead time in days for backordered/made-to-order items.
price_breaks	array	—	Quantity-tiered pricing offered for this item — essential for B2B comparison and RFQs. See below.
notes	string	—	Free-text line notes.

Each entry in price_breaks:

Field	Type	Required	Description
min_quantity	number	✓	Quantity at which this tier's price applies.
unit_price_excl_tax	number	—	Per-unit price at this tier, before tax.
unit_price_incl_tax	number	—	Per-unit price at this tier, after tax.

The discount object:

Field	Type	Description
description	string	Human-readable label (“Summer sale”, “Contract price”).
amount	number	Absolute discount per line, in cart currency (already reflected in line totals).
percentage	number	Discount as a decimal fraction of the RRP (0.15 = 15%).
type	string	"automatic", "manual", "coupon", "negotiated", Or "promotion" — distinguishes an agent/human manually applied discount from an automatic one.

3.5 shipping

Field	Type	Description
ship_to	object	Destination address (same shape as merchant.address, plus optional name and address_type).
ship_to.address_type	string	"residential", "business", "warehouse", Or "other". Affects carrier options and, sometimes, surcharges.
incoterms	string	Delivery terms for (international) B2B, e.g. "EXW", "FOB", "CIF", "DAP", "DDP" (ICC Incoterms® 2020).
selected_method_id	string	id of the chosen method from available_methods.
available_methods	array	All methods the merchant offered at capture time.

Each entry in available_methods:

Field	Type	Description
id	string	Stable identifier within this file (e.g. "courier-48h").
name	string	Display name (“Courier, 2 business days”).
cost_excl_tax	number	Shipping cost before VAT/tax.
cost_incl_tax	number	Shipping cost after VAT/tax.
estimated_delivery	string	Free-text or ISO 8601 estimate.

3.6 totals

All amounts in currency. Consumers SHOULD verify arithmetic on import and flag mismatches rather than silently correct them.

Field	Type	Required	Description
items_subtotal_excl_tax	number	✓	Sum of line totals before tax, before shipping.
items_subtotal_incl_tax	number	✓	Sum of line totals after tax, before shipping.
shipping_excl_tax	number	—	Cost of the selected shipping method, before tax.
shipping_incl_tax	number	—	Cost of the selected shipping method, after tax.
discount_total	number	—	Cart-level discounts (already reflected in the grand total).
tax_total	number	✓	Total VAT/tax across items and shipping.
grand_total	number	✓	Final amount: items + shipping – discounts, tax included.
amount_pending_payment	number	✓	Amount currently awaiting payment (usually equals grand_total; may differ with deposits/partial payments).

3.7 payment

Field	Type	Description
status	string	"cart" (not yet ordered), "quote_requested", "po_issued", "awaiting_payment", "partially_paid", "paid".
method	string	The payment method the buyer has selected or intends to use — a single value drawn from <code>methods_accepted</code> (e.g. "credit_card", "bank_transfer", "purchase_order").
terms	string	Payment terms if known (e.g. "net_30", "net_60", "prepaid", "cod").
methods_accepted	array of strings	Methods the merchant offers, e.g. ["credit_card", "bank_transfer", "purchase_order"].
reference	string	Quote/PO/invoice reference once one exists.
card	object	The payment card the buyer intends to use. Sensitive — see §8. See below.

The card object:

Field	Type	Description	Sensitivity
cardholder_name	string	Name printed on the card.	Sensitive PII
is_business_card	boolean	<code>true</code> if a business/corporate card, <code>false</code> for a personal card. Useful for splitting business vs. personal purchasing.	Normal
brand	string	Card network if known (e.g. "visa", "mastercard", "amex").	Normal
last4	string	Last four digits, when a masked reference is needed to disambiguate which card	Sensitive PII

⚠ **Never store full card data.** A `.cart` file MUST NOT contain a full primary account number (PAN), CVV/CVC, PIN, or full expiry beyond what is needed for reference. Producers SHOULD store no more than a masked `last4`. Handling full card data would put the file in scope for PCI-DSS and is out of scope for this format. See §8.

3.8 extensions

A place for data that doesn't fit the core spec, keyed by a namespace chosen by the producing tool (reverse-DNS or tool name recommended):

```
"extensions": {  
  "com.example-supplier": { "customer_tier": "gold", "internal_cart_id": "8842-A" }  
}
```

Consumers MUST preserve `extensions` content they don't understand when re-writing a file.

3.9 buyer

Present chiefly for **B2B** transactions (`transaction_type`: "b2b"). Identifies the purchasing party so a `.cart` can be turned into a compliant purchase order, RFQ, or reverse-charge VAT document (see §9).

Field	Type	Description
business_name	string	Legal name of the purchasing organization.
vat_id	string	Buyer's VAT/tax registration number.
vat_issuing_authority	string	Authority/country that issued the buyer's VAT ID.
country	string	Buyer's country (ISO 3166-1 alpha-2).
contact	object	Optional name, email, phone for the purchaser.

For a b2c cart this object is normally absent. Buyer identifiers are personal / business data — see §8. Together, merchant (seller) and buyer supply both sides of a transaction — including each party's VAT ID and issuing authority — which is what reverse-charge and cross-border VAT documents require.

4. Intended operations

The format is designed to make these agentic operations trivial:

- **Export** — a browser agent or MCP server serializes a live cart to `.cart`.
- **Import / restore** — re-populate a cart on the same merchant from a file.
- **Compare** — diff two `.cart` files (same BOM across vendors; same vendor across time) on price, availability, and shipping.
- **Transform** — generate a purchase order, RFQ, or BOM document from a file, mapped onto an open business-document standard (see §9).
- **Merge / split** — combine carts or split one across vendors.

5. Validation

A JSON Schema for v0.1 is provided at [schema/cart.schema.json](#). A file is a valid `.cart` document if it validates against the schema; the prose in this document governs where the two disagree.

6. Versioning

The spec follows semantic versioning. Within a major version, changes are additive only: new optional fields may appear, existing fields will not change meaning or be removed. Consumers should accept any `cart_version` sharing their supported major version.

7. Open questions (feedback welcome)

- Money as decimal numbers vs. strings vs. integer minor units.
- Multi-currency carts (marketplaces quoting lines in different currencies).
- Signed/attested carts — letting a merchant cryptographically vouch that a `.cart` reflects a real quote.
- A companion `.order` format for the post-checkout lifecycle.

Candidate fields under consideration for the Expanded profile (feedback wanted on which are worth standardizing vs. leaving to extensions):

- `coupons / promo_codes` — applied discount codes, separate from computed discount amounts.
- Free-shipping / minimum-order thresholds the merchant advertised.
- `item.hs_code` and `item.country_of_origin` — customs data for cross-border B2B.
- Per-item weight / cart estimated_weight — for freight estimation.
- A profile hint field so a producer can declare "core" vs "expanded" explicitly (see §2.1).

8. Data sensitivity & PII

A `.cart` file is a portable document that may be emailed, synced, stored, or handed to an AI agent. Producers and consumers **MUST** be aware that some fields carry **personally identifiable information (PII)** or otherwise sensitive data, while most do not. The classification below is normative guidance, not a security guarantee.

Sensitivity	Fields	Handling
Normal (not PII)	merchant.*, currency, tax.*, items[].* (SKU, name, quantity, prices, GTIN, etc.), shipping.available_methods[].*, totals.*, payment.status/terms/methods_accepted/reference, payment.card.is_business_card/brand	shared and stored freely. This is the bulk of a cart.
Personal data	merchant.contact.* (when a named person), buyer.* (business name, VAT ID, contact), shipping.ship_to.* (name + address), source.cart_url (may embed a session token)	reated as personal/business data under GDPR/similar. Minimize, and avoid sharing beyond the intended recipient.
Sensitive PII	payment.card.cardholder_name, payment.card.last4	Include only when genuinely required. Never persist longer than needed.
Prohibited	Full card PAN, CVV/CVC, PIN, full expiry	NOT appear in

Guidance for implementers:

1. **Data minimization.** Omit sensitive and personal fields unless the use case requires them. A cart intended for price comparison needs no cardholder data at all.
2. **Redaction on export.** Tools that export or forward a .cart file SHOULD offer to strip the *Personal data* and *Sensitive PII* fields, leaving a shareable, non-personal document.
3. **Transport & storage.** When a file does contain personal or sensitive fields, it SHOULD be encrypted at rest and in transit.
4. **Schema annotation.** Fields carrying personal or sensitive data are tagged in the JSON Schema with an x-sensitivity annotation ("personal" or "sensitive") so tools can enforce the above programmatically.

9. Transformations: purchase orders & RFQs

Turning a captured cart into a business document is a primary use case, and a natural fit for an MCP server or agent. .cart defines two standardized transformations so that different tools produce compatible output:

- .cart → **purchase order (PO)** — the buyer commits to purchase the cart's line items from the merchant.
- .cart → **request for quotation (RFQ)** — the buyer asks one or more merchants to quote the cart's line items; prices in the source cart become *indicative* and the merchant is expected to return firm prices.

9.1 Interoperability baseline

.cart is the *capture* format; a PO or RFQ is a *business document* with its own established standards. Rather than invent new document formats, .cart transformations SHOULD target an existing open standard so output flows into procurement systems:

Standard	Body	PO document	RFQ document
UBL 2.1 (recommended baseline)	OASIS	Order	RequestForQuotation (+ Quotation for the response)
UN/EDIFACT	UN/CEFACT	ORDERS	REQOTE (+ QUOTES)
PEPPOL BIS (built on UBL)	OpenPEPPOL	Order transaction	—

On the ICC. The International Chamber of Commerce standardizes **Incoterms**® (delivery terms — already carried in `shipping.incoterms`) and publishes model international sale contracts, but it does **not** publish an electronic PO/RFQ *document schema*. For the documents themselves, **OASIS UBL** and **UN/CEFACT** are the appropriate baselines; ICC Incoterms slot into the `DeliveryTerms` of those documents.

9.2 Field mapping

Both transformations draw on the same .cart fields. The mapping below uses UBL element names as the canonical target (UN/CEFACT and PEPPOL have direct equivalents).

.cart field	PO / RFQ target (UBL)	Notes
buyer	BuyerCustomerParty	business_name→PartyName; vat_id+vat_issuing_authority→PartyTaxScheme/ CompanyID; country→PostalAddress/ Country.
merchant	SellerSupplierParty	Same party mapping on the seller side.
id / reference	ID / OrderReference	Document identity.
created_at	IssueDate	
expires_at	ValidityPeriod (RFQ) / Delivery/RequestedDeliveryPeriod	Drives quote validity in an RFQ.
currency	DocumentCurrencyCode	
items[]	OrderLine / RequestForQuotationLine	One per cart line.
item.sku	SellersItemIdentification/ID	
item.mpn / item.gtin	ManufacturersItemIdentification / StandardItemIdentification (GTIN)	GTIN is the cross-vendor key.
item.name / description	Item/Name / Description	
item.quantity / unit	Quantity (@unitCode)	
item.unit_price_excl_tax	Price/PriceAmount	Omitted or marked in- indicative in an RFQ — the merchant fills it.
item.price_breaks	—	Carry as RFQ context; the quote returns firm tiers.
item.tax_rate / tax	TaxTotal / ClassifiedTaxCategory	
totals.grand_total	LegalMonetaryTotal/PayableAmount	PO only; an RFQ has no committed total.
payment.terms	PaymentTerms	
shipping.incoterms	Delivery/DeliveryTerms	ICC Incoterms® code.
shipping.ship_to	Delivery/DeliveryLocation/Address	

9.3 PO vs RFQ differences

- **RFQ** — item prices, totals, and payment are treated as indicative or omitted; the emphasis is on identifiers, quantities, delivery terms, and expires_at (how long a quote is wanted for). Producing an RFQ from a cart lets a buyer send the *same* basket to several merchants and compare returned quotations.
- **PO** — prices and totals are authoritative and become a commitment; buyer, payment.terms, and delivery terms must be complete.

A reference implementation of both transformations is planned as part of the .cart MCP server.

Appendix A — Data dictionary

Every field defined by the format in one flat table, keyed by dotted path (array elements use the `[]` suffix). Generated from the canonical `spec/data-dictionary.json`; the JSON Schema remains authoritative for validation. **Profile** — `core` = minimum Core profile, `expanded` = everything else (§2.1). **Sensitivity** — `normal` (not PII), `personal` (GDPR-relevant), `sensitive` (sensitive PII); see §8.

Path	Type	Req.	Profile	Sens.	Description
<code>cart_version</code>	string	yes	core	normal	Version of this spec the file conforms to ("0.1").
<code>id</code>	string	yes	core	normal	Unique identifier for this cart snapshot (UUID recommended).
<code>created_at</code>	string	yes	core	normal	When the cart was captured / exported (ISO 8601 with time-zone).
<code>updated_at</code>	string	no	expanded	normal	Last modification of this file.
<code>expires_at</code>	string	no	expanded	normal	When the captured prices / quote stop being valid, if the merchant states it.
<code>locale</code>	string	no	expanded	normal	Locale the cart was presented in (e.g. "he-IL", "en-US"). Helps interpret names and formatting.
<code>transaction_type</code>	string	no	expanded	normal	"b2c" (default) or "b2b". When "b2b", the buyer object and both parties' VAT details become relevant. Values: b2c, b2b.
<code>source</code>	object	no	expanded	normal	Where and how the cart was captured (provenance).
<code>source.platform</code>	string	no	expanded	normal	Ecommerce backend / storefront platform powering the merchant, if known or detected (e.g. "shopify", "woocommerce", "magento", "custom").
<code>source.platform_detected</code>	boolean	no	expanded	normal	true if platform was auto-detected by the capturing agent rather than supplied by a human.
<code>source.platform_confidence</code>	number	no	expanded	normal	Detector confidence in platform, 0-1, when <code>platform_detected</code> is true.
<code>source.cart_url</code>	string	no	expanded	personal	URL of the live cart, if one exists. May embed a session token.
<code>source.captured_by</code>	string	no	expanded	normal	Tool / agent that produced the file (e.g. "cart-exporter-mcp/0.3").
<code>source.capture_method</code>	string	no	expanded	normal	How the cart was captured. Values: <code>browser_agent</code> , <code>api</code> , <code>manual</code> , <code>ocr</code> , <code>other</code> .
<code>merchant</code>	object	yes	core	normal	The seller the cart belongs to.
<code>merchant.name</code>	string	yes	core	normal	Merchant / vendor display name.
<code>merchant.legal_name</code>	string	no	expanded	normal	Registered company name, if different.
<code>merchant.website</code>	string	no	expanded	normal	Merchant homepage URL.
<code>merchant.merchant_id</code>	string	no	expanded	normal	Merchant identifier (company / VAT registration number, etc.).
<code>merchant.vat_id</code>	string	no	expanded	normal	Seller VAT / tax registration number (the seller side of a B2B transaction).
<code>merchant.vat_issuing_authority</code>	string	no	expanded	normal	Authority / country that issued the seller's VAT ID (e.g. "IL", "DE").

Path	Type	Req.	Profile	Sens.	Description
merchant.country	string	no	expanded	normal	Seller's country; defaults to merchant.address.country when present.
merchant.compliance	array<string>	no	expanded	normal	Data-protection / security standards the merchant attests to (e.g. ["soc2", "iso_27001", "gdpr"]). Useful for (government) procurement.
merchant.global_merchant_id	string	no	expanded	normal	Reserved / placeholder for a future universal merchant identifier. Populate only against an agreed scheme.
merchant.address	object	no	expanded	normal	Postal address.
merchant.address.street	string	no	expanded	normal	Street address.
merchant.address.city	string	no	expanded	normal	City.
merchant.address.region	string	no	expanded	normal	State / province / region.
merchant.address.postal_code	string	no	expanded	normal	Postal / ZIP code.
merchant.address.country	string	no	expanded	normal	Country code.
merchant.contact	object	no	expanded	personal	Merchant contact details. Treat as personal data when a named person.
merchant.contact.email	string	no	expanded	personal	Contact email.
merchant.contact.phone	string	no	expanded	personal	Contact phone.
buyer	object	no	expanded	personal	The purchasing party (chiefly for B2B). Supplies the buyer side of a PO / RFQ / reverse-charge VAT document.
buyer.business_name	string	no	expanded	personal	Legal name of the purchasing organization.
buyer.vat_id	string	no	expanded	personal	Buyer's VAT / tax registration number.
buyer.vat_issuing_authority	string	no	expanded	personal	Authority / country that issued the buyer's VAT ID.
buyer.country	string	no	expanded	personal	Buyer's country.
buyer.contact	object	no	expanded	personal	Optional name / email / phone for the purchaser.
buyer.contact.name	string	no	expanded	personal	Purchaser name.
buyer.contact.email	string	no	expanded	personal	Purchaser email.
buyer.contact.phone	string	no	expanded	personal	Purchaser phone.
currency	string	yes	core	normal	Currency for all amounts in the file.
multi_currency	boolean	no	expanded	normal	false by default. When true, individual items may carry their own currency and exchange-rate fields; totals stay in the top-level currency.
tax	object	no	expanded	normal	Cart-level tax context.
tax.scheme	string	no	expanded	normal	Tax scheme in effect. Values: vat, sales_tax, none.
tax.default_rate	number	no	expanded	normal	Default rate (decimal fraction, 0.18 = 18%) applied to items unless overridden per item.
tax.prices_include_tax	boolean	no	expanded	normal	Whether the merchant's displayed prices include tax.
items	array<object>	yes	core	normal	Line items.
items[].sku	string	conditional	expanded	normal	Merchant SKU / catalog number. Required if the merchant exposes a SKU.
items[].name	string	yes	core	normal	Short product name.
items[].description	string	no	expanded	normal	Longer description.

Path	Type	Req.	Profile	Sens.	Description
items[].variant	string	no	expanded	normal	Human-readable variant label (e.g. "Blue / XL").
items[].options	array<object>	no	expanded	normal	Structured variant selections.
items[].options[].name	string	no	expanded	normal	Option name (e.g. "Color").
items[].options[].value	string	no	expanded	normal	Option value (e.g. "Blue").
items[].quantity	number	yes	core	normal	Quantity in the cart.
items[].unit	string	no	expanded	normal	Unit of measure ("each", "box", "m", "kg"...). Defaults to "each".
items[].unit_price_excl_tax	number	yes	expanded	normal	Price per unit before VAT / tax.
items[].unit_price_incl_tax	number	yes	core	normal	Price per unit after VAT / tax.
items[].rrp_excl_tax	number	no	expanded	normal	Regular / reference price (RRP) per unit before tax, when discounted from a higher list price.
items[].rrp_incl_tax	number	no	expanded	normal	Regular / reference price (RRP) per unit after tax. Discount percentage is measured against this.
items[].currency	string	no	expanded	normal	Item currency. Only when cart multi_currency is true and it differs from the cart currency.
items[].exchange_rate	number	no	expanded	normal	Rate used to convert this item's currency to the cart currency.
items[].exchange_rate_calculated	boolean	no	expanded	normal	true if the rate was computed by the producing tool rather than quoted by the merchant.
items[].tax_rate	number	no	expanded	normal	Item-level tax rate if it differs from tax.default_rate.
items[].line_total_excl_tax	number	yes	expanded	normal	quantity x unit_price_excl_tax (after line discounts).
items[].line_total_incl_tax	number	yes	core	normal	Line total after VAT / tax.
items[].discount	object	no	expanded	normal	Discount applied to this line (already reflected in line totals).
items[].discount.description	string	no	expanded	normal	Human-readable label ("Summer sale", "Contract price").
items[].discount.amount	number	no	expanded	normal	Absolute discount per line, in cart currency.
items[].discount.percentage	number	no	expanded	normal	Discount as a decimal fraction of the RRP (0.15 = 15%).
items[].discount.type	string	no	expanded	normal	Distinguishes a manually applied discount from an automatic one. Values: automatic, manual, coupon, negotiated, promotion.
items[].product_url	string	no	expanded	normal	Canonical product page URL.
items[].image_url	string	no	expanded	normal	Product image URL.
items[].manufacturer	string	no	expanded	normal	Brand / manufacturer name.
items[].mpn	string	no	expanded	normal	Manufacturer part number (distinct from merchant SKU).
items[].gtin	string	no	expanded	normal	Barcode (EAN / UPC), when known. Key for cross-vendor comparison.
items[].availability	string	no	expanded	normal	Stock availability. Values: in_stock, backorder, preorder, unknown.
items[].min_order_quantity	number	no	expanded	normal	Minimum orderable quantity (MOQ), common in B2B.
items[].lead_time_days	number	no	expanded	normal	Quoted lead time in days for backordered / made-to-order items.

Path	Type	Req.	Profile	Sens.	Description
items[].price_breaks	array<object>	no	expanded	normal	Quantity-tiered pricing offered for this item. Essential for B2B comparison and RFQs.
items[].price_breaks[].min_quantity	number	yes	expanded	normal	Quantity at which this tier's price applies.
items[].price_breaks[].unit_price_excl_tax	number	no	expanded	normal	Per-unit price at this tier, before tax.
items[].price_breaks[].unit_price_incl_tax	number	no	expanded	normal	Per-unit price at this tier, after tax.
items[].notes	string	no	expanded	normal	Free-text line notes.
shipping	object	no	expanded	normal	Shipping methods and destination.
shipping.ship_to	object	no	expanded	personal	Destination address (merchant.address shape plus name and address_type).
shipping.ship_to.name	string	no	expanded	personal	Recipient name.
shipping.ship_to.address_type	string	no	expanded	normal	Affects carrier options and, sometimes, surcharges. Values: residential, business, warehouse, other.
shipping.ship_to.street	string	no	expanded	personal	Street address.
shipping.ship_to.city	string	no	expanded	personal	City.
shipping.ship_to.region	string	no	expanded	personal	State / province / region.
shipping.ship_to.postal_code	string	no	expanded	personal	Postal / ZIP code.
shipping.ship_to.country	string	no	expanded	personal	Country code.
shipping.incoterms	string	no	expanded	normal	Delivery terms for (international) B2B (ICC Incoterms 2020). Values: EXW, FCA, CPT, CIP, DAP, DPU, DDP, FAS, FOB, CFR, CIF.
shipping.selected_method_id	string	no	expanded	normal	id of the chosen method from available_methods.
shipping.available_methods	array<object>	no	expanded	normal	All methods the merchant offered at capture time.
shipping.available_methods[].id	string	no	expanded	normal	Stable identifier within this file (e.g. "courier-48h").
shipping.available_methods[].name	string	no	expanded	normal	Display name ("Courier, 2 business days").
shipping.available_methods[].cost_excl_tax	number	no	expanded	normal	Shipping cost before VAT / tax.
shipping.available_methods[].cost_incl_tax	number	no	expanded	normal	Shipping cost after VAT / tax.
shipping.available_methods[].estimated_delivery	string	no	expanded	normal	Free-text or ISO 8601 estimate.
totals	object	yes	core	normal	Cart-level monetary totals (all in currency).
totals.items_subtotal_excl_tax	number	yes	expanded	normal	Sum of line totals before tax, before shipping.
totals.items_subtotal_incl_tax	number	yes	expanded	normal	Sum of line totals after tax, before shipping.
totals.shipping_excl_tax	number	no	expanded	normal	Cost of the selected shipping method, before tax.
totals.shipping_incl_tax	number	no	expanded	normal	Cost of the selected shipping method, after tax.
totals.discount_total	number	no	expanded	normal	Cart-level discounts (already reflected in the grand total).
totals.tax_total	number	yes	expanded	normal	Total VAT / tax across items and shipping.
totals.grand_total	number	yes	core	normal	Final amount: items + shipping - discounts, tax included.
totals.amount_pending_payment	number	yes	core	normal	Amount currently awaiting payment (usually equals grand_total; may differ with deposits / partial payments).

Path	Type	Req.	Profile	Sens.	Description
payment	object	no	expanded	normal	Payment status / expectations.
payment.status	string	no	expanded	normal	Lifecycle status of the cart / order. Values: cart, quote_requested, po_issued, awaiting_payment, partially_paid, paid.
payment.method	string	no	expanded	normal	The payment method the buyer has selected or intends to use (a single value drawn from methods_accepted).
payment.terms	string	no	expanded	normal	Payment terms if known (e.g. "net_30", "net_60", "prepaid", "cod").
payment.methods_accepted	array<string>	no	expanded	normal	Methods the merchant offers (e.g. ["credit_card", "bank_transfer", "purchase_order"]).
payment.reference	string	no	expanded	normal	Quote / PO / invoice reference once one exists.
payment.card	object	no	expanded	sensitive	The payment card the buyer intends to use. Never store full PAN, CVV/CVC, PIN, or full expiry.
payment.card.cardholder_name	string	no	expanded	sensitive	Name printed on the card.
payment.card.is_business_card	boolean	no	expanded	normal	true if a business / corporate card, false for a personal card.
payment.card.brand	string	no	expanded	normal	Card network if known (e.g. "visa", "mastercard", "amex").
payment.card.last4	string	no	expanded	sensitive	Last four digits, when a masked reference is needed to disambiguate which card.
notes	string	no	expanded	normal	Free-text notes (human or agent authored).
extensions	object	no	expanded	normal	Namespaced vendor / tool-specific data. Consumers must preserve content they don't understand when re-writing a file.