



DEVELOPERS

Contents

[Overview & base URL](#)[Enums](#)[Shipments](#)[COD](#)[Returns \(RVP\)](#)[Authentication](#)[Booking prerequisites](#)[Sales Channel orders](#)[Wallet](#)[Webhooks](#)[Responses & errors](#)[Rates & serviceability](#)[Tracking](#)[NDR](#)

Downloads

[API reference \(PDF\)](#)[Postman collection \(.json\)](#)

Import the Postman collection, set the `api_key` variable to your key, and every request is ready to run.

Overview

The Skyfleet API is a JSON REST API. All endpoints are relative to the base URL below and require authentication (except the public tracking endpoint). Use it to check rates, create and book shipments, track parcels, reconcile COD and wallet, handle NDR/returns, and receive real-time webhook events.



URL: `https://api.skyfleetnow.com/api/v1`
 Content-Type: `application/json`

All requests and responses are UTF-8 JSON. Timestamps are ISO-8601 (UTC). Monetary values are in INR (₹) unless a field name ends in Paise. Weights are in grams, dimensions in centimetres.

Authentication

Authenticate every request with an **API key**. Generate one in the dashboard under **Settings** → **Developer** → **API Keys**. The full key (`sf_live_...`) is shown **once** at creation — store it securely; you can revoke and re-create at any time. A key is scoped to your organization.

Send it in the X-API-Key header:

```
curl https://api.skyfleetnow.com/api/v1/wallet \
  -H "X-API-Key: sf_live_XXXXXXXXXXXXXXXXXXXXXXX"
```

Keep keys server-side only — never ship them in a browser, mobile app, or public repo. Rotate immediately if exposed (revoke the old key, create a new one).

Troubleshooting: {"message": "Access token required"}

The same key works on **every** endpoint — there is no separate token. This error means the X-API-Key header didn't reach the request. Check, in order:

- **Use HTTPS, not HTTP.** `http://api.skyfleetnow.com` returns a 301 redirect to `https://`, and most HTTP clients **drop custom headers on redirect** — so the key is lost. Always call `https://api.skyfleetnow.com` directly.
- **No trailing slash.** Use `/channels/orders`, not `/channels/orders/`.
- **Send the header on every request.** If it's set on one call but not another, set it as a default header on your HTTP client so all requests include it.
- **Exact header name:** X-API-Key (not X_API_Key or X-Api_Key).

Other messages: "Invalid, revoked, or expired API key" = the key was received but doesn't match (regenerate it); 401 with a key = same.

Responses & errors

Every response uses a consistent envelope.

Success



```
"success": true, "data": { ... }, "message": "optional" }
```

Paginated list

```
{
  "success": true,
  "data": [ ... ],
  "meta": { "total": 120, "page": 1, "limit": 20, "totalPages": 6 }
}
```

List endpoints accept `page` (default 1) and `limit` (default 20).

Error

```
{ "success": false, "message": "Human-readable reason", "errors": { "field": ["..."] } }
```

Status	Meaning
400	Bad request / validation failed
401	Missing / invalid / revoked API key
402	Wallet not configured (booking)
403	Forbidden — e.g. KYC not approved
404	Not found
409	Duplicate order reference (DUPLICATE_ORDER_REF)

Enums

- **ShipmentStatus:** CREATED, LABEL_GENERATED, PICKUP_SCHEDULED, PICKED_UP, IN_TRANSIT, OUT_FOR_DELIVERY, DELIVERED, DELIVERY_FAILED, RTO_INITIATED, RTO_IN_TRANSIT, RTO_DELIVERED, CANCELLED, LOST, DAMAGED
- **PaymentMode:** PREPAID, COD
- **ShipmentType:** FORWARD, REVERSE, EXCHANGE
- **NdrAction:** REATTEMPT, RTO, CHANGE_ADDRESS, CONTACT_CUSTOMER

Booking prerequisites (KYC + wallet)

POST /shipments creates a **draft** (status CREATED, no AWB, no charge). Booking a draft (assigning a courier + AWB) is gated:

- Your organization's KYC must be **APPROVED** (else 403).
- Wallet must be configured (else 402); prepaid wallets must cover the freight.

Complete KYC and top up your wallet in the dashboard before booking via API.

Rates & serviceability

POST <https://api.skyfleetnow.com/api/v1/rates/calculate>

Get all available courier quotes for a lane.

```
curl https://api.skyfleetnow.com/api/v1/rates/calculate -X POST \
  -H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
  -d '{
    "pickupPincode": "110011",
    "dropPincode": "400001",
    "weightGrams": 500,
    "paymentMode": "PREPAID",
    "lengthCm": 20, "breadthCm": 15, "heightCm": 10,
    "invoiceValue": 1499
  }'

// 200 → data: [
  {
    "courier": "Delhivery Surface", "carrierId": "delhivery-surface",
    "zone": "METRO", "chargeableWeightGrams": 500,
    "baseRate": 46.61, "codCharge": 0, "gstAmount": 12.05,
    "totalCharge": 78.95, "eddDays": 3, "isAvailable": true
  }
]
```

GET

<https://api.skyfleetnow.com/api/v1/rates/serviceability/detailed?pickup=110011&drop=400001&serviceType=FORWARD>

Per-courier serviceability (prepaid/COD/reverse support) for a lane.

GET <https://api.skyfleetnow.com/api/v1/rates/classify?pickup=110011&drop=400001>

Zone classification only.

Shipments

POST <https://api.skyfleetnow.com/api/v1/shipments>

Create a shipment draft.



```
https://api.skyfleetnow.com/api/v1/shipments -X POST \
-H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
-d '{
  "shipmentType": "FORWARD",
  "paymentMode": "COD",
  "orderReferenceNumber": "ORD-1042",
  "invoiceValue": 1499,
  "codAmount": 1499,
  "pickup": {
    "name": "Acme Warehouse", "phone": "9876543210",
    "address": "Plot 4, Industrial Area", "city": "New Delhi",
    "state": "Delhi", "pincode": "110011"
  },
  "drop": {
    "name": "Ravi Kumar", "phone": "9812345678",
    "address": "12 MG Road", "city": "Mumbai",
    "state": "Maharashtra", "pincode": "400001"
  },
  "packages": [{
    "weightGrams": 500, "lengthCm": 20, "breadthCm": 15, "heightCm": 10,
    "items": [{ "name": "Cotton T-Shirt", "sku": "TS-01", "quantity": 1, "unitPrice": 1499 } ]
  }]
}'

// 201 → data: { "shipment": { "id": "...", "status": "CREATED", ... } }
```

Required: shipmentType, paymentMode, invoiceValue, pickup, drop, packages (≥1, each with weight + dimensions + items). codAmount is required for COD. orderReferenceNumber must be unique among live orders — a duplicate returns 409 unless you pass "allowDuplicate": true. You can reference a saved pickup with pickupAddressId instead of an inline pickup object.

POST <https://api.skyfleetnow.com/api/v1/shipments/:id/book>

Book a draft — assigns courier + AWB (KYC + wallet gate applies).

```
curl https://api.skyfleetnow.com/api/v1/shipments/SHIP_ID/book -X POST \
-H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
-d '{ "carrierId": "bluedart-air" }' // pick an exact courier; omit body to auto-allocate

// 200 → data: { "id": "...", "awbNumber": "...", "carrierId": "bluedart-air", "status": "BOOKED",
//               "labelUrl": "https://api.skyfleetnow.com/api/v1/shipments/SHIP_ID/label.pdf",
//               "labelUrl": "https://api.skyfleetnow.com/api/v1/shipments/SHIP_ID/label.pdf" }
// labelUrl is a Skyfleet-branded label link — fetch it with your X-API-Key header (follows same rules as /api/v1/shipments/:id/book)
```

Pass carrierId to book a specific courier + service. Omit the body to auto-allocate by your allocation rules. **Best practice:** take the carrierId from /rates/calculate for the lane — availability and pricing vary by route and merchant.

All carrierId values



Courier	Air	Surface
Blue Dart	bluedart-air	bluedart-surface
Delhivery	delhivery-air	delhivery-surface
DTDC	dtdc-air	dtdc-surface
Xpressbees	xpressbees-air	xpressbees-surface
Ekart	—	ekart-surface
Amazon	—	amazon-surface
Professional	—	professional-surface
Shadowfax	—	shadowfax-surface

Uppercase forms (BLUEDART_AIR, ...) also work. If a carrierId isn't serviceable/priced for that lane + merchant, the booking returns a clear error rather than silently switching couriers.

GET https://api.skyfleetnow.com/api/v1/shipments?status=IN_TRANSIT&page=1&limit=20

List shipments. Filters: status, statusGroup, paymentMode, carrier, search, from, to, awb, orderId, hasAwb, unprocessed.

status takes a single **ShipmentStatus** (see Enums). For broad buckets use statusGroup:

statusGroup	Includes
pickups	Awaiting/scheduled pickup
inTransit	Picked up → out for delivery
deliveryFailed	Failed delivery attempts (NDR)
delivered	Delivered
rto	RTO initiated / in transit / delivered

Also: hasAwb=yes | no (booked vs draft), unprocessed=yes (drafts needing action).

Note: statusGroup only groups results for *filtering* — it never collapses the data. Every shipment always returns its **exact** status (e.g. OUT_FOR_DELIVERY, RTO_IN_TRANSIT). To filter to one precise state use ?status=OUT_FOR_DELIVERY, and there's a dedicated shipment.out_for_delivery webhook. You never lose granularity.

GET <https://api.skyfleetnow.com/api/v1/shipments/:id>

Full shipment detail incl. packages, items, and tracking events.

PATCH <https://api.skyfleetnow.com/api/v1/shipments/:id>

Edit a draft (only while CREATED, no AWB).



POST <https://api.skyfleetnow.com/api/v1/shipments/:id/cancel>

Cancel. Body: { "reason": "..." } (optional).

GET <https://api.skyfleetnow.com/api/v1/shipments/:id/label.pdf>

Stable Skyfleet-branded label link — 302-redirects to the PDF. This is the URL returned as `labelUrl` in booking / shipment responses.

GET <https://api.skyfleetnow.com/api/v1/shipments/:id/label-url>

Returns a 1-hour presigned label PDF URL as JSON({ `url`, `ours` }).

GET <https://api.skyfleetnow.com/api/v1/shipments/:id/invoice-url>

1-hour presigned commercial-invoice PDF URL.

Labels are Skyfleet-branded. The `labelUrl` in booking and shipment responses points at `/shipments/:id/label.pdf` — fetch it **with your X-API-Key header** and follow the redirect (any standard HTTP client does this automatically). It always serves the current Skyfleet label, so don't cache the redirected URL — re-fetch the endpoint when you need the file. (*Amazon-carrier shipments return Amazon's own label.*)

POST <https://api.skyfleetnow.com/api/v1/shipments/bulk-download>

Merge labels or zip invoices. Body: { "shipmentIds": [...], "type": "labels" | "invoices" }.

Sales Channel orders

Orders synced from a connected store (Shopify, Amazon, WooCommerce, etc.) are a **separate resource** from shipments. They live under `/channels/orders`. A channel order only appears under `/shipments` **after** it's imported — so if you're looking for freshly-synced store orders, use the endpoints here, not `/shipments`.

GET <https://api.skyfleetnow.com/api/v1/channels/orders?status=NEW&page=1&limit=20>

List synced channel orders. Filters: `status`, `integrationId`, `errorType`.

```
curl "https://api.skyfleetnow.com/api/v1/channels/orders?status=NEW" -H "X-API-Key: sf_live_..."

// 200 → data: [
  {
    "id": "co_abc", "externalOrderId": "5521", "externalOrderNumber": "#1042",
    "customerName": "Ravi Kumar", "destinationCity": "Mumbai", "destinationPincode": "4000...",
    "paymentMode": "COD", "totalValue": 1499, "status": "NEW",
    "integration": { "name": "My Shopify Store", "type": "SHOPIFY" },
    "shipmentId": null, "shipmentStatus": null, "shipmentAwb": null
  }
]
```



```
{ "total": 12, "page": 1, "limit": 20, "totalPages": 1 }
```

status values: NEW(synced, not yet imported), IMPORTED, SKIPPED, FAILED, CANCELLED_ON_CHANNEL.

GET <https://api.skyfleetnow.com/api/v1/channels>

List the merchant's connected sales channels.

GET <https://api.skyfleetnow.com/api/v1/channels/orders/stats>

Counts by status / error type.

POST <https://api.skyfleetnow.com/api/v1/channels/orders/:id/book>

One call: import + book a single order — the simplest way to ship one synced order. Returns the AWB.

```
curl "https://api.skyfleetnow.com/api/v1/channels/orders/co_abc/book" -X POST \
  -H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
  -d '{ "carrierId": "bluedart-air" }' // optional; omit to auto-allocate

// 200 → data: { "channelOrderId":"co_abc", "shipmentId":"ship_x", "awbNumber":"778123456"
```

POST <https://api.skyfleetnow.com/api/v1/channels/orders/:id/import>

Import only — turn one channel order into a shipment draft (book later). Returns the created shipment.

POST <https://api.skyfleetnow.com/api/v1/channels/orders/bulk-import>

Import many at once. Body: { "orderIds": [...], "mode": "draft" | "allocate" | "book" }.

POST <https://api.skyfleetnow.com/api/v1/channels/orders/:id/skip>

Mark a channel order as skipped (ignore it).

Typical flow: pull store orders → ship

```
GET /channels/orders?status=NEW           → un-imported synced orders
POST /channels/orders/{id}/import         → creates a draft shipment (returns shipmentId)
POST /shipments/{id}/book                 → book it (returns AWB)

// One-shot alternative:
POST /channels/orders/bulk-import { "orderIds": ["co_abc"], "mode": "book" }
```

Orders sync but don't auto-import? Turn on auto-create

By default, synced orders sit as NEW in the channel-order feed until you import them. To have Skyfleet **auto-create a draft shipment** for every incoming order, enable `autoCreateShipments` on the integration (and set a default pickup, since a shipment needs one):

PATCH <https://api.skyfleetnow.com/api/v1/channels/:id>

Update an integration's sync settings.



```
curl "https://api.skyfleetnow.com/api/v1/channels/INTEGRATION_ID" -X PATCH \
-H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
-d '{
  "autoCreateShipments": true,
  "defaultPickupAddressId": "addr_123",    // required for auto-create
  "defaultPaymentMode": "COD",            // optional fallback
  "defaultWeightGrams": 500,              // optional fallback
  "initialSyncDays": 30
}'
```

Get integration IDs from `GET /channels`, and pickup-address IDs from `GET /shipments/pickup-addresses`. With `autoCreateShipments` off, orders stay in the feed and you import them yourself (above). Either way, auto-create produces **drafts** — call `/shipments/:id/book` to book, or use `bulk-import` with `mode: "book"`.

Mental model: `/channels/orders` = raw orders from the store · `/shipments` = orders in Skyfleet's shipping pipeline (drafts + booked). Import bridges the two.

Tracking

GET `https://api.skyfleetnow.com/api/v1/shipments/:awb/track`

Authenticated tracking — returns `{ shipment, tracking }` with the live courier timeline.

GET `https://api.skyfleetnow.com/api/v1/public/track/:awb`

Public (no auth) — safe to expose to buyers. Sanitized (no full address/phone/COD).

```
curl https://api.skyfleetnow.com/api/v1/public/track/DL1234567890

// 200 → data: {
  "awbNumber": "DL1234567890", "status": "OUT_FOR_DELIVERY",
  "courier": "Delhivery", "customerName": "Ravi K.",
  "timeline": { "orderedAt": "...", "pickedUpAt": "...", "deliveredAt": null, "expectedDel",
  "events": [ { "status": "OUT_FOR_DELIVERY", "description": "...", "location": "Mumbai",
  }
```

For most integrations, prefer [webhooks](#) over polling — you get pushed a status event the moment it changes.

COD

GET <https://api.skyfleetnow.com/api/v1/cod/remittances?status=PROCESSING&page=1>
You COD remittances (payouts).

GET <https://api.skyfleetnow.com/api/v1/cod/remittance-schedule>
Per-cycle COD / freight / VAS statement.

GET <https://api.skyfleetnow.com/api/v1/cod/eligible>
COD past its cycle window, ready to remit.

GET <https://api.skyfleetnow.com/api/v1/cod/upcoming>
Forward-looking COD with expected remittance dates.

GET <https://api.skyfleetnow.com/api/v1/cod/transactions?from=2026-07-01&to=2026-07-31>
Per-parcel COD ledger (paginated). CSV at </cod/transactions.csv>.

Wallet

GET <https://api.skyfleetnow.com/api/v1/wallet>
Current balance and wallet record.

GET <https://api.skyfleetnow.com/api/v1/wallet/config>
Wallet type, COD cycle config, and bank details.

GET <https://api.skyfleetnow.com/api/v1/wallet/transactions?category=orders&page=1>
Wallet ledger (paginated). category: recharges | orders. CSV at </wallet/transactions.csv>.

NDR (failed deliveries)

GET <https://api.skyfleetnow.com/api/v1/ndr?status=PENDING&page=1>
Open NDRs. Stats at </ndr/stats>.

POST <https://api.skyfleetnow.com/api/v1/ndr/action>
Act on an NDR.

```
curl https://api.skyfleetnow.com/api/v1/ndr/action -X POST \
  -H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
  -d '{
    "ndrEventId": "NDR_ID",
    "action": "REATTEMPT",           // or RTO | CHANGE_ADDRESS | CONTACT_CUSTOMER
    "preferredDate": "2026-07-12",
    "notes": "Customer will be home after 5pm"
  }'
```

`POST https://api.skyfleetnow.com/api/v1/ndr/bulk-action``Body: { "ndrEventIds": [...], "action": "REATTEMPT", "notes": "..." }.`

Returns (Reverse pickup)

POST `https://api.skyfleetnow.com/api/v1/rvp/from-forward`

Create a reverse pickup from a delivered forward shipment.

```
curl https://api.skyfleetnow.com/api/v1/rvp/from-forward -X POST \
  -H "X-API-Key: sf_live_..." -H "Content-Type: application/json" \
  -d '{
    "forwardAwb": "DL1234567890", // or "forwardShipmentId"
    "isQcRequired": false,
    "reasonForReturn": "Size mismatch"
  }'
```

// → returns a REVERSE shipment draft; book it to schedule the pickup.

GET `https://api.skyfleetnow.com/api/v1/rvp?status=PICKED_UP&page=1`

List reverse pickups.

Webhooks (real-time events)

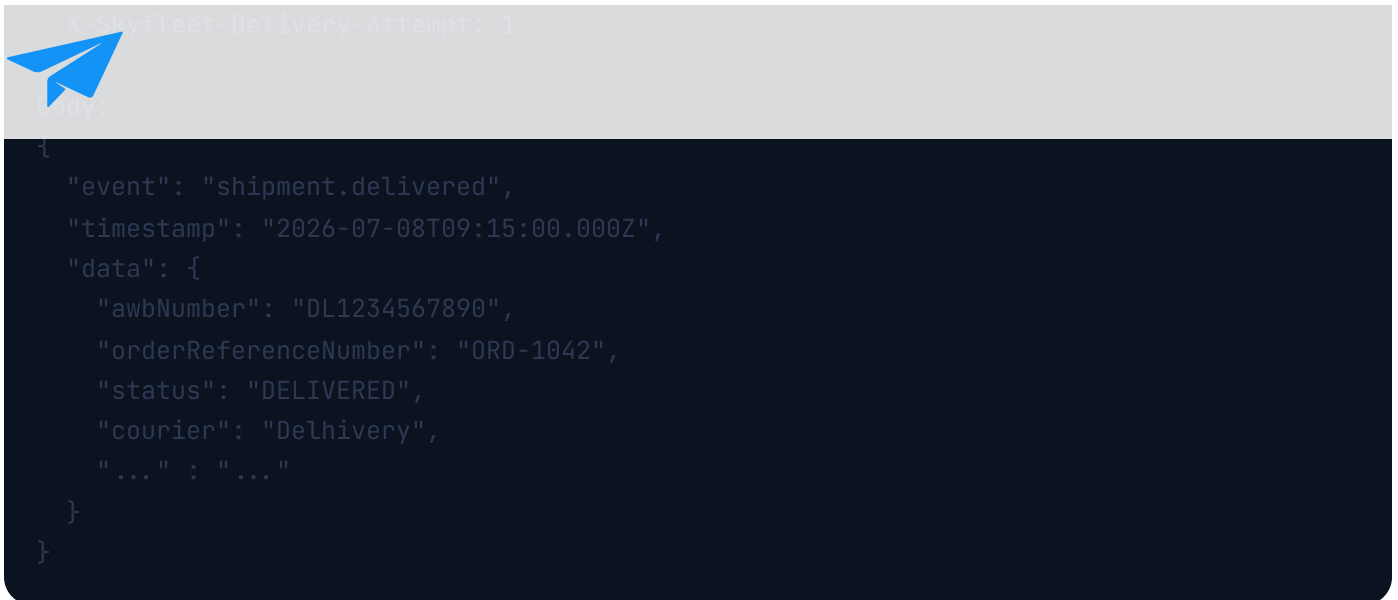
Instead of polling, subscribe to events and Skyfleet will POST them to your URL the moment they happen. Create and manage endpoints in the dashboard under **Settings → Webhooks** (set a URL, pick events, get a signing secret).

Events

<code>shipment.created</code>	<code>shipment.out_for_delivery</code>
<code>shipment.label_generated</code>	<code>shipment.delivered</code>
<code>shipment.picked_up</code>	<code>shipment.delivery_failed</code>
<code>shipment.in_transit</code>	<code>shipment.rto_delivered</code>
<code>shipment.cancelled</code>	

Payload

```
POST <your-url>
Headers:
  X-Skyfleet-Event: shipment.delivered
  X-Skyfleet-Signature: <hmac-sha256 hex of the raw body>
```



Verify the signature

Compute an HMAC-SHA256 of the **raw request body** using your endpoint's secret and compare (constant-time) to the X-SkyFleet-Signature header. Reject if they differ.

```

// Node.js (Express)
import crypto from "crypto";

app.post("/skyfleet-webhook", express.raw({ type: "*" }), (req, res) => {
  const expected = crypto.createHmac("sha256", process.env.SKYFLEET_WEBHOOK_SECRET)
    .update(req.body) // raw Buffer, not parsed JSON
    .digest("hex");
  const got = req.headers["x-skyfleet-signature"];
  if (!crypto.timingSafeEqual(Buffer.from(expected), Buffer.from(String(got)))) {
    return res.status(401).end();
  }
  const event = JSON.parse(req.body.toString());
  // ... handle event.event / event.data ...
  res.sendStatus(200); // 2xx = acknowledged
});

```

Respond 2xx to acknowledge. Non-2xx, timeouts, and network errors are retried with backoff (4xx is treated as a caller bug and not retried). Every attempt is logged in **Settings → Webhooks → Deliveries**, where you can inspect and manually retry.

Questions or need a sandbox key? Email support@skyfleetnow.com. See also our [Security](#) and [Privacy](#) pages.



COURIERS

[Delhivery](#)

[Bluedart](#)

[DTDC](#)

[Xpressbees](#)

[Ekart](#)

[Shadowfax](#)

[Smartr Logistics](#)

[All couriers →](#)

COMPARE

[Skyfleet vs ShipRocket](#)

[Skyfleet vs Pickrr](#)

[Skyfleet vs Shyplite](#)

[Skyfleet vs NimbusPost](#)

[Skyfleet vs Rapidshyp](#)

[Skyfleet vs Shipway](#)

[All comparisons →](#)

TOOLS

[Rate calculator](#)

[Pincode serviceability](#)

[Track shipment](#)

[Blog](#)

INTEGRATIONS

[Shopify](#)

[WooCommerce](#)

[EasyEcom](#)

[Unicommerce](#)

[Magento](#)

[All integrations →](#)

SHIP FROM

[Mumbai](#)

[Delhi NCR](#)

[Bangalore](#)

[Hyderabad](#)

[Chennai](#)

[Pune](#)

[Kolkata](#)

[All cities →](#)



India's smartest freight aggregator. Built in Chandigarh. Made for sellers everywhere.

🌐 skyfleetnow.com ✉ support@skyfleetnow.com 📍 Chandigarh, India

PRODUCT

[Features](#)[Couriers](#)[Integrations](#)[Pricing](#)[Track shipment](#)

COMPANY

[About](#)[Blog](#)[Help](#)[Developers](#)[Privacy](#)[Terms](#)[Security](#)

GET STARTED

[Sign in](#)[Create account](#)[Talk to us](#)[Help centre](#)

© 2026 Skyfleet · Chandigarh, India

Made with ❤️ for Indian sellers · Map data SimpleMaps