LynkaHub API Documentation – Endpoints

Version 1.07 · 4 November 2026 · Environment: Test

> Note: This is the current, up-to-date version of this document as of the date it was sent to you. See Change history for what changed since the previous version.

Contents

1. General information
2. Endpoint – Health
3. Endpoint – Submit order
4. Endpoint – Order status
5. Endpoint – Availability
6. Change history

1. General information

Environment and base URL

Environment	Base URL
Test	`https://apilynka-test.lynka.eu`
Production	The URL will be provided before go-live

Before you start

- Send us the list of **public IP addresses** your system will call us from. Access to the API is restricted to allowlisted addresses – requests coming from any other address are rejected before they reach the API. Let us know in advance whenever those addresses change.
- We will provide you with your `X-Client-Code` and `X-Api-Key`. The API key is shown only once at the moment it is created and cannot be recovered later. Store it securely; if it is lost we issue a new one.

Authentication

Every endpoint except the health check requires both headers below, in addition to the IP allowlist described above.

Header	Description
`X-Client-Code`	Your client code, assigned by Lynka
`X-Api-Key`	Your API key, assigned by Lynka

Whether the headers are missing, the client code is unknown, the account is inactive or the key is wrong, the API always answers with the same `401` response. The exact reason is deliberately not disclosed.

Request and response conventions

- Request and response bodies are JSON, UTF-8 encoded. Send `Content-Type: application/json` with every POST request.
- Timestamps follow ISO-8601.
- Decimal values – for example quantities in the availability response – are returned as JSON **strings** with four decimal places, e.g. `"12.5000"`. Parse them as decimals, not as floating point numbers.

Common error responses

Code	When it happens	Response body
`401 Unauthorized`	Missing, unknown or invalid credentials, or an inactive account.	`{ "detail": "Invalid client credentials" }`
`404 Not Found`	Order status lookup for a `correlationId` that does not exist, or that belongs to another client. Both cases give the same answer.	`{ "detail": "Order not found" }`
`422 Unprocessable Entity`	A query or path parameter is missing, has the wrong type, or is out of range.	See the example below.
`500 Internal Server Error`	Unexpected error on our side. The request can be safely retried.	`{ "detail": "Internal server error" }`

Example of a `422` response:

```
{
  "detail": [
    {
      "type": "missing",
      "loc": ["query", "orderNo"],
      "msg": "Field required",
      "input": null
    }
  ]
}
```

Order lifecycle

Orders are accepted asynchronously. The submission call and the outcome of processing are two separate steps:

1. Call `POST /api/v1/orders`. A `202` response confirms the order was queued.
2. Store the returned `correlationId`. It identifies this submission in every later lookup and in any support request.
3. Validation and delivery to the target system happen afterwards, in the background.
4. Poll `GET /api/v1/orders/{correlationId}` for the outcome.

Because validation is asynchronous, a `202` does not mean the order is correct. An order with invalid or missing fields is also answered with `202` and is reported as `REJECTED`, together with a list of field errors, by the order status endpoint (section 4).

2. Endpoint – Health

`GET /api/v1/health`

Description: Health check endpoint used to verify if the API is reachable.

Authentication: No API key required – this is the only endpoint that does not need `X-Client-Code` and `X-Api-Key`. The IP allowlist still applies.

Example request:

```
curl -X 'GET' \
  'https://apilynka-test.lynka.eu/api/v1/health' \
  -H 'accept: application/json'
```

Example response: `200`

```
{
  "status": "OK",
  "service": "api"
}
```

Errors:

- `500 Internal Server Error`

```
{
  "detail": "Internal server error"
}
```

3. Endpoint – Submit order

`POST /api/v1/orders`

Description: Submits a new order for processing. The order is accepted asynchronously: a `202` response means the order was received into the queue, not that it has already been validated or delivered to the target system.

Authentication: `X-Client-Code` and `X-Api-Key` headers plus the IP allowlist – see Authentication.

Request body

Field	Required	Rules	Description
`orderNo`	Yes	1–50 characters	Unique order number from the client's system
`shipToName`	Yes	1–100 characters	Recipient name
`shipToAddress`	Yes	1–200 characters	Shipping address
`shipToPostCode`	Yes	1–20 characters; format `NN-NNN` is required when `shipToCountryCode` is `PL`	Shipping postal code
`shipToCity`	Yes	1–100 characters	Shipping city
`shipToCountryCode`	Yes	Exactly 2 uppercase letters	2-letter ISO country code, e.g. `PL`
`shipToPhoneNo`	Yes	Leading `+`, then a digit 1–9, then 6–14 more digits (7–15 digits total)	Recipient phone number in international format, e.g. `+48123456789`
`shipToEmail`	Yes	Valid email address	Recipient email address
`remarks`	No	Max 250 characters	Optional free-text remarks
`requestedShipmentDate`	No	ISO-8601 date (`YYYY-MM-DD`)	Shipment date requested for this order
`orderLines`	Yes	At least one line	Array of order lines – see fields below

`orderLines[]` fields

Field	Required	Rules	Description
`lineNo`	Yes	Integer greater than 0, unique within the order	Line number
`itemNo`	Yes	1–50 characters	Lynka SKU
`quantity`	Yes	Integer greater than 0	Quantity ordered
`orderType`	Yes	One of: `Standard` , `POD` , `Stock`	Type of the order line
`designCode`	Conditional	Max 50 characters	Required when `orderType` is `Standard` or `POD` ; not required for `Stock`
`decorationLines`	Conditional	Array; at least one entry required when `orderType` is `Standard` ; optional for `POD` and `Stock`	Decoration data for the line – see `decorationLines[]` fields below

`decorationLines[]` fields

Required (at least one entry) when the order line's `orderType` is `Standard`; optional for `POD` and `Stock`.

Field	Required	Rules	Description
`externalDesignCode`	Yes	1–50 characters	Identifier of the design/artwork
`decorationType`	Yes	1–50 characters	Type of decoration, e.g. `Embroidery`, `ScreenPrint`, `DigitalPrint`, `Transfer`, `Sewing`. Not validated by the API – an unsupported value is only reported later as a decoration error (see section 4).
`placement`	No	Max 20 characters	Location on the item where the decoration is applied. Allowed values come from a fixed, closed list maintained by Lynka; the list itself will be published in a future update of this document.
`numberOfColors`	No	Integer, 0 or greater	Number of colors used in the decoration
`numberOfStitches`	No	Integer, 0 or greater	Number of stitches (embroidery)
`width`	No	Decimal number	Width of the decoration in centimeters (cm)
`height`	No	Decimal number	Height of the decoration in centimeters (cm)
`personalization`	No	`true` / `false`, default `false`	Whether this is a personalized (per-recipient) decoration
`decorationComment`	No	Max 250 characters	Free-text comment
`decorationUrl1` / `decorationUrl2`	No	Max 250 characters each	URLs to design files or previews
`colors`	No	Max 250 characters	Free-text list of colors
`decorationCode`	No	Max 50 characters	Optional internal decoration code

Example request:

```
curl -X 'POST' \
  'https://apilynka-test.lynka.eu/api/v1/orders' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-Client-Code: CLIENT_TEST' \
  -H 'X-Api-Key: XXX' \
  -d '{
  "orderNo": "ORD-2026-00123",
  "shipToName": "John Smith",
  "shipToAddress": "123 Test Street",
  "shipToPostCode": "00-001",
  "shipToCity": "Warsaw",
  "shipToCountryCode": "PL",
  "shipToPhoneNo": "+48123456789",
  "shipToEmail": "john.smith@example.com",
  "remarks": "Please contact before delivery",
  "requestedShipmentDate": "2026-10-01",
  "orderLines": [
    {
      "lineNo": 10000,
      "itemNo": "TBC/TU01T/001/3",
      "quantity": 5,
      "orderType": "Standard",
      "designCode": "NZ02033-26-P00062",
      "decorationLines": [
        {
          "externalDesignCode": "NZ02033-26-P00062",
          "decorationType": "Embroidery",
          "placement": "FRONT",
          "numberOfColors": 0,
          "numberOfStitches": 0,
          "width": 8,
          "height": 6,
          "personalization": false,
          "decorationComment": "Company logo",
          "colors": "PANTONE 186C, White, Black"
        }
      ]
    }
  ]
}'
```

Example response: `202`

```
{
  "id": 110,
  "correlationId": "f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0",
  "status": "NEW",
  "clientCode": "CLIENT_TEST",
  "orderNo": "ORD-2026-00123"
}
```

Processing

- A `202` response only confirms that the order was accepted into the queue. Validation and delivery to the target system happen asynchronously afterwards and are not reflected in this response.
- Store the `correlationId` – it identifies this submission for any future status lookup (section 4) and is the reference we need in any support request.
- The payload is not validated at this point. An order with invalid or missing fields also receives a `202` and is reported as `REJECTED`, with a list of field errors, by the order status endpoint.

Re-submitting the same order number

Every call creates a new submission with its own `correlationId` – a repeated call is never silently ignored. If an order with the same `orderNo` has already been processed, the new submission ends with status `DUPLICATE` and `duplicateOfCorrelationId` pointing at the first one. `GET /api/v1/orders?orderNo=...` returns the full history of attempts made under one order number.

Errors:

- `401 Unauthorized`

```
{
  "detail": "Invalid client credentials"
}
```

- `500 Internal Server Error`

```
{
  "detail": "Internal server error"
}
```

4. Endpoint – Order status

Two endpoints returning the same object. Use the `correlationId` variant when you have the identifier from the `202` response; use the `orderNo` variant to see every submission made under one order number.

Status of a single submission

`GET /api/v1/orders/{correlationId}`

Description: Returns the status of the submission identified by `correlationId` – the value returned in the `202` response when the order was accepted.

Authentication: `X-Client-Code` and `X-Api-Key` headers plus the IP allowlist – see Authentication.

Path parameter:

Parameter	Required	Description
`correlationId`	Yes	UUID returned in the `202` response of `POST /api/v1/orders`

Example request:

```
curl -X 'GET' \
  'https://apilynka-test.lynka.eu/api/v1/orders/f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0' \
  -H 'accept: application/json' \
  -H 'X-Client-Code: CLIENT_TEST' \
  -H 'X-Api-Key: XXX'
```

Example response: `200`

```
{
  "correlationId": "f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0",
  "orderNo": "ORD-2026-00123",
  "status": "ACCEPTED",
  "reason": null,
  "message": null,
  "receivedAt": "2026-08-27T09:12:44.512000Z",
  "updatedAt": "2026-08-27T09:13:02.870000Z",
  "validationErrors": null,
  "duplicateOfCorrelationId": null,
  "decorationErrors": null,
  "trackingNumbers": null
}
```

Errors:

- `404 Not Found` – unknown `correlationId`, or one that belongs to another client. Both cases return the same response.

```
{
  "detail": "Order not found"
}
```

- `422 Unprocessable Entity` – the value in the path is not a valid UUID.
- `401 Unauthorized` – see section 1.

Statuses of every submission under an order number

`GET /api/v1/orders`

Description: Returns every submission accepted under the given order number, newest first. Re-sending the same number creates another entry, so the list shows the full history of attempts.

Authentication: `X-Client-Code` and `X-Api-Key` headers plus the IP allowlist – see Authentication.

Query parameters:

Parameter	Required	Rules	Description
`orderNo`	Yes	1–50 characters	Order number from your system, as sent in the request body of `POST /api/v1/orders`

Example request:

```
curl -X 'GET' \
  'https://apilynka-test.lynka.eu/api/v1/orders?orderNo=ORD-2026-00123' \
  -H 'accept: application/json' \
  -H 'X-Client-Code: CLIENT_TEST' \
  -H 'X-Api-Key: XXX'
```

Example response: `200`


```
{
  "total": 2,
  "items": [
    {
      "correlationId": "3c1b90a4-52e7-4a1f-9c02-8b6e0a77d311",
      "orderNo": "ORD-2026-00123",
      "status": "DUPLICATE",
      "reason": "DUPLICATE_ORDER",
      "message": "Order with this number already exists - see duplicateOfCorrelationId.",
      "receivedAt": "2026-08-27T11:40:18.004000Z",
      "updatedAt": "2026-08-27T11:40:23.119000Z",
      "validationErrors": null,
      "duplicateOfCorrelationId": "f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0",
      "decorationErrors": null,
      "trackingNumbers": null
    },
    {
      "correlationId": "f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0",
      "orderNo": "ORD-2026-00123",
      "status": "ACCEPTED",
      "reason": null,
      "message": null,
      "receivedAt": "2026-08-27T09:12:44.512000Z",
      "updatedAt": "2026-08-27T09:13:02.870000Z",
      "validationErrors": null,
      "duplicateOfCorrelationId": null,
      "decorationErrors": null,
      "trackingNumbers": null
    }
  ]
}
```


If you have not submitted anything under this number, the response is `200` with `total` set to `0` and an empty `items` array – not a `404`. An order number belonging to another client gives the same empty result.

Errors:

- `422 Unprocessable Entity` – `orderNo` is missing or empty.
- `401 Unauthorized` – see section 1.

Response fields

Field	Type	Description
`correlationId`	string (UUID)	Identifier of this submission
`orderNo`	string or null	Order number as received from you

Field	Type	Description
`status`	string	See Statuses below
`reason`	string or null	Machine-readable cause. Present only when something other than normal processing happened – see Reason codes below.
`message`	string or null	Human-readable explanation. Present together with `reason`.
`receivedAt`	string (ISO-8601)	When we received the submission
`updatedAt`	string (ISO-8601)	When the state of this submission last changed
`validationErrors`	array or null	Present only when `reason` is `VALIDATION_ERROR`. Each entry has `field` and `message`.
`duplicateOfCorrelationId`	string (UUID) or null	Present only when `status` is `DUPLICATE`. Points at the first submission made under the same order number.
`decorationErrors`	array or null	Present only when at least one decoration line was rejected by the backend. Independent of `status` / `reason` above – see note below. Each entry has `lineNo`, `externalDesignCode` and `message`.
`trackingNumbers`	string or null	Carrier tracking number(s) reported by the target system. Independent of `status` – see note below.

Statuses

Status	Meaning	What to do
`PENDING`	Accepted and still being processed.	Nothing. Poll again later.
`ACCEPTED`	Delivered to the target system.	Nothing.
`SHIPPED`	The order has been shipped.	Nothing. Tracking number(s), when available, are in `trackingNumbers`.
`REJECTED`	The payload failed validation. No order was created.	Correct the fields listed in `validationErrors` and submit the same `orderNo` again.
`DUPLICATE`	An order with this number already exists.	Check `duplicateOfCorrelationId`. If this was meant to be a new order, submit it under a different `orderNo`.
`FAILED`	Processing stopped on our side.	Contact Lynka support and quote the `correlationId`.

No status is final. `ACCEPTED` means the order reached the target system, not that it has been produced or shipped – shipment is reported separately as `SHIPPED`. `FAILED` can go back to `PENDING` if we retry the order manually. Do not treat any value as a terminal state.

`trackingNumbers` is filled in as soon as the target system reports a tracking number, independent of `status` – it can appear before the order reaches `SHIPPED`. Once present, it is never removed. The field is a single text value that may contain more than one tracking number.

`decorationErrors` is independent of `status` / `reason` above. An order can be `ACCEPTED` while one of its decoration lines was rejected by the backend – `status` describes the order itself, `decorationErrors` describes its decorations. Fix a rejected line with `PUT /api/v1/orders/{correlationId}/lines/{lineNo}/decorations/{externalDesignCode}` – see Fix a rejected decoration line.

Reason codes

reason	Returned with	Meaning
`VALIDATION_ERROR`	`REJECTED`	The payload did not pass validation. Details in `validationErrors`.
`DUPLICATE_ORDER`	`DUPLICATE`	An order with the same order number already exists.
`CLIENT_CONFIG_ERROR`	`FAILED`	Configuration issue on the Lynka side. Nothing that can be fixed in the payload.
`BC_ERROR`	`FAILED`	The target system rejected the order for a business reason. This is the only reason code where `message` carries the actual reason returned by that system instead of a fixed text.
`BC_CONNECTION_ERROR`	`FAILED`	The order could not be delivered to the target system.
`INTERNAL_ERROR`	`FAILED`	Unexpected processing error.

Example – shipped order

`200`


```
{
  "correlationId": "f9d72f25-7e0d-4ddd-9256-4c7a3ad5dde0",
  "orderNo": "ORD-2026-00123",
  "status": "SHIPPED",
  "reason": null,
  "message": null,
  "receivedAt": "2026-08-27T09:12:44.512000Z",
  "updatedAt": "2026-08-29T14:20:11.305000Z",
  "validationErrors": null,
  "duplicateOfCorrelationId": null,
  "decorationErrors": null,
  "trackingNumbers": "00340434161094042557"
}
```

Example – rejected order

`200`


```
{
  "correlationId": "8a51f0c7-2d33-4a0e-9f10-1c7b5f2ab904",
  "orderNo": "ORD-2026-00124",
  "status": "REJECTED",
  "reason": "VALIDATION_ERROR",
  "message": "Order payload failed validation - see validationErrors.",
  "receivedAt": "2026-08-27T12:05:11.220000Z",
  "updatedAt": "2026-08-27T12:05:16.480000Z",
  "validationErrors": [
    {
      "field": "orderLines.0.quantity",
      "message": "Input should be greater than 0"
    },
    {
      "field": "shipToCountryCode",
      "message": "String should match pattern '^[A-Z]{2}$'"
    }
  ],
  "duplicateOfCorrelationId": null,
  "decorationErrors": null,
  "trackingNumbers": null
}
```

Example – accepted order with a rejected decoration line

`200`


```
{
  "correlationId": "f1a2b3c4-1111-2222-3333-444455556666",
  "orderNo": "ORD-2026-00130",
  "status": "ACCEPTED",
  "reason": null,
  "message": null,
  "receivedAt": "2026-09-20T09:00:00.000000Z",
  "updatedAt": "2026-09-20T09:05:00.000000Z",
  "validationErrors": null,
  "duplicateOfCorrelationId": null,
  "decorationErrors": [
    {
      "lineNo": 10000,
      "externalDesignCode": "NZ02033-26-P00062",
      "message": "'Print' is not an option. The existing options are:
Embroidery,ScreenPrint,DigitalPrint,Transfer,Sewing"
    }
  ],
  "trackingNumbers": null
}
```


The order itself was delivered to the backend (status `ACCEPTED`), but the decoration on line `10000` was rejected because of an unsupported `decorationType` value. Fix it with the endpoint below – there is no need to resubmit the whole order.

Polling

Poll no more often than **once every 15 minutes per order**. Statuses change on the order of minutes, not seconds. There is no push notification or callback at the moment.

Fix a rejected decoration line

`PUT /api/v1/orders/{correlationId}/lines/{lineNo}/decorations/{externalDesignCode}`

Description: Replaces a decoration line that was rejected by the backend (see `decorationErrors` above) with corrected data, and queues it for another delivery attempt. The request body must be the full, corrected decoration line – the same shape as `decorationLines[]` items in section 3. `externalDesignCode` cannot be changed this way; it must match the value in the URL.

Authentication: `X-Client-Code` and `X-Api-Key` headers plus the IP allowlist – see Authentication.

Path parameters:

Parameter	Required	Description
`correlationId`	Yes	UUID of the order, from the `202` response of `POST /api/v1/orders`
`lineNo`	Yes	Line number of the order line that carries the rejected decoration (see `decorationErrors`)

Parameter	Required	Description
`externalDesignCode`	Yes	Design code of the rejected decoration line (see `decorationErrors`); must match the same field in the request body

Request body: Same fields as `decorationLines[]` items in section 3. Only `externalDesignCode` and `decorationType` are required (as in that table) – everything else is optional.

> **Important:** this is a full replacement, not a partial update. Any field left out of the body is cleared to its default (`null` , or `false` for `personalization`) – even if it had a value before. Resend the complete, corrected decoration line, not just the field you changed.

Example request:

```
curl -X 'PUT' \
  'https://apilynka-test.lynka.eu/api/v1/orders/8a51f0c7-2d33-4a0e-9f10-
1c7b5f2ab904/lines/10000/decorations/NZ02033-26-P00062' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'X-Client-Code: CLIENT_TEST' \
  -H 'X-Api-Key: XXX' \
  -d '{
  "externalDesignCode": "NZ02033-26-P00062",
  "decorationType": "Embroidery",
  "placement": "FRONT",
  "numberOfColors": 3,
  "numberOfStitches": 4500,
  "width": 8,
  "height": 6,
  "personalization": false,
  "decorationUrl1": "https://example.com/designs/NZ02033-26-P00062.png",
  "decorationUrl2": "https://example.com/files/design-preview.pdf",
  "decorationComment": "Corrected decoration type",
  "colors": "PANTONE 186C, White, Black"
}'
```

Example response: `200`

```
{
  "status": "NEW",
  "externalDesignCode": "NZ02033-26-P00062"
}
```

Errors:

- `404 Not Found` – the order, or this decoration line, does not exist (or does not belong to you).

```
{
  "detail": "Decoration line not found"
}
```

- `409 Conflict` – the decoration line is not currently in a rejected state.
- `400 Bad Request` – `externalDesignCode` in the body does not match the value in the URL.
- `401 Unauthorized` – see section 1.

5. Endpoint – Availability

`GET /api/v1/availability`

Description: Returns product stock availability for the authenticated client. The visible catalogue can be limited per client, so `total` is the number of records within your scope, not the size of the whole Lynka catalogue.

Authentication: `X-Client-Code` and `X-Api-Key` headers plus the IP allowlist – see Authentication.

Query parameters:

Parameter	Default	Rules	Description
`limit`	`1000`	Integer, 1–5000	Number of records to return
`offset`	`0`	Integer, 0 or greater	Number of records to skip
`onlyAvailable`	`false`	`true` / `false`	When `true`, returns only records where `availabilityQtyLynka` or `availabilityQtyVend` is greater than 0
`lynkaSKU`	–	Exact match	Optional filter to return availability for a specific Lynka SKU
`modifiedAfter`	–	ISO-8601 timestamp	Optional filter – returns only items whose `lastModifiedDateTime` is strictly later than the given value

Example request:

```
curl -X 'GET' \
  'https://apilynka-test.lynka.eu/api/v1/availability?limit=3&offset=0&onlyAvailable=false' \
  -H 'accept: application/json' \
  -H 'X-Client-Code: CLIENT_TEST' \
  -H 'X-Api-Key: XXX'
```

Example response: `200`


```
{
  "total": 16269,
  "limit": 3,
  "offset": 0,
  "items": [
    {
      "lynkaSKU": "XXX/00512/01",
      "availabilityQtyLynka": "0.0000",
      "availabilityQtyVend": "0.0000",
      "lastModifiedDateTime": "2026-08-01T10:15:00Z"
    },
    {
      "lynkaSKU": "XXX/00512/02",
      "availabilityQtyLynka": "3.0000",
      "availabilityQtyVend": "0.0000",
      "lastModifiedDateTime": "2026-08-02T09:30:00Z"
    },
    {
      "lynkaSKU": "XXX/00512/03",
      "availabilityQtyLynka": "10.0000",
      "availabilityQtyVend": "100.0000",
      "lastModifiedDateTime": "2026-08-03T14:45:00Z"
    }
  ]
}
```


Response fields

Field	Type	Description
`total`	integer	Number of records matching the query within your scope
`limit`	integer	Value of `limit` used for this response
`offset`	integer	Value of `offset` used for this response
`items[].lynkaSKU`	string	Lynka SKU
`items[].availabilityQtyLynka`	string (decimal)	Quantity available in the Lynka warehouse
`items[].availabilityQtyVend`	string (decimal)	Quantity available at the vendor
`items[].lastModifiedDateTime`	string (ISO-8601) or null	When the record was last modified in the source system. Null when the source system did not provide a date.

Pagination

Results are always sorted by `lynkaSKU` ascending, so paging with `limit` and `offset` is stable.

Example:

```
limit=1000&offset=0
limit=1000&offset=1000
limit=1000&offset=2000
```

Keep paging until fewer than `limit` items come back, or until `offset` reaches `total`.

Data refresh

Availability is refreshed from the source system every 3 hours. Polling more often than that returns the same data. One full download every few hours, or an incremental call using `modifiedAfter`, is enough.

Optional SKU filter example

```
/api/v1/availability?lynkaSKU=XXX/00512/01
```

Errors:

- `401 Unauthorized`

```
{
  "detail": "Invalid client credentials"
}
```

- `422 Unprocessable Entity` – `limit` outside the range 1–5000, or `offset` below 0, or an unparsable `modifiedAfter` value.
- `500 Internal Server Error`

```
{
  "detail": "Internal server error"
}
```

6. Change history

Newest changes first. Each entry links to the section it affects. **Action** marks changes that may require an update on your side.

Version	Date	Type	Section	Change
1.07	4 November 2026	Changed, **Action**	`decorationLines[]` fields	Clarified that `width` and `height` are in centimeters (cm); examples updated accordingly. If you have been sending these values in another unit, convert them to centimeters.
1.06	29 October 2026	Added	Statuses	New order status `SHIPPED`.
1.06	29 October 2026	Added, **Action**	Response fields	New field `trackingNumbers` in every order status response (`null` until a tracking number is known).