

Smart Delivery Management System - API Documentation

Overview

This document provides detailed information about the API endpoints available in the Smart Delivery Management System. The API allows you to manage delivery partners, orders, and assignments through RESTful endpoints.

Base URL

All API endpoints are relative to the base URL of your deployment.

```
https://your-domain.com/api
```

Data Models

DeliveryPartner

```
{
  _id: string;
  name: string;
  email: string;
  phone: string;
  status: "active" | "inactive";
  currentLoad: number;
  areas: string[];
  shift: {
    start: string; // Format: "HH:MM"
    end: string;   // Format: "HH:MM"
  };
  metrics: {
    rating: number;
    completedOrders: number;
    cancelledOrders: number;
  };
}
```

Order

```
{
  _id: string;
  orderNumber: string;
  customer: {
    name: string;
    phone: string;
    address: string;
  };
  area: string;
  items: Array<{
    name: string;
    quantity: number;
    price: number;
  }>;
  status: "pending" | "assigned" | "picked" | "delivered";
  assignedTo?: string; // Partner ID
  scheduledFor: string; // Format: "HH:MM"
  totalAmount: number;
  createdAt: Date;
  updatedAt: Date;
}
```

Assignment

```
{
  orderId: string;
  partnerId: string;
  timestamp: Date;
  status: "success" | "failed";
  reason?: string; // Reason for failure if status is "failed"
}
```

AssignmentMetrics

```
{
  totalAssigned: number;
  successRate: number; // Percentage
  averageTime: number; // In minutes
  failureReasons: Array<{
    reason: string;
    count: number;
  }>;
}
```

API Endpoints

Partners

Get All Partners

Retrieves all delivery partners and calculates metrics.

URL: /partners

Method: GET

Response:

- Status Code: 200 OK
- Content-Type: application/json

```
{
  "partners": [
    {
      "_id": "p1",
      "name": "John Doe",
      "email": "john@example.com",
      "phone": "+1234567890",
      "status": "active",
      "currentLoad": 2,
      "areas": ["Downtown", "Midtown"],
      "shift": {
        "start": "09:00",
        "end": "17:00"
      },
      "metrics": {
        "rating": 4.8,
        "completedOrders": 156,
        "cancelledOrders": 3
      }
    },
    // More partners...
  ],
  "metrics": {
    "totalActive": 2,
    "avgRating": 4.73,
    "topAreas": ["Downtown", "Midtown", "Uptown"]
  }
}
```

Create Partner

Creates a new delivery partner.

URL: /partners

Method: POST

Request Body:

```
{
  "name": "New Partner",
  "email": "new@example.com",
  "phone": "+1234567890",
  "status": "active",
  "currentLoad": 0,
  "areas": ["Downtown", "Midtown"],
  "shift": {
    "start": "09:00",
    "end": "17:00"
  },
  "metrics": {
    "rating": 5.0,
    "completedOrders": 0,
    "cancelledOrders": 0
  }
}
```

Response:

- Status Code: 201 Created
- Content-Type: application/json

```
{
  "_id": "p4",
  "name": "New Partner",
  "email": "new@example.com",
  "phone": "+1234567890",
  "status": "active",
  "currentLoad": 0,
  "areas": ["Downtown", "Midtown"],
  "shift": {
    "start": "09:00",
    "end": "17:00"
  },
  "metrics": {
    "rating": 5.0,
    "completedOrders": 0,
    "cancelledOrders": 0
  }
}
```

Get Partner by ID

Retrieves a specific delivery partner by ID.

URL: /partners/:id

Method: GET

URL Parameters:

- **id:** The ID of the partner to retrieve

Response:

- **Status Code:** 200 OK
- **Content-Type:** application/json

```
{
  "_id": "p1",
  "name": "John Doe",
  "email": "john@example.com",
  "phone": "+1234567890",
  "status": "active",
  "currentLoad": 2,
  "areas": ["Downtown", "Midtown"],
  "shift": {
    "start": "09:00",
    "end": "17:00"
  },
  "metrics": {
    "rating": 4.8,
    "completedOrders": 156,
    "cancelledOrders": 3
  }
}
```

Error Response:

- **Status Code:** 404 Not Found
- **Content-Type:** application/json

```
{
  "error": "Partner not found"
}
```

Update Partner

Updates an existing delivery partner.

URL: /partners/:id

Method: PUT

URL Parameters:

- **id:** The ID of the partner to update

Request Body:

```
{
  "name": "John Doe Updated",
  "email": "john@example.com",
  "phone": "+1234567890",
  "status": "active",
  "currentLoad": 2,
  "areas": ["Downtown", "Midtown", "Uptown"],
  "shift": {
    "start": "08:00",
    "end": "16:00"
  },
  "metrics": {
    "rating": 4.8,
    "completedOrders": 156,
    "cancelledOrders": 3
  }
}
```

Response:

- **Status Code:** 200 OK
- **Content-Type:** application/json

```
{
  "_id": "p1",
  "name": "John Doe Updated",
  "email": "john@example.com",
  "phone": "+1234567890",
  "status": "active",
  "currentLoad": 2,
  "areas": ["Downtown", "Midtown", "Uptown"],
  "shift": {
    "start": "08:00",
    "end": "16:00"
  },
  "metrics": {
    "rating": 4.8,
    "completedOrders": 156,
    "cancelledOrders": 3
  }
}
```

Error Response:

- Status Code: 404 Not Found
- Content-Type: application/json

```
{
  "error": "Partner not found"
}
```

Delete Partner

Deletes a delivery partner.

URL: /partners/:id

Method: DELETE

URL Parameters:

- **id:** The ID of the partner to delete

Response:

- Status Code: 200 OK
- Content-Type: application/json


```
{
  "success": true
}
```

Error Response:

- Status Code: 404 Not Found
- Content-Type: application/json

```
{
  "error": "Partner not found"
}
```

Orders

Get Orders

Retrieves orders with optional filtering.

URL: `/orders`

Method: GET

Query Parameters:

- `status` (optional): Comma-separated list of order statuses to filter by
- `areas` (optional): Comma-separated list of areas to filter by
- `date` (optional): Date to filter by (format: YYYY-MM-DD)

Example Request:

```
GET /api/orders?status=pending,assigned&areas=Downtown&date=2023-05-15
```

Response:

- Status Code: 200 OK
- Content-Type: application/json

```

{
  "orders": [
    {
      "_id": "o1",
      "orderNumber": "ORD-1001",
      "customer": {
        "name": "Alice Brown",
        "phone": "+1234567890",
        "address": "123 Main St, Anytown"
      },
      "area": "Downtown",
      "items": [
        {
          "name": "Burger",
          "quantity": 2,
          "price": 12.99
        },
        {
          "name": "Fries",
          "quantity": 1,
          "price": 4.99
        }
      ],
      "status": "pending",
      "scheduledFor": "14:30",
      "totalAmount": 30.97,
      "createdAt": "2023-05-15T10:30:00.000Z",
      "updatedAt": "2023-05-15T10:30:00.000Z"
    }
    // More orders...
  ],
  "filters": {
    "status": ["pending", "assigned"],
    "areas": ["Downtown"],
    "date": "2023-05-15"
  }
}

```

Assign Order

Assigns an order to a delivery partner.

URL: /orders/assign

Method: POST

Request Body:

```
{
  "orderId": "o1",
  "partnerId": "p1"
}
```

Response:

- Status Code: 200 OK
- Content-Type: application/json

```
{
  "_id": "o1",
  "orderNumber": "ORD-1001",
  "customer": {
    "name": "Alice Brown",
    "phone": "+1234567890",
    "address": "123 Main St, Anytown"
  },
  "area": "Downtown",
  "items": [
    {
      "name": "Burger",
      "quantity": 2,
      "price": 12.99
    },
    {
      "name": "Fries",
      "quantity": 1,
      "price": 4.99
    }
  ],
  "status": "assigned",
  "assignedTo": "p1",
  "scheduledFor": "14:30",
  "totalAmount": 30.97,
  "createdAt": "2023-05-15T10:30:00.000Z",
  "updatedAt": "2023-05-15T10:30:00.000Z"
}
```

Error Response:

- Status Code: 404 Not Found
- Content-Type: application/json

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

Update Order Status

Updates the status of an order.

URL: `/orders/:id/status`

Method: PUT

URL Parameters:

- `id`: The ID of the order to update

Request Body:

```
{  
  "status": "picked"  
}
```

Response:

- Status Code: 200 OK
- Content-Type: application/json

```
{
  "_id": "o1",
  "orderNumber": "ORD-1001",
  "customer": {
    "name": "Alice Brown",
    "phone": "+1234567890",
    "address": "123 Main St, Anytown"
  },
  "area": "Downtown",
  "items": [
    {
      "name": "Burger",
      "quantity": 2,
      "price": 12.99
    },
    {
      "name": "Fries",
      "quantity": 1,
      "price": 4.99
    }
  ],
  "status": "picked",
  "assignedTo": "p1",
  "scheduledFor": "14:30",
  "totalAmount": 30.97,
  "createdAt": "2023-05-15T10:30:00.000Z",
  "updatedAt": "2023-05-15T10:30:00.000Z"
}
```

Error Response:

- Status Code: 404 Not Found
- Content-Type: application/json

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

Assignments

Get Assignments

Retrieves all active assignments and partner availability data.

URL: /assignments

Method: GET

Response:

- Status Code: 200 OK
- Content-Type: application/json

```
{
  "activeAssignments": [
    {
      "orderId": "o2",
      "partnerId": "p1",
      "timestamp": "2023-05-15T11:30:00.000Z",
      "status": "success"
    },
    {
      "orderId": "o3",
      "partnerId": "p2",
      "timestamp": "2023-05-15T10:15:00.000Z",
      "status": "success"
    },
    {
      "orderId": "o5",
      "partnerId": "p3",
      "timestamp": "2023-05-15T14:45:00.000Z",
      "status": "failed",
      "reason": "Partner unavailable"
    }
    // More assignments...
  ],
  "partners": {
    "available": 5,
    "busy": 8,
    "offline": 3
  }
}
```

Create Assignment

Creates a new assignment manually.

URL: /assignments

Method: POST

Request Body:

```
{  
  "orderId": "o1",  
  "partnerId": "p1"  
}
```

Response:

- Status Code: 201 Created
- Content-Type: application/json

```
{  
  "orderId": "o1",  
  "partnerId": "p1",  
  "timestamp": "2023-05-15T16:30:00.000Z",  
  "status": "success"  
}
```

Get Assignment Metrics

Retrieves metrics about the assignment system.

URL: /assignments/metrics

Method: GET

Response:

- Status Code: 200 OK
- Content-Type: application/json

```
{
  "totalAssigned": 156,
  "successRate": 92,
  "averageTime": 3.5,
  "failureReasons": [
    {
      "reason": "Partner unavailable",
      "count": 8
    },
    {
      "reason": "Area not covered",
      "count": 3
    },
    {
      "reason": "Load capacity exceeded",
      "count": 2
    }
  ]
}
```

Run Smart Assignment Algorithm

Runs the smart assignment algorithm to automatically assign pending orders to available partners.

URL: /assignments/run

Method: POST

Response:

- Status Code: 200 OK
- Content-Type: application/json


```
{
  "activeAssignments": [
    {
      "orderId": "o1",
      "partnerId": "p3",
      "timestamp": "2023-05-15T16:45:00.000Z",
      "status": "success"
    },
    {
      "orderId": "o5",
      "partnerId": "p2",
      "timestamp": "2023-05-15T16:45:00.000Z",
      "status": "failed",
      "reason": "Partner unavailable"
    }
  ],
  "metrics": {
    "totalAssigned": 158,
    "successRate": 91,
    "averageTime": 3.6,
    "failureReasons": [
      {
        "reason": "Partner unavailable",
        "count": 9
      },
      {
        "reason": "Area not covered",
        "count": 3
      },
      {
        "reason": "Load capacity exceeded",
        "count": 2
      }
    ]
  }
}
```

Error Handling

All API endpoints follow a consistent error response format:

```
{
  "error": "Error message describing what went wrong"
}
```

Common HTTP status codes:

- 200 OK: The request was successful
- 201 Created: A new resource was successfully created
- 400 Bad Request: The request was malformed or invalid
- 404 Not Found: The requested resource was not found
- 500 Internal Server Error: An unexpected error occurred on the server

Rate Limiting

The API implements rate limiting to prevent abuse. Clients are limited to 100 requests per minute. When the rate limit is exceeded, the API will respond with a 429 Too Many Requests status code.

Authentication

The API uses JWT (JSON Web Token) authentication. Include the JWT token in the Authorization header of your requests:

```
Authorization: Bearer <your_token>
```

Versioning

The current API version is v1. The version is included in the URL path:

```
/api/v1/partners
```

Future versions will be available at /api/v2/, /api/v3/, etc.