

# API Documentation: IMSOP Platform

---

## 1. Introduction

---

This document provides comprehensive API documentation for the **Intelligent Multi-Cloud Supply Chain & Operations Platform (IMSOP)**. The platform exposes multiple APIs to support various integration patterns and use cases.

## 2. API Overview

---

The IMSOP platform provides the following API types:

API Type	Protocol	Use Case	Authentication
REST API	HTTP/HTTPS	Standard CRUD operations, third-party integrations	OAuth 2.0
GraphQL API	HTTP/HTTPS	Efficient data fetching for frontend applications	OAuth 2.0
WebSocket API	WebSocket	Real-time updates, telemetry ingestion, live chat	OAuth 2.0
AsyncAPI	Kafka	Event-driven integration between microservices	Internal

## 3. Authentication

---

### 3.1 OAuth 2.0 Flow

All external APIs require OAuth 2.0 authentication using the Authorization Code flow with PKCE for web applications or Client Credentials flow for server-to-server integrations.

**Token Endpoint:** POST /auth/token

### Request:

```
POST /auth/token HTTP/1.1
Host: api.imsop.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id=your_client_id
&client_secret=your_client_secret
&scope=operations:read analytics:read
```

### Response:

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "operations:read analytics:read"
}
```

## 3.2 Using Access Tokens

Include the access token in the Authorization header for all API requests:

```
GET /api/v1/operations/shipments HTTP/1.1
Host: api.imsop.example.com
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

## 4. REST API

---

### 4.1 Base URL

**Production:** https://api.imsop.example.com/api/v1

**Staging:** https://api-staging.imsop.example.com/api/v1

## 4.2 Operations API

### List Shipments

Retrieve a list of shipments with optional filtering and pagination.

**Endpoint:** `GET /operations/shipments`

#### Query Parameters:

- `status` (optional): Filter by shipment status (e.g., `IN_TRANSIT`, `DELIVERED`)
- `carrier_id` (optional): Filter by carrier ID
- `page` (optional): Page number (default: 1)
- `limit` (optional): Items per page (default: 20, max: 100)

#### Example Request:

```
GET /api/v1/operations/shipments?status=IN_TRANSIT&limit=10 HTTP/1.1
Host: api.imsop.example.com
Authorization: Bearer {access_token}
```

#### Example Response:

```
{
  "data": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "tracking_number": "1Z999AA10123456784",
      "status": "IN_TRANSIT",
      "origin": {
        "address": "123 Main St",
        "city": "New York",
        "state": "NY",
        "postal_code": "10001",
        "country": "US"
      },
      "destination": {
        "address": "456 Oak Ave",
        "city": "Los Angeles",
        "state": "CA",
        "postal_code": "90001",
        "country": "US"
      },
      "estimated_delivery": "2026-01-10T18:00:00Z",
      "carrier_id": "carrier-123",
      "created_at": "2026-01-05T10:30:00Z",
      "updated_at": "2026-01-07T08:15:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 10,
    "total": 45,
    "total_pages": 5
  }
}
```

## Get Shipment Details

Retrieve detailed information about a specific shipment.

**Endpoint:** `GET /operations/shipments/{id}`

### Path Parameters:

- `id` (required): Shipment ID (UUID)

### Example Request:

```
GET /api/v1/operations/shipments/550e8400-e29b-41d4-a716-446655440000 HTTP/1.1
Host: api.imsop.example.com
Authorization: Bearer {access_token}
```

### Example Response:

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "tracking_number": "1Z999AA10123456784",
  "status": "IN_TRANSIT",
  "origin": {
    "address": "123 Main St",
    "city": "New York",
    "state": "NY",
    "postal_code": "10001",
    "country": "US"
  },
  "destination": {
    "address": "456 Oak Ave",
    "city": "Los Angeles",
    "state": "CA",
    "postal_code": "90001",
    "country": "US"
  },
  "estimated_delivery": "2026-01-10T18:00:00Z",
  "actual_delivery": null,
  "carrier_id": "carrier-123",
  "events": [
    {
      "timestamp": "2026-01-05T10:30:00Z",
      "status": "CREATED",
      "location": "New York, NY",
      "description": "Shipment created"
    },
    {
      "timestamp": "2026-01-06T14:20:00Z",
      "status": "IN_TRANSIT",
      "location": "Philadelphia, PA",
      "description": "In transit to next facility"
    }
  ],
  "created_at": "2026-01-05T10:30:00Z",
  "updated_at": "2026-01-07T08:15:00Z"
}
```

## Create Shipment

Create a new shipment in the system.

**Endpoint:** POST /operations/shipments

**Request Body:**

```
{
  "tracking_number": "1Z999AA10123456784",
  "origin": {
    "address": "123 Main St",
    "city": "New York",
    "state": "NY",
    "postal_code": "10001",
    "country": "US"
  },
  "destination": {
    "address": "456 Oak Ave",
    "city": "Los Angeles",
    "state": "CA",
    "postal_code": "90001",
    "country": "US"
  },
  "estimated_delivery": "2026-01-10T18:00:00Z",
  "carrier_id": "carrier-123"
}
```

**Example Response:**

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "tracking_number": "1Z999AA10123456784",
  "status": "CREATED",
  "created_at": "2026-01-07T09:00:00Z"
}
```

## Update Shipment Status

Update the status of an existing shipment.

**Endpoint:** PATCH /operations/shipments/{id}

**Path Parameters:**

- `id` (required): Shipment ID (UUID)

#### Request Body:

```
{
  "status": "DELIVERED",
  "actual_delivery": "2026-01-09T16:30:00Z",
  "location": "Los Angeles, CA",
  "notes": "Delivered to recipient"
}
```

#### Example Response:

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "status": "DELIVERED",
  "updated_at": "2026-01-09T16:30:00Z"
}
```

## 4.3 Analytics API

### Run Prediction

Run a machine learning model to generate predictions.

**Endpoint:** `POST /analytics/predict`

#### Request Body:



```
{
  "model_type": "delay_prediction",
  "shipment_id": "550e8400-e29b-41d4-a716-446655440000",
  "features": {
    "carrier_id": "carrier-123",
    "origin_city": "New York",
    "destination_city": "Los Angeles",
    "weather_conditions": "clear",
    "traffic_level": "moderate"
  }
}
```

### Example Response:

```
{
  "prediction_id": "pred-12345",
  "shipment_id": "550e8400-e29b-41d4-a716-446655440000",
  "prediction_type": "delay_prediction",
  "predicted_delay_hours": 2.5,
  "confidence": 0.87,
  "factors": [
    "High traffic in destination city",
    "Weather conditions in transit route"
  ],
  "model_version": "v1.2.3",
  "created_at": "2026-01-07T09:30:00Z"
}
```

### Submit Analytics Job

Submit a batch analytics job for processing.

**Endpoint:** POST /analytics/jobs

**Request Body:**

```
{
  "job_type": "batch_analytics",
  "input_data_path": "s3://imsop-data/input/shipments-2026-01.csv",
  "output_data_path": "s3://imsop-data/output/analytics-2026-01/",
  "parameters": {
    "analysis_type": "demand_forecast",
    "time_range": "30d"
  }
}
```

### Example Response:

```
{
  "job_id": "job-67890",
  "status": "PENDING",
  "created_at": "2026-01-07T10:00:00Z",
  "estimated_completion": "2026-01-07T11:00:00Z"
}
```

### Get Job Status

Retrieve the status of an analytics job.

**Endpoint:** `GET /analytics/jobs/{job_id}`

### Example Response:

```
{
  "job_id": "job-67890",
  "job_type": "batch_analytics",
  "status": "COMPLETED",
  "input_data_path": "s3://imsop-data/input/shipments-2026-01.csv",
  "output_data_path": "s3://imsop-data/output/analytics-2026-01/",
  "created_at": "2026-01-07T10:00:00Z",
  "started_at": "2026-01-07T10:05:00Z",
  "completed_at": "2026-01-07T10:45:00Z"
}
```

## 4.4 Chatbot API

### Send Message

Send a message to the chatbot and receive a response.

**Endpoint:** POST /chatbot/message

### Request Body:

```
{
  "conversation_id": "conv-123",
  "message": "What is the status of shipment 1Z999AA10123456784?",
  "user_id": "user-456"
}
```

### Example Response:

```
{
  "conversation_id": "conv-123",
  "response": "Shipment 1Z999AA10123456784 is currently in transit. It departed from Philadelphia, PA at 2:20 PM yesterday and is expected to arrive in Los Angeles, CA on January 10th at 6:00 PM.",
  "confidence": 0.95,
  "timestamp": "2026-01-07T11:00:00Z"
}
```

## 5. GraphQL API

---

### 5.1 Endpoint

**URL:** POST /graphql

## 5.2 Example Queries

### Dashboard Data Query

```
query DashboardData {  
  shipments(status: IN_TRANSIT, limit: 10) {  
    id  
    trackingNumber  
    status  
    estimatedDelivery  
    origin {  
      city  
      state  
    }  
    destination {  
      city  
      state  
    }  
  }  
  
  predictions(type: DELAY, limit: 5) {  
    id  
    shipmentId  
    predictedDelayHours  
    confidence  
    factors  
  }  
  
  recentOrders(limit: 5) {  
    id  
    orderNumber  
    status  
    totalAmount  
    orderDate  
  }  
}
```

## Shipment Details Query

```
query ShipmentDetails($id: ID!) {  
  shipment(id: $id) {  
    id  
    trackingNumber  
    status  
    origin {  
      address  
      city  
      state  
      postalCode  
      country  
    }  
    destination {  
      address  
      city  
      state  
      postalCode  
      country  
    }  
    estimatedDelivery  
    actualDelivery  
    events {  
      timestamp  
      status  
      location  
      description  
    }  
    predictions {  
      predictedDelayHours  
      confidence  
      factors  
    }  
  }  
}
```

## 5.3 Example Mutations

### Create Shipment Mutation

```
mutation CreateShipment($input: CreateShipmentInput!) {  
  createShipment(input: $input) {  
    id  
    trackingNumber  
    status  
    createdAt  
  }  
}
```

### Variables:

```
{  
  "input": {  
    "trackingNumber": "1Z999AA10123456784",  
    "origin": {  
      "address": "123 Main St",  
      "city": "New York",  
      "state": "NY",  
      "postalCode": "10001",  
      "country": "US"  
    },  
    "destination": {  
      "address": "456 Oak Ave",  
      "city": "Los Angeles",  
      "state": "CA",  
      "postalCode": "90001",  
      "country": "US"  
    },  
    "estimatedDelivery": "2026-01-10T18:00:00Z",  
    "carrierId": "carrier-123"  
  }  
}
```

## 6. WebSocket API

---

### 6.1 Telemetry Ingestion

**Endpoint:** `wss://api.imsop.example.com/ingest/telemetry`

**Connection:**

```
const ws = new WebSocket('wss://api.imsop.example.com/ingest/telemetry');

ws.onopen = () => {
  // Authenticate
  ws.send(JSON.stringify({
    type: 'auth',
    token: 'your_access_token'
  }));

  // Send telemetry data
  ws.send(JSON.stringify({
    type: 'telemetry',
    device_id: 'device-123',
    event_type: 'location_update',
    payload: {
      latitude: 40.7128,
      longitude: -74.0060,
      speed: 55.5,
      heading: 270
    },
    timestamp: '2026-01-07T12:00:00Z'
  }));
};
```

### 6.2 Live Dashboard Updates

**Endpoint:** `wss://api.imsop.example.com/viz/live`

**Connection:**

```
const ws = new WebSocket('wss://api.imsop.example.com/viz/live');

ws.onopen = () => {
  ws.send(JSON.stringify({
    type: 'auth',
    token: 'your_access_token'
  }));

  ws.send(JSON.stringify({
    type: 'subscribe',
    channels: ['shipments', 'predictions', 'alerts']
  }));
};

ws.onmessage = (event) => {
  const data = JSON.parse(event.data);
  console.log('Received update:', data);
};
```

## 7. AsyncAPI (Kafka Events)

### 7.1 Event Topics

Topic	Description	Producers	Consumers
telemetry.received	Raw telemetry data from IoT devices	Ingestion Service	Analytics Service
shipment.created	New shipment created	Operations Service	Analytics, Notification
shipment.status_changed	Shipment status updated	Operations Service	Analytics, Notification
prediction.generated	ML prediction generated	Analytics Service	Operations, Visualization
anomaly.detected	Anomaly detected in data	Analytics Service	Operations, Notification



## 7.2 Event Schemas

### Shipment Status Changed Event

```
{
  "event_id": "evt-12345",
  "event_type": "shipment.status_changed",
  "timestamp": "2026-01-07T12:30:00Z",
  "data": {
    "shipment_id": "550e8400-e29b-41d4-a716-446655440000",
    "tracking_number": "1Z999AA10123456784",
    "old_status": "IN_TRANSIT",
    "new_status": "OUT_FOR_DELIVERY",
    "location": "Los Angeles, CA",
    "updated_by": "system"
  }
}
```

### Anomaly Detected Event

```
{
  "event_id": "evt-67890",
  "event_type": "anomaly.detected",
  "timestamp": "2026-01-07T13:00:00Z",
  "data": {
    "anomaly_type": "unusual_delay",
    "shipment_id": "550e8400-e29b-41d4-a716-446655440000",
    "severity": "HIGH",
    "description": "Shipment delayed by 48 hours, significantly exceeding normal variance",
    "confidence": 0.92,
    "factors": [
      "Weather conditions",
      "Carrier performance degradation"
    ]
  }
}
```

## 8. Rate Limiting

---

All APIs are subject to rate limiting to ensure fair usage and system stability.

API Type	Rate Limit	Window
REST API	1000 requests	Per hour per user
GraphQL API	500 queries	Per hour per user
WebSocket	100 messages	Per minute per connection

### Rate Limit Headers:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 987
X-RateLimit-Reset: 1704632400
```

## 9. Error Handling

---

### 9.1 Error Response Format

All API errors follow a consistent format:

```
{
  "error": {
    "code": "RESOURCE_NOT_FOUND",
    "message": "Shipment with ID 550e8400-e29b-41d4-a716-446655440000 not found",
    "details": {
      "resource_type": "shipment",
      "resource_id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "timestamp": "2026-01-07T14:00:00Z",
    "request_id": "req-abc123"
  }
}
```

## 9.2 Common Error Codes

HTTP Status	Error Code	Description
400	INVALID_REQUEST	Request validation failed
401	UNAUTHORIZED	Authentication required or failed
403	FORBIDDEN	Insufficient permissions
404	RESOURCE_NOT_FOUND	Requested resource does not exist
409	CONFLICT	Resource conflict (e.g., duplicate)
429	RATE_LIMIT_EXCEEDED	Too many requests
500	INTERNAL_ERROR	Server error
503	SERVICE_UNAVAILABLE	Service temporarily unavailable

## 10. API Versioning

---

The IMSOP API uses URL-based versioning. The current version is `v1`.

**Example:** `https://api.imsop.example.com/api/v1/operations/shipments`

Breaking changes will result in a new API version (e.g., `v2`). Non-breaking changes (new fields, new endpoints) will be added to the existing version.

## 11. SDK and Client Libraries

---

Official SDKs are available for the following languages:

- **JavaScript/TypeScript:** `npm install @imsop/sdk`
- **Python:** `pip install imsop-sdk`
- **C#/.NET:** `dotnet add package IMSOP.SDK`
- **Java:** Maven/Gradle dependency available

**Example Usage (JavaScript):**

```
import { IMSOPClient } from '@imsop/sdk';

const client = new IMSOPClient({
  apiUrl: 'https://api.imsop.example.com',
  clientId: 'your_client_id',
  clientSecret: 'your_client_secret'
});

const shipments = await client.operations.listShipments({
  status: 'IN_TRANSIT',
  limit: 10
});
```

---

**Document Version:** 1.0

**Last Updated:** January 7, 2026

**Author:** Manus AI