Developer Guide

3rd-Party Delivery Integration API

Welcome to the official developer documentation. This guide provides comprehensive instructions for integrating your systems with our highly scalable delivery ecosystem.

Learn how to securely authenticate, programmatically inject delivery requests, track active orders, and configure real-time webhooks.

1. Authentication & Security

The API uses Bearer Token authentication. You must generate an API Key from your Developer Dashboard and include it in the `Authorization` header of every HTTP request.

```
Authorization: Bearer YOUR_API_KEY
Accept: application/json
Content-Type: application/json
```

Security Note: Your API keys carry significant privileges. Do not expose them in public repositories or client-side code.

Webhook Authentication

When our server sends webhooks to your endpoint, we include an `Authorization` header using the webhook secret you configured in your dashboard. Use this to verify the payload is genuinely from us.

```
Authorization: Bearer YOUR_WEBHOOK_SECRET
```

2. Delivery Endpoints

GET /api/v1/deliveries/categories

Retrieves the list of active parcel categories and their IDs. Use these IDs when estimating or creating a delivery.

POST /api/v1/deliveries/estimate

Simulates an order creation to return a 100% accurate price quote based on current zones, distances, and applicable taxes without actually creating a delivery request.

POST /api/v1/deliveries

Creates a new delivery request within the system.

Request Body (JSON)

Parameter	Type	Description
pickup_address Required	string	The full formatted address for pickup.
pickup_coordinates Required	object	Contains `lat` and `lng` floats.
dropoff_address Required	string	The full formatted address for dropoff.
dropoff_coordinates Required	object	Contains `lat` and `lng` floats.
customer_name Required	string	Name of the recipient.
customer_phone Required	string	Contact number for the recipient.

Success Response (201 Created)


```
{ "status": "success", "message": "Delivery request created
successfully", "data": { "delivery_id": 1042, "tracking_number":
"TRK-987654321", "status": "pending", "delivery_charge": 120.00,
"total_amount": 138.00 } }
```

GET /api/v1/deliveries/{id}

Retrieves the live status and details of a specific delivery.

PATCH /api/v1/deliveries/{id}/cancel

Cancels a delivery if it has not yet been picked up.

POST /api/v1/deliveries/webhook-test

Triggers a test webhook payload (`system.heartbeat`) to your configured Webhook URL to verify connectivity.

3. Webhooks & Event Streaming

Webhooks allow your application to receive automated HTTP POST requests from our servers whenever a delivery changes status. You can configure your Webhook Endpoint URL in the Developer Dashboard.

Event Lifecycle

- `delivery.created`: Fired when an order is injected.
- `delivery.assigned`: Fired when a rider accepts the order.
- `delivery.picked_up`: Fired when the rider collects the package.
- `delivery.location_update`: Fired every time the driver's GPS location is updated.
- `delivery.delivered`: Fired upon successful completion.
- `delivery.cancelled`: Fired if the order is aborted.
- `system.heartbeat`: Fired when triggering the webhook test endpoint.

Payload Structure


```
{ "event": "delivery.location_update", "delivery_id": "1042",
"tracking_number": "TRK-987654321", "driver": { "latitude": 9.0305,
"longitude": 38.7456, "heading": 180.5 }, "timestamp": "2026-07-06T14:30:00Z"
}
```


Your server must respond with a **2xx** HTTP status code within 5 seconds to acknowledge receipt. If the request times out or returns a non-2xx code, the payload will be marked as failed.