

Starmile Partner API

Migration guide: v1 → v2

Revision 1.0 · 21 September 2026 · Online reference: <https://docs.starmile.io/v2>

1. What this is

v1 and v2 are separate, parallel contracts. v1 keeps serving exactly as it does today and is not being switched off as part of this migration. v2 is where new vocabulary — and new endpoints — land. Pick *one* version per integration and move deliberately; never mix versions inside one workflow.

Three things change, and nothing else:

1. The **box vocabulary**. A box is an *item*, addressed only by your own references.
2. The **status strings**. v1 is frozen on the names it was handshaked with; v2 carries Starmile's current catalogue.
3. Two **new endpoints** that exist on v2 only: the delivery code and the proof of delivery.

2. What does not change

Your credential, the token endpoint, the base URL, the error shapes, the idempotency rules and the polling model are all identical. Nothing about authentication needs to be touched.

Base URL	<code>https://api.starmile.io</code>
Token	<code>POST /oauth/token · grant client_credentials</code>
Auth header	<code>Authorization: Bearer <access_token></code>
Credential	The same one you use on v1. It pins your partner account; no new credential is issued for v2.

```
curl -X POST https://api.starmile.io/oauth/token \  
-d grant_type=client_credentials \  
-d client_id=<your client id> \  
-d client_secret=<your client secret>
```

3. Endpoint map

v1	v2	Difference
<code>GET /api/v1/services</code>	<code>GET /api/v2/services</code>	None — only the pa
<code>GET /api/v1/rates</code>	<code>GET /api/v2/rates</code>	None.
<code>POST /api/v1/orders</code>	<code>POST /api/v2/orders</code>	parcels[] is items[] names the referenc See §4.
<code>GET /api/v1/orders/label</code>	<code>GET /api/v2/orders/label</code>	Same PDF, differ parameters. See § one place a v1 hab silently.
<code>POST /api/v1/orders/{order}/parcels</code>	<code>POST /api/v2/orders/{order_id}/items</code>	Same body shape a

		items[].
PATCH .../parcels/{parcel}	PATCH .../items/{item}	The item is address reference.
POST .../parcels/{parcel}/cancel	POST .../items/{item}/cancel	As above.
POST .../split	POST .../split	Body unchanged: i new_order_id.
POST .../cancel	POST .../cancel	None.
GET /api/v1/partner/changes	GET /api/v2/partner/changes	New cursor space a strings. See §5.
POST /api/v1/partner/events	POST /api/v2/partner/events	None.
—	GET /api/v2/orders/detail	New. See §7.
—	GET /api/v2/orders/delivery-code	New. See §8.
—	GET /api/v2/orders/pod	New. See §9.

4. Vocabulary: which reference means what

On v1, order_id is overloaded — it is your reference in the request body and Starmile's number in the response. v2 separates them once and for all:

v2 field	Whose reference it is
tracking_number	Starmile's. The order's only tracking reference, returned on create.
order_id	Yours. The order reference you sent on create, echoed back everywhere.
item_id	Yours. Your per-box reference.
merchant_tracking	Yours. The physical sticker code on the box.

An item has no Starmile tracking number of its own. v1's parcels[].parcel_id does not exist on v2 and is not replaced: you address a box by item_id or merchant_tracking, and the order's tracking_number is the only Starmile reference in play.

v1	v2
parcels[]	items[]
parcels[].parcel_id (Starmile's per-box number)	— removed, nothing replaces it
create response: order_id = Starmile's number	tracking_number = Starmile's number, order_id = yours
status rows: external_parent_id	status rows: order_id
status rows: external_id	status rows: item_id

Everything else about creating an order is unchanged: the same required fields, the same service_id resolution, the same region handling, the same idempotency — a replayed order_id answers 200 with duplicate: true.

5. Status strings are not the same

Since v1 was handshaked, Starmile's internal status catalogue was renamed, and several statuses were split so the journey names *which* hub or *which* customs leg a package reached. v1 is deliberately frozen: those statuses are projected back to the string v1 always sent.

v2 does no such projection. It carries the current name, so a switch written against the left column stops matching the day you move. Where one v1 string split into several, v2 tells you which one actually happened.

v1 sends	v2 sends instead
waiting_for_arrival	awaiting_arrival
accepted_at_hub	dws_processed
received_at_hub	received_at_origin_hub, received_at_central_hub, received_at_destination_hub
sorted	sorted_at_central_hub, sorted_at_destination_hub, sorted_at_regional_hub, sorted_at_transfer_hub
consolidation_racked	consolidation_stuffed
parcel_racked_at_hub	parcel_stuffed_at_hub
dispatched_from_hub	dispatched_from_origin_hub, dispatched_from_central_hub, dispatched_from_destination_hub, dispatched_from_regional_hub
loading_in_progress	received_by_tms
in_transit_to_hub	in_transit_to_destination_hub
arrived_at_terminal	arrived_at_import_terminal, arrived_at_export_terminal
customs_in_progress	import_customs_clearance_in_progress, export_customs_clearance_in_progress
customs_completed	import_customs_clearance_completed, export_customs_clearance_completed
customs_released	handed_over_to_tms
handover_to_carrier	handed_over_to_carrier
out_for_delivery	delivery_attempt
forwarded_to_destination	pudo_forward
accepted_at_pudo	parcel_stuffed_at_pudo
cancelled	order_cancelled

Paperwork statuses are the exception — they do not change.

customs_hold, customs_hold_resolved, declaration_pushed and declaration_confirmed describe paperwork rather than a movement, and cross the wire as the same strings on *both* versions. A customs hold is always customs_hold — never a direction-specific spelling. Which hold it is comes from reason, not from the status.

The status pool's cursor space is new.

The v2 pool's cursor comes from a different id space than v1's, so a cursor stored while polling v1 is meaningless on v2 — never carry it over. When you switch, drain v2 from since=0 (optionally per order via order_id), de-duplicate against what you already processed on v1, then continue with v2 cursors only.

6. Labels: the one trap

GET /api/v2/orders/label returns the same PDF as v1, but it addresses the order the v2 way. This is the only endpoint where the same call, unchanged, means something different on the two versions:

Parameter	v1 meaning	v2 meaning
order_id	Starmile's tracking number	Your order reference
tracking_number	—	Starmile's tracking number
merchant_tracking	One box	One box (unchanged)
parcel_id	One box, by Starmile's number	Does not exist

An order_id holding one of *our* tracking numbers answers 404 on v2. If your label calls silently start returning 404 after the switch, this is why.

7. New: order detail

GET /api/v2/orders/detail · scope orders:read

One of your orders in full. The status pool says what *happened* to an order; this says what the order *is*: its status, where it is going, the carton's measured dimensions and weights, and the items inside with the products you declared.

Order-level. After consolidation the order *is* the physical box: the items you announced are packed into one carton that carries the order's tracking_number, and every measurement, scan and handover from then on happens to that carton. So status and dimensions belong to the order, and items[] are its *contents* — each named by your own item_id and merchant_tracking, with no status and no dimensions of its own.

Query	
tracking_number	Starmile's order reference. Required unless order_id is given.
order_id	Your own order reference. Required unless tracking_number is given.

```
curl -G 'https://api.starmile.io/api/v2/orders/detail' \
-H 'Authorization: Bearer <access_token>' \
--data-urlencode 'tracking_number=CMX0000012345'

{
  "data": {
    "tracking_number": "CMX0000012345",
    "order_id": "PO-10294",
    "status": "sorted",
    "created_at": "2026-10-01T12:30:00+00:00",
    "cancelled_at": null,
    "cancel_reason": null,
    "origin_country": "CN",
    "destination_country": "AZ",
    "consolidation": { "required": true, "consolidated_at": "2026-10-03T09:10:00+00:00",
"package_sealed_at": "2026-10-03T09:12:00+00:00" },
    "delivery": { "type": "home", "address": null, "address_first": "Nizami küç. 10", "address_second":
"m?nzil 4", "zip": "AZ1000", "notes": null, "region": "N?simi", "parent_region": "Bak?", "pudo": null
    },
    "declared_value": "150.00",
    "currency": "USD",
    "dimensions": { "weight_grams": 2350, "volumetric_weight_grams": 4000, "chargeable_weight_grams":
```

```

4000, "length_mm": 400, "width_mm": 300, "height_mm": 200, "volume_cm3": 24000, "measured_at":
"2026-10-03T09:10:00+00:00" },
  "items": [
    { "item_id": "BOX-1", "merchant_tracking": "MT-0001", "package_type": "fragile", "products": [ {
"name": "Running shoes", "hs_code": "640411", "quantity": 1, "unit_price": "50.00", "currency": "USD",
"weight_grams": 900, "description": null } ] },
    { "item_id": "BOX-2", "merchant_tracking": "MT-0002", "package_type": null, "products": [ {
"name": "T-shirt", "hs_code": "610910", "quantity": 2, "unit_price": "12.50", "currency": "USD",
"weight_grams": 200, "description": "cotton" } ] }
  ]
}

```

dimensions	What it is
weight_grams	Actual (scale) weight of the carton, in grams.
volumetric_weight_grams	Dimensional weight, computed where the carton was measured. Null until then.
chargeable_weight_grams	The greater of the two — the weight declared to customs.
length_mm, width_mm, height_mm, volume_cm3	The carton's measurements. Null until measured.
measured_at	When the hub scale last measured the carton. Null means every number above is still what you declared.

The recipient's name, phone, email and government ID are deliberately **not** returned. An order that is not yours answers 404, never 403.

8. New: delivery code

GET /api/v2/orders/delivery-code · scope delivery_code:read

The code the recipient reads to the courier at the door; a wrong code refuses the handover. Starmile's own tracking page shows it to the recipient, but your customers may never visit that page — so fetch it here and show it in your own app.

Order-level. One order carries one code however many items it ships in, so there is no item parameter, and the code does *not* change after a failed delivery attempt.

Available as soon as the order exists. The code is minted when the order is created, not when the recipient is messaged — there is nothing to poll for.

Query	
tracking_number	Starmile's order reference. Required unless order_id is given.
order_id	Your own order reference. Required unless tracking_number is given.

```

curl -G 'https://api.starmile.io/api/v2/orders/delivery-code' \
-H 'Authorization: Bearer <access_token>' \
--data-urlencode 'order_id=P0-10294'

{

```

```

"data": {
  "tracking_number": "CMX0000012345",
  "order_id": "PO-10294",
  "delivery_code": "4821",
  "status": "active"
}
}

```

Read status before you display anything. A missing code is an ordinary answer, not an error — the response is still 200, with `delivery_code: null`. The four values mean different things and deserve different handling:

status	What it means
active	The code is in force — show it to your customer.
used	The parcel is delivered, so the code has done its job. Still returned, for a completed-order screen.
not_required	This organization does not use delivery codes at all. <code>delivery_code</code> is null — show nothing.
not_yet_issued	No code on the order. Only orders created before delivery codes existed; you will not see this on new orders.

An order that is not yours answers 404, never 403 — the two are deliberately indistinguishable, so the endpoint cannot be used to discover which references exist.

9. New: proof of delivery

GET `/api/v2/orders/pod` · scope `pod:read` · returns `application/pdf`

The record of a handover, as a document you can forward to a customer or attach to a dispute.

One section per handover, and the two delivery paths are shaped differently. A *courier* records one stop per box, so each box gets its own section. A pickup at a *PUDO point* is one collection for the whole order — a single act, verified once — so it is one section listing every box it covered. A multi-box order may show either shape.

Each section carries your own references, the recipient, the address, when it happened **in the delivery country's own timezone**, who took the parcel, and then only the evidence that was actually captured: the delivery code in full, the signature, the photo.

Query	
<code>tracking_number</code>	Starmile's order reference. Required unless <code>order_id</code> is given.
<code>order_id</code>	Your own order reference. Required unless <code>tracking_number</code> is given.
<code>merchant_tracking</code>	Optional. Narrows the document to one box. Omit it and every delivered box of the order is included.

```

curl -G 'https://api.starmile.io/api/v2/orders/pod' \
  -H 'Authorization: Bearer <access_token>' \
  -H 'Accept: application/pdf' \
  --data-urlencode 'order_id=PO-10294' \
  -o pod.pdf

```

Nothing is printed empty.

The document describes what was *recorded*, never what the proof-of-delivery policy requires today. A proof that was never captured does not appear at all — not as a blank row — and a handover with no evidence whatsoever says so in one sentence. A receipt for a handover in August must keep describing August, whatever the policy demands now.

409 and 404 are not interchangeable

Status	Meaning
409	The order — or the box you named — is not delivered yet . It exists and it is yours, so there is no reference problem to go looking for.
404	The order is not yours, or we do not hold it. The two are deliberately the same answer.

A **partly** delivered order still returns its document: the handovers that happened are real evidence, and the boxes still outstanding are listed under "Not yet delivered".

10. Scopes

The three new endpoints carry their own grants:

Scope	Grants
orders:read	GET /api/v2/orders/detail
delivery_code:read	GET /api/v2/orders/delivery-code
pod:read	GET /api/v2/orders/pod

The order detail is separate from status:read: the status pool is a movement log, and following an order does not imply reading its declared values and address. A delivery code is a *live secret* that still authorises a handover; a proof of delivery is a *record* of one that already happened. A credential may hold any of them without the others, which is why they were never merged into one grant.

No action is needed on your side. Existing credentials already carry all three scopes, and newly issued ones receive them by default. A call made without the right grant answers 403.

11. PHP SDK

The official SDK covers all three endpoints from **7.4.0**:

```
composer require starmile/partner-sdk:^7.4

$order = $starmile->v2()->orders()->detailByTrackingNumber('CMX0000012345');

// Or by your own reference instead of ours:
$order = $starmile->v2()->orders()->detail('P0-10294');

echo $order['status']; // "sorted"
echo $order['dimensions']['chargeable_weight_grams']; // 4000

$code = $starmile->v2()->orders()->deliveryCode('P0-10294');
```

```
// Or by our reference instead of yours:
$code = $starmile->v2()->orders()->deliveryCodeByTrackingNumber('CMX0000012345');

if ($code['status'] === 'active') {
    echo $code['delivery_code']; // "4821"
}

$pdf = $starmile->v2()->orders()->proofOfDelivery('PO-10294');
file_put_contents('pod.pdf', $pdf);

// Narrow it to one box:
$pdf = $starmile->v2()->orders()->proofOfDelivery('PO-10294', null, 'MT-0001');
```

12. Cutover checklist

✓	Step
1	Point order creation at /api/v2/orders and rename parcels[] to items[].
2	Store the create response's tracking_number as Starmile's reference. Stop reading order_id as ours.
3	Remove every use of parcel_id. Address boxes by item_id or merchant_tracking.
4	Update label calls: on v2, order_id is <i>yours</i> and tracking_number is <i>ours</i> .
5	Map the new status strings (\$5) onto whatever the v1 strings meant in your system, including the split ones.
6	Drain the v2 status pool from since=0, de-duplicate against v1, then keep only v2 cursors.
7	Read the order detail where you keep your own order record in sync or show the shipment to your customer; treat dimensions.measured_at null as "still my own numbers".
8	Wire up the delivery code where you show delivery details to your customer, branching on status.
9	Wire up the proof of delivery where your support desk handles disputes, treating 409 as "not delivered yet" rather than an error.
10	Run both versions side by side until the v2 pool is caught up, then stop calling v1.

The full reference, with every field and every response shape, is maintained at <https://docs.starmile.io/v2> and is always the authoritative version. A Postman collection covering both versions is linked from the same portal.