



Complete API Reference (Canonical)

Version: 3.1.0

Base URL: `http://localhost:5000` (Development) | `https://driversklub-backend.onrender.com/` (Production) **Last**

Updated: January 9, 2026

Table of Contents

- [Authentication](#)
- [Core Endpoints](#)
 - [Health Check](#)
 - [Auth](#)
 - [Users](#)
 - [Drivers](#)
 - [Fleet Management](#)
 - [Fleet Managers](#)
 - [Vehicles](#)
 - [Assignments](#)
 - [Attendance](#)
 - [Trips](#)
 - [Admin Trips](#)
 - [Pricing](#)
- [Partner Integrations](#)
 - [MakeMyTrip \(MMT\)](#)
- [Webhooks](#)
- [Error Handling](#)
- [Rate Limiting](#)
- [Support](#)
 - [Payment System](#)
- [Rapido \(Fleet\)](#)

Authentication

All authenticated endpoints require a valid JWT token in the `Authorization` header:

```
Authorization: Bearer <access_token>
```

User Roles

- `SUPER_ADMIN` - Full system access
- `OPERATIONS` - Operations team access
- `MANAGER` - Fleet/Hub manager access
- `DRIVER` - Driver-specific access

Core Endpoints

Health Check

GET `/health`

Check server and database connectivity.

Authentication: None

Response:

```
{
  "status": "ok",
  "service": "drivers-klub-backend",
  "database": "connected",
  "timestamp": "2025-12-26T15:51:32.202Z"
}
```

Auth

POST `/auth/send-otp`

Send OTP to user's phone number.

Authentication: None

Request Body:

```
{
  "phone": "9876543210"
}
```

Response:

```
{
  "success": true,
  "message": "OTP sent successfully"
}
```

Notes:

- In development, OTP is logged to console
- OTP expires in 5 minutes (configurable)
- Maximum 3 verification attempts per OTP

POST `/auth/verify-otp`

Verify OTP and authenticate user.

Authentication: None

Request Body:

```
{
  "phone": "9876543210",
  "otp": "123456",
  "verifiedKey": "pass" // Optional: Dev bypass key
}
```

Response:

```
{
  "success": true,
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "user": {
      "id": "uuid",
      "name": "Rajesh Kumar",
      "phone": "9876543210",
      "role": "DRIVER",
      "isActive": true
    }
  }
}
```

Security:

- OTP is deleted after successful verification (prevents reuse)
- Dev bypass only works in non-production environments

POST `/auth/refresh`

Refresh access token using refresh token.

Authentication: None

Request Body:

```
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Response:

```
{
  "success": true,
  "data": {
```

```
    "accessToken": "new_access_token",
    "refreshToken": "new_refresh_token"
  }
}
```

POST /auth/logout

Logout user and revoke tokens.

Authentication: Optional (Bearer token)

Request Body:

```
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Response:

```
{
  "success": true,
  "message": "Logged out successfully"
}
```

Users

All user endpoints require authentication.

POST /users

Create a new user.

Authentication: Required

Roles: SUPER_ADMIN

Request Body:

```
{
  "name": "John Doe",
  "phone": "9876543210",
  "role": "DRIVER",
  "isActive": true
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "John Doe",
    "phone": "9876543210",
    "role": "DRIVER",
    "isActive": true,
    "createdAt": "2025-12-26T10:00:00.000Z"
  }
}
```

GET /users

Get all users.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "name": "John Doe",
      "phone": "9876543210",
      "role": "DRIVER",
      "isActive": true
    }
  ]
}
```

GET /users/:id

Get user by ID.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "John Doe",
    "phone": "9876543210",
    "role": "DRIVER",
    "isActive": true
  }
}
```

```
}  
}
```

PATCH /users/:id/deactivate

Deactivate a user.

Authentication: Required

Roles: SUPER_ADMIN

Response:

```
{  
  "success": true,  
  "message": "User deactivated successfully"  
}
```

Drivers

All driver endpoints require authentication.

POST /drivers

Create a new driver profile.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{  
  "userId": "uuid",  
  "fleetId": "uuid",  
  "firstName": "Rajesh",  
  "lastName": "Kumar",  
  "mobile": "9876543210",  
  "licenseNumber": "DL-0120230012345",  
  "profilePic": "https://example.com/photo.jpg",  
  "licenseFront": "https://example.com/license-front.jpg",  
  "licenseBack": "https://example.com/license-back.jpg",  
  "aadharFront": "https://example.com/aadhaar-front.jpg",  
  "aadharBack": "https://example.com/aadhaar-back.jpg",  
  "panCardImage": "https://example.com/pan.jpg",  
  "livePhoto": "https://example.com/live.jpg",  
  "bankIdProof": "https://example.com/bank-proof.jpg",  
  "kycStatus": "PENDING",  
  "status": "ACTIVE",  
  "isAvailable": true  
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "userId": "uuid",
    "fleetId": "uuid",
    "firstName": "Rajesh",
    "lastName": "Kumar",
    "mobile": "9876543210",
    "licenseNumber": "DL-0120230012345",
    "kycStatus": "PENDING",
    "status": "ACTIVE",
    "isAvailable": true
  }
}
```

GET `/drivers/fleet/:fleetId`

Get all drivers for a specific fleet.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "firstName": "Rajesh",
      "lastName": "Kumar",
      "mobile": "9876543210",
      "status": "ACTIVE",
      "isAvailable": true
    }
  ]
}
```

GET `/drivers/hub/:hubId`

Get all drivers for a specific hub.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
```

```
{
  "id": "uuid",
  "firstName": "Rajesh",
  "lastName": "Kumar",
  "mobile": "9876543210",
  "hubId": "uuid"
}
```

GET `/drivers/me`

Get current driver's profile.

Authentication: Required

Roles: DRIVER

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "firstName": "Rajesh",
    "lastName": "Kumar",
    "mobile": "9876543210",
    "licenseNumber": "DL-0120230012345",
    "kycStatus": "APPROVED",
    "status": "ACTIVE",
    "isAvailable": true
  }
}
```

GET `/drivers/:id`

Get driver by ID.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "firstName": "Rajesh",
    "lastName": "Kumar",
    "mobile": "9876543210",
    "licenseNumber": "DL-0120230012345",
    "kycStatus": "APPROVED",
    "status": "ACTIVE"
  }
}
```

```
}  
}
```

PATCH `/drivers/:id`

Update driver details.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Request Body:

```
{  
  "firstName": "Rajesh",  
  "lastName": "Kumar",  
  "mobile": "9876543210",  
  "licenseNumber": "DL-0120230012345"  
}
```

Response:

```
{  
  "success": true,  
  "data": {  
    "id": "uuid",  
    "firstName": "Rajesh",  
    "lastName": "Kumar"  
  }  
}
```

PATCH `/drivers/:id/status`

Update driver status.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS`

Request Body:

```
{  
  "status": "ACTIVE"  
}
```

Possible Status Values:

- `ACTIVE`
 - `INACTIVE`
 - `SUSPENDED`
-

PATCH /drivers/:id/availability

Update driver availability.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Request Body:

```
{
  "isAvailable": true
}
```

Fleet Management

All fleet endpoints require authentication.

POST /fleets

Create a new fleet.

Authentication: Required

Roles: SUPER_ADMIN

Request Body:

```
{
  "name": "Demo Fleet Pvt Ltd",
  "city": "GURGAON",
  "mobile": "9876543210",
  "panNumber": "ABCDE1234F",
  "modeId": "CAB",
  "fleetType": "COMPANY",
  "status": "ACTIVE"
}
```

Fleet Types:

- COMPANY
- INDIVIDUAL

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "Demo Fleet Pvt Ltd",
    "city": "GURGAON",
    "mobile": "9876543210",
    "status": "ACTIVE"
  }
}
```

```
}  
}
```

GET /fleets

Get all fleets.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": "uuid",  
      "name": "Demo Fleet Pvt Ltd",  
      "city": "GURGAON",  
      "status": "ACTIVE"  
    }  
  ]  
}
```

GET /fleets/:id

Get fleet by ID.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{  
  "success": true,  
  "data": {  
    "id": "uuid",  
    "name": "Demo Fleet Pvt Ltd",  
    "city": "GURGAON",  
    "mobile": "9876543210",  
    "panNumber": "ABCDE1234F",  
    "status": "ACTIVE"  
  }  
}
```

PATCH /fleets/:id/deactivate

Deactivate a fleet.

Authentication: Required

Roles: SUPER_ADMIN

Response:

```
{
  "success": true,
  "message": "Fleet deactivated successfully"
}
```

POST `/fleets/:id/hubs`

Create a hub for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "name": "Gurgaon Hub",
  "location": "Sector 29, Gurgaon",
  "city": "GURGAON"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "Gurgaon Hub",
    "fleetId": "uuid",
    "location": "Sector 29, Gurgaon",
    "city": "GURGAON"
  }
}
```

GET `/fleets/:id/hubs`

Get all hubs for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": [
    {
```

```
    "id": "uuid",
    "name": "Gurgaon Hub",
    "location": "Sector 29, Gurgaon",
    "city": "GURGAON"
  }
]
```

POST /fleets/:id/hub-managers

Create a hub manager for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "userId": "uuid",
  "name": "Manager Name",
  "phone": "9876543210"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "userId": "uuid",
    "fleetId": "uuid",
    "name": "Manager Name",
    "phone": "9876543210"
  }
}
```

GET /fleets/:id/hub-managers

Get all hub managers for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
```

```
    "name": "Manager Name",
    "phone": "9876543210"
  }
]
```

GET `/fleets/hub-manager/:id`

Get hub manager by ID.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS`

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "name": "Manager Name",
    "phone": "9876543210",
    "fleetId": "uuid"
  }
}
```

POST `/fleets/hubs/:hubId/assign-manager`

Assign a manager to a hub.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS`

Request Body:

```
{
  "managerId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Manager assigned to hub successfully"
}
```

POST `/fleets/hubs/:id/add-vehicle`

Add a vehicle to a hub.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "vehicleId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Vehicle added to hub successfully"
}
```

POST /fleets/hubs/:id/add-driver

Add a driver to a hub.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "driverId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Driver added to hub successfully"
}
```

POST /fleets/hubs/:id/remove-vehicle

Remove a vehicle from a hub.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "vehicleId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Vehicle removed from hub successfully"
}
```

POST `/fleets/hubs/:id/remove-driver`

Remove a driver from a hub.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS`

Request Body:

```
{
  "driverId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Driver removed from hub successfully"
}
```

Fleet Managers

All fleet manager endpoints require authentication.

POST `/fleet-managers`

Create a fleet manager.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS`

Request Body:

```
{
  "userId": "uuid",
  "fleetId": "uuid",
  "name": "Manager Name",
  "phone": "9876543210"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "userId": "uuid",
    "fleetId": "uuid",
    "name": "Manager Name",
    "phone": "9876543210"
  }
}
```

GET `/fleet-managers/fleet/:fleetId`

Get all fleet managers for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "name": "Manager Name",
      "phone": "9876543210"
    }
  ]
}
```

PATCH `/fleet-managers/:id/deactivate`

Deactivate a fleet manager.

Authentication: Required

Roles: SUPER_ADMIN

Response:

```
{
  "success": true,
  "message": "Fleet manager deactivated successfully"
}
```

GET `/drivers/:id/preference`

Get current driver preferences.

Authentication: Required **Roles:** DRIVER

Response:

```
{
  "success": true,
  "data": {
    "preferences": {
      "smokingAllowed": false,
      "musicPreference": "Rock",
      "acPreferred": true
    }
  }
}
```

POST `/drivers/:id/preference/update`

Request a change in driver preferences.

Authentication: Required **Roles:** DRIVER

Request Body:

```
{
  "requestedPreference": {
    "smokingAllowed": true
  }
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "status": "PENDING",
    "requestedPreference": { "smokingAllowed": true }
  },
}
```

```
    "message": "Preference change request submitted successfully"
  }
}
```

GET /drivers/preference/pending-requests

Get all pending preference change requests.

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "driverId": "uuid",
      "driverName": "Rajesh Kumar",
      "currentPreference": { "smokingAllowed": false },
      "requestedPreference": { "smokingAllowed": true },
      "status": "PENDING",
      "requestAt": "2026-01-08T10:00:00Z"
    }
  ]
}
```

POST /drivers/preference/update-status

Approve or reject a preference change request.

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "requestId": "uuid",
  "status": "APPROVED",
  "rejectionReason": "Optional reason if rejected"
}
```

Response:

```
{
  "success": true,
  "message": "Preference request status updated successfully"
}
```

Vehicles

All vehicle endpoints require authentication.

POST /vehicles

Create a new vehicle.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "fleetId": "uuid",
  "vehicleNumber": "DL01AB1234",
  "vehicleName": "Tata Tigor EV",
  "vehicleModel": "Tigor EV 2024",
  "vehicleColor": "White",
  "fuelType": "ELECTRIC",
  "ownership": "OWNED",
  "status": "ACTIVE",
  "rcFrontImage": "https://example.com/rc-front.jpg",
  "rcBackImage": "https://example.com/rc-back.jpg",
  "permitImage": "https://example.com/permit.jpg",
  "permitExpiry": "2034-12-31",
  "fitnessImage": "https://example.com/fitness.jpg",
  "fitnessExpiry": "2026-12-31",
  "insuranceImage": "https://example.com/insurance.jpg",
  "insuranceExpiry": "2025-12-31"
}
```

Fuel Types:

- PETROL
- DIESEL
- CNG
- ELECTRIC

Ownership Types:

- OWNED
- LEASED

Vehicle Status:

- ACTIVE
- INACTIVE
- MAINTENANCE

Response:

```
{
  "success": true,
  "data": {
```

```
    "id": "uuid",
    "vehicleNumber": "DL01AB1234",
    "vehicleName": "Tata Tigor EV",
    "status": "ACTIVE"
  }
}
```

GET `/vehicles/fleet/:fleetId`

Get all vehicles for a fleet.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "vehicleNumber": "DL01AB1234",
      "vehicleName": "Tata Tigor EV",
      "status": "ACTIVE"
    }
  ]
}
```

GET `/vehicles/hub/:hubId`

Get all vehicles for a hub.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "vehicleNumber": "DL01AB1234",
      "vehicleName": "Tata Tigor EV",
      "hubId": "uuid"
    }
  ]
}
```

GET `/vehicles/:id`

Get vehicle by ID.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "vehicleNumber": "DL01AB1234",
    "vehicleName": "Tata Tigor EV",
    "vehicleModel": "Tigor EV 2024",
    "fuelType": "ELECTRIC",
    "status": "ACTIVE"
  }
}
```

PATCH /vehicles/:id

Update vehicle details.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "vehicleName": "Tata Tigor EV",
  "vehicleModel": "Tigor EV 2024",
  "vehicleColor": "White"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "vehicleName": "Tata Tigor EV"
  }
}
```

GET /drivers/:id/preference

Get driver preferences.

Authentication: Required

Roles: DRIVER

Response:

```
{
  "success": true,
  "statusCode": 200,
  "message": "Driver preferences retrieved successfully",
  "data": [
    {
      "key": "prefer_airport_rides",
      "displayName": "Prefer airport rides",
      "description": "Prioritize airport pickup and drop trips",
      "category": "TRIP",
      "approvalRequired": true,
      "value": true
    }
  ]
}
```

POST `/drivers/:id/preference/update`

Request preference change.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "prefer_airport_rides": true,
  "accept_rentals": true,
  "auto_assign_rides": true
}
```

Response:

```
{
  "success": true,
  "message": "Preference change request submitted successfully"
}
```

GET `/drivers/preference/pending-requests`

Get pending preference requests.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "statusCode": 200,
  "message": "Pending preference requests retrieved successfully",
  "data": [
    {
      "id": "bd3c2df9-d58d-4b5a-8d20-2ecd8db1b63e",
      "driverId": "ad8324ca-2dea-4618-ba5e-3095fa123d06",
      "currentPreference": {
        "accept_rentals": false,
        "prefer_airport_rides": false
      },
      "requestedPreference": {
        "accept_rentals": true,
        "prefer_airport_rides": true
      },
      "status": "PENDING",
      "requestAt": "2026-01-08T04:38:38.415Z"
    }
  ]
}
```

POST /drivers/preference/update-status

Update preference request status.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body (Approve):

```
{
  "id": "bd3c2df9-d58d-4b5a-8d20-2ecd8db1b63e",
  "status": "APPROVED"
}
```

Request Body (Reject):

```
{
  "id": "bd3c2df9-d58d-4b5a-8d20-2ecd8db1b63e",
  "status": "REJECTED",
  "rejection_reason": "demo test"
}
```

Response:

```
{
  "success": true,
```

```
"statusCode": 200,
"message": "Preference change request updated successfully",
"data": {
  "id": "bd3c2df9-d58d-4b5a-8d20-2ecd8db1b63e",
  "status": "REJECTED",
  "rejectionReason": "demo test"
}
```

PATCH /vehicles/:id/docs

Update vehicle documents.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "rcFrontImage": "https://example.com/rc-front.jpg",
  "rcBackImage": "https://example.com/rc-back.jpg",
  "permitImage": "https://example.com/permit.jpg",
  "permitExpiry": "2034-12-31"
}
```

Response:

```
{
  "success": true,
  "message": "Vehicle documents updated successfully"
}
```

PATCH /vehicles/:id/status

Update vehicle status.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "status": "ACTIVE"
}
```

PATCH /vehicles/:id/deactivate

Deactivate a vehicle.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "message": "Vehicle deactivated successfully"
}
```

PATCH_DOCS /vehicles/:id/docs

Update vehicle documents.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "rcFrontImage": "https://example.com/rc-front.jpg",
  "rcBackImage": "https://example.com/rc-back.jpg",
  "permitImage": "https://example.com/permit.jpg",
  "permitExpiry": "2034-12-31",
  "fitnessImage": "https://example.com/fitness.jpg",
  "fitnessExpiry": "2026-12-31",
  "insuranceImage": "https://example.com/insurance.jpg",
  "insuranceExpiry": "2025-12-31"
}
```

Response:

```
{
  "success": true,
  "message": "Vehicle documents updated successfully"
}
```

PATCH_STATUS /vehicles/:id/status

Update vehicle status.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "status": "ACTIVE"
}
```

```
}
```

Status Values:

- ACTIVE
- INACTIVE
- MAINTENANCE

Response:

```
{
  "success": true,
  "message": "Vehicle status updated successfully"
}
```

PATCH_DEACTIVATE /vehicles/:id/deactivate

Deactivate a vehicle.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "message": "Vehicle deactivated successfully"
}
```

Assignments

All assignment endpoints require authentication.

POST /assignments

Create a driver-vehicle assignment.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Request Body:

```
{
  "fleetId": "uuid",
  "driverId": "uuid",
  "vehicleId": "uuid",
  "status": "ACTIVE"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "fleetId": "uuid",
    "driverId": "uuid",
    "vehicleId": "uuid",
    "status": "ACTIVE"
  }
}
```

GET `/assignments/fleet/:fleetId`

Get all assignments for a fleet.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "driver": {
        "id": "uuid",
        "firstName": "Rajesh",
        "lastName": "Kumar"
      },
      "vehicle": {
        "id": "uuid",
        "vehicleNumber": "DL01AB1234"
      },
      "status": "ACTIVE"
    }
  ]
}
```

GET `/assignments/trip/:tripId`

Get assignments for a specific trip.

Authentication: Required

Roles: `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "tripId": "uuid",
      "driverId": "uuid",
      "vehicleId": "uuid"
    }
  ]
}
```

PATCH /assignments/:id/end

End an assignment.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Response:

```
{
  "success": true,
  "message": "Assignment ended successfully"
}
```

Attendance

All attendance endpoints require authentication.

POST /attendance/check-in

Driver check-in.

Authentication: Required

Roles: Any authenticated user

Request Body:

```
{
  "driverId": "uuid",
  "location": {
    "latitude": 28.4595,
    "longitude": 77.0266
  },
  "timestamp": "2025-12-26T10:00:00.000Z"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "driverId": "uuid",
    "checkInTime": "2025-12-26T10:00:00.000Z",
    "status": "PENDING"
  }
}
```

POST /attendance/check-out

Driver check-out.

Authentication: Required

Roles: Any authenticated user

Request Body:

```
{
  "attendanceId": "uuid",
  "location": {
    "latitude": 28.4595,
    "longitude": 77.0266
  },
  "odometer": 10500,
  "cashDeposited": 500,
  "timestamp": "2025-12-26T18:00:00.000Z"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "checkOutTime": "2025-12-26T18:00:00.000Z"
  }
}
```

POST /attendance/start-break

Start a break during active attendance.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "attendanceId": "uuid"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "attendanceId": "uuid",
    "startTime": "2025-12-26T12:00:00.000Z"
  }
}
```

POST /attendance/end-break

End current break.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "breakId": "uuid"
}
```

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "endTime": "2025-12-26T12:30:00.000Z"
  }
}
```

GET /attendance/history

Get attendance history.

Authentication: Required

Roles: Any authenticated user

Query Parameters:

- `driverId` (optional) - Filter by driver
- `startDate` (optional) - Start date
- `endDate` (optional) - End date

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "driverId": "uuid",
      "checkInTime": "2025-12-26T10:00:00.000Z",
      "checkOutTime": "2025-12-26T18:00:00.000Z",
      "status": "APPROVED"
    }
  ]
}
```

POST `/attendance/:id/approve`

Approve attendance record.

Authentication: Required

Roles: `SUPER_ADMIN` , `MANAGER`

Response:

```
{
  "success": true,
  "message": "Attendance approved successfully"
}
```

POST `/attendance/:id/reject`

Reject attendance record.

Authentication: Required

Roles: `SUPER_ADMIN` , `MANAGER`

Request Body:

```
{
  "reason": "Invalid check-in location"
}
```

Response:

```
{
  "success": true,
  "message": "Attendance rejected successfully"
}
```

Trips

All trip endpoints require authentication.

POST /trips

Create a new trip.

Authentication: Required

Request Body:

```
{
  "pickupLocation": {
    "address": "Sector 29, Gurgaon",
    "latitude": 28.4595,
    "longitude": 77.0266
  },
  "dropLocation": {
    "address": "IGI Airport, Delhi",
    "latitude": 28.5562,
    "longitude": 77.1000
  },
  "pickupTime": "2025-12-26T14:00:00.000Z",
  "passengerName": "John Doe",
  "passengerPhone": "9876543210",
  "vehicleType": "SEDAN",
  "bookingType": "PREBOOK"
}
```

Booking Types:

- PREBOOK - Scheduled in advance
- INSTANT - Immediate booking

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "pickupLocation": {
      "address": "Sector 29, Gurgaon",
      "latitude": 28.4595,
      "longitude": 77.0266
    },
  },
}
```

```
    "dropLocation": {
      "address": "IGI Airport, Delhi",
      "latitude": 28.5562,
      "longitude": 77.1000
    },
    "pickupTime": "2025-12-26T14:00:00.000Z",
    "status": "PENDING",
    "estimatedFare": 625,
    "distance": 25.5
  }
}
```

GET /trips

Get trips for current driver.

Authentication: Required

Roles: DRIVER

Query Parameters:

- `status` (optional) - Filter by status
- `date` (optional) - Filter by date

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "pickupLocation": {
        "address": "Sector 29, Gurgaon"
      },
      "dropLocation": {
        "address": "IGI Airport, Delhi"
      },
      "pickupTime": "2025-12-26T14:00:00.000Z",
      "status": "DRIVER_ASSIGNED",
      "estimatedFare": 625
    }
  ]
}
```

GET /trips/:id

Get trip details by ID.

Authentication: Required

Response:

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "pickupLocation": {
      "address": "Sector 29, Gurgaon",
      "latitude": 28.4595,
      "longitude": 77.0266
    },
    "dropLocation": {
      "address": "IGI Airport, Delhi",
      "latitude": 28.5562,
      "longitude": 77.1000
    },
    "pickupTime": "2025-12-26T14:00:00.000Z",
    "status": "DRIVER_ASSIGNED",
    "driver": {
      "id": "uuid",
      "firstName": "Rajesh",
      "lastName": "Kumar",
      "mobile": "9876543210"
    },
    "vehicle": {
      "id": "uuid",
      "vehicleNumber": "DL01AB1234"
    },
    "estimatedFare": 625,
    "distance": 25.5
  }
}
```

POST /trips/:id/assign

Assign driver to trip.

Authentication: Required

Request Body:

```
{
  "driverId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Driver assigned successfully",
  "data": {
    "tripId": "uuid",
```

```
    "driverId": "uuid",
    "status": "DRIVER_ASSIGNED"
  }
}
```

POST /trips/:id/start

Start a trip (driver en route to pickup).

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "location": {
    "latitude": 28.4595,
    "longitude": 77.0266
  },
  "timestamp": "2025-12-26T13:30:00.000Z"
}
```

Constraints:

- Must be within 2.5 hours of pickup time
- Trip must be in DRIVER_ASSIGNED status

Response:

```
{
  "success": true,
  "message": "Trip started successfully",
  "data": {
    "tripId": "uuid",
    "status": "STARTED"
  }
}
```

POST /trips/:id/arrived

Mark driver as arrived at pickup location.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "location": {
    "latitude": 28.4595,
```

```
    "longitude": 77.0266
  },
  "timestamp": "2025-12-26T13:55:00.000Z"
}
```

Constraints:

- Must be within 30 minutes of pickup time
- Must be within 500m radius of pickup location
- Trip must be in `STARTED` status

Response:

```
{
  "success": true,
  "message": "Arrival confirmed",
  "data": {
    "tripId": "uuid",
    "status": "ARRIVED"
  }
}
```

POST `/trips/:id/onboard`

Mark passenger as onboarded.

Authentication: Required

Roles: `DRIVER`

Request Body:

```
{
  "otp": "1234",
  "timestamp": "2025-12-26T14:00:00.000Z"
}
```

Response:

```
{
  "success": true,
  "message": "Passenger onboarded successfully",
  "data": {
    "tripId": "uuid",
    "status": "ONBOARDED"
  }
}
```

POST `/trips/:id/noshow`

Mark trip as no-show.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "reason": "Passenger did not arrive",
  "timestamp": "2025-12-26T14:30:00.000Z"
}
```

Constraints:

- Can only be marked after 30 minutes past pickup time

Response:

```
{
  "success": true,
  "message": "Trip marked as no-show",
  "data": {
    "tripId": "uuid",
    "status": "NO_SHOW"
  }
}
```

POST /trips/:id/complete

Complete a trip.

Authentication: Required

Roles: DRIVER

Request Body:

```
{
  "location": {
    "latitude": 28.5562,
    "longitude": 77.1000
  },
  "actualDistance": 26.2,
  "timestamp": "2025-12-26T15:00:00.000Z"
}
```

Response:

```
{
  "success": true,
```

```
"message": "Trip completed successfully",
"data": {
  "tripId": "uuid",
  "status": "COMPLETED",
  "actualDistance": 26.2,
  "finalFare": 650
}
```

GET `/trips/:id/tracking`

Get real-time trip tracking information.

Authentication: Required

Response:

```
{
  "success": true,
  "data": {
    "tripId": "uuid",
    "status": "ONBOARDED",
    "driver": {
      "name": "Rajesh Kumar",
      "phone": "9876543210",
      "currentLocation": {
        "latitude": 28.5000,
        "longitude": 77.0500
      }
    },
    "vehicle": {
      "number": "DL01AB1234",
      "model": "Tata Tigor EV"
    },
    "eta": "2025-12-26T15:00:00.000Z"
  }
}
```

Admin Trips

All admin trip endpoints require authentication and `SUPER_ADMIN` role.

GET `/admin/trips`

Get all trips (admin view).

Authentication: Required

Roles: `SUPER_ADMIN`

Query Parameters:

- `status` (optional) - Filter by status
- `date` (optional) - Filter by date

- `driverId` (optional) - Filter by driver
- `fleetId` (optional) - Filter by fleet

Response:

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "pickupLocation": {
        "address": "Sector 29, Gurgaon"
      },
      "dropLocation": {
        "address": "IGI Airport, Delhi"
      },
      "status": "COMPLETED",
      "driver": {
        "id": "uuid",
        "firstName": "Rajesh",
        "lastName": "Kumar"
      },
      "estimatedFare": 625,
      "finalFare": 650
    }
  ]
}
```

POST `/admin/trips/assign`

Manually assign driver to trip (admin).

Authentication: Required

Roles: `SUPER_ADMIN`

Request Body:

```
{
  "tripId": "uuid",
  "driverId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Driver assigned successfully"
}
```

POST /admin/trips/unassign

Unassign driver from trip.

Authentication: Required

Roles: SUPER_ADMIN

Request Body:

```
{
  "tripId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Driver unassigned successfully"
}
```

POST /admin/trips/reassign

Reassign trip to different driver.

Authentication: Required

Roles: SUPER_ADMIN

Request Body:

```
{
  "tripId": "uuid",
  "newDriverId": "uuid"
}
```

Response:

```
{
  "success": true,
  "message": "Trip reassigned successfully"
}
```

Pricing

POST /pricing/preview

Get pricing preview for a trip.

Authentication: None

Request Body:

```
{
  "pickupLocation": {
    "latitude": 28.4595,
    "longitude": 77.0266
  },
  "dropLocation": {
    "latitude": 28.5562,
    "longitude": 77.1000
  },
  "pickupTime": "2025-12-26T14:00:00.000Z",
  "vehicleType": "SEDAN"
}
```

Response:

```
{
  "success": true,
  "data": {
    "estimatedFare": 625,
    "distance": 25.5,
    "duration": 45,
    "breakdown": {
      "baseFare": 100,
      "distanceFare": 450,
      "timeFare": 75,
      "taxes": 0,
      "total": 625
    }
  }
}
```

Partner Integrations

MakeMyTrip (MMT)

Partner integration endpoints for MakeMyTrip.

POST `/partner/mmt/partnersearchendpoint`

Search for available vehicles (MMT → Driver's Klub).

Authentication: Basic Auth (Username/Password)

Request Body:

```
{
  "pickupLocation": "Sector 29, Gurgaon",
  "dropLocation": "IGI Airport, Delhi",
  "pickupTime": "2025-12-26T14:00:00.000Z",
  "vehicleType": "SEDAN"
}
```

Response:

```
{
  "success": true,
  "data": {
    "available": true,
    "vehicles": [
      {
        "type": "SEDAN",
        "fare": 625,
        "eta": 10
      }
    ]
  }
}
```

POST `/partner/mmt/partnerblockendpoint`

Block a vehicle for booking (MMT → Driver's Klub).

Authentication: Basic Auth (Username/Password)

Request Body:

```
{
  "bookingId": "MMT123456",
  "vehicleType": "SEDAN",
  "pickupLocation": "Sector 29, Gurgaon",
  "dropLocation": "IGI Airport, Delhi",
  "pickupTime": "2025-12-26T14:00:00.000Z"
}
```

Response:

```
{
  "success": true,
  "data": {
    "blockId": "uuid",
    "expiresAt": "2025-12-26T13:50:00.000Z"
  }
}
```

```
}  
}
```

POST /partner/mmt/partnerpaidendpoint

Confirm booking payment (MMT → Driver's Klub).

Authentication: Basic Auth (Username/Password)

Request Body:

```
{  
  "bookingId": "MMT123456",  
  "blockId": "uuid",  
  "paymentStatus": "PAID"  
}
```

Response:

```
{  
  "success": true,  
  "data": {  
    "tripId": "uuid",  
    "status": "CONFIRMED"  
  }  
}
```

POST /partner/mmt/partnercancelendpoint

Cancel a booking (MMT → Driver's Klub).

Authentication: None (Partner API)

Request Body:

```
{  
  "bookingId": "MMT123456",  
  "reason": "Customer cancellation"  
}
```

Response:

```
{  
  "success": true,  
  "message": "Booking cancelled successfully"  
}
```

GET `/partner/mmt/booking/details`

Get booking details (MMT → Driver's Klub).

Authentication: None (Partner API)

Query Parameters:

- `bookingId` - MMT booking ID

Response:

```
{
  "success": true,
  "data": {
    "bookingId": "MMT123456",
    "tripId": "uuid",
    "status": "CONFIRMED",
    "driver": {
      "name": "Rajesh Kumar",
      "phone": "9876543210"
    },
    "vehicle": {
      "number": "DL01AB1234",
      "model": "Tata Tigor EV"
    }
  }
}
```

POST `/partner/mmt/partnerrescheduleblockendpoint`

Validate logic for reschedule request (MMT → Driver's Klub).

Authentication: Basic Auth (Username/Password)

Request Body:

```
{
  "order_reference_number": "MMT123456",
  "start_time": "2025-12-31T10:00:00Z"
}
```

Response:

```
{
  "response": {
    "success": true,
    "verification_code": "1234",
    "fare_details": {
      "total_amount": 1000,
      "payable_amount": 1000
    }
  },
}
```

```
    "driver_details": {
      "name": "Rajesh",
      "phone": "9876543210"
    }
  }
}
```

POST `/partner/mmt/partnerrescheduleconfirmendpoint`

Confirm reschedule request (MMT → Driver's Klub).

Authentication: Basic Auth (Username/Password)

Request Body:

```
{
  "order_reference_number": "MMT123456"
}
```

Response:

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

POST `/trips/:id/location`

Update driver live location for a trip.

Authentication: Required **Roles:** DRIVER

Request Body:

```
{
  "lat": 28.5355,
  "lng": 77.3910
}
```

Response:

```
{
  "success": true,
  "message": "Location updated successfully"
}
```

Webhooks

Webhooks allow real-time notifications for system events, primarily payment updates.

Easebuzz Payment Webhook

POST `/webhooks/easebuzz`

Webhook endpoint for Easebuzz payment gateway notifications.

Authentication: Signature Verification (Salt-based)

Request Body:

- Standard Easebuzz webhook payload (FormData) including `txnid`, `status`, `amount`, `hash`, etc.

Response:

```
{
  "success": true,
  "message": "Webhook processed successfully"
}
```

Error Handling

All API errors follow a consistent format:

```
{
  "success": false,
  "message": "Error message",
  "error": {
    "code": "ERROR_CODE",
    "details": "Detailed error information"
  }
}
```

Common HTTP Status Codes

- 200 - Success
- 201 - Created
- 400 - Bad Request (validation errors)
- 401 - Unauthorized (missing or invalid token)
- 403 - Forbidden (insufficient permissions)
- 404 - Not Found
- 429 - Too Many Requests (rate limit exceeded)
- 500 - Internal Server Error

- `503` - Service Unavailable (database connection issues)

Common Error Codes

- `INVALID_CREDENTIALS` - Invalid phone or OTP
 - `OTP_EXPIRED` - OTP has expired
 - `OTP_ATTEMPT_EXCEEDED` - Too many OTP attempts
 - `UNAUTHORIZED` - Missing or invalid authentication token
 - `FORBIDDEN` - Insufficient permissions
 - `NOT_FOUND` - Resource not found
 - `VALIDATION_ERROR` - Request validation failed
 - `TRIP_CONSTRAINT_VIOLATION` - Trip timing/location constraints not met
-

Rate Limiting

The API implements rate limiting to prevent abuse:

- **Window:** 15 minutes
- **Limit:** 100 requests per IP
- **Headers:**
 - `X-RateLimit-Limit` - Maximum requests allowed
 - `X-RateLimit-Remaining` - Remaining requests
 - `X-RateLimit-Reset` - Time when limit resets

When rate limit is exceeded:

```
{
  "success": false,
  "message": "Too many requests from this IP, please try again later."
}
```

Additional Notes

CORS Configuration

- **Development:** All origins allowed (*)
- **Production:** Specific origins from `ALLOWED_ORIGINS` environment variable

Timezone

All timestamps are in **IST (Indian Standard Time)** and follow ISO 8601 format.

Pagination

Endpoints returning lists support pagination:

Query Parameters:

- `page` - Page number (default: 1)
- `limit` - Items per page (default: 20, max: 100)

Response:

```
{
  "success": true,
  "data": [...],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 150,
    "totalPages": 8
  }
}
```

Support

For API support and questions:

- **Email:** gourav.singh@triborefin.com
- **Documentation:** `/api-docs` (Coming Soon)
- **Health Check:** `/health`

Payment System

Complete payment and payout system with Easebuzz integration.

Base Path: `/payment`

Documentation: See [Payment System Documentation](#) for complete details.

Driver Endpoints

GET `/payment/balance`

Get driver balance & rental status.

Authentication: Required

Roles: `DRIVER`

Response:

```
{
  "depositBalance": 5000,
  "paymentModel": "RENTAL",
  "hasActiveRental": true,
  "rental": {
    "planName": "Weekly Plan",
    "startDate": "2025-12-23T00:00:00.000Z",
    "expiryDate": "2025-12-30T00:00:00.000Z",
    "daysRemaining": 3,
    "isExpired": false
  }
}
```

```
}  
}
```

GET /payment/rental/plans

Get available rental plans for the driver's fleet.

Authentication: Required

Roles: DRIVER

Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": "uuid",  
      "name": "Weekly Plan",  
      "rentalAmount": 3000,  
      "depositAmount": 5000,  
      "validityDays": 7,  
      "isActive": true  
    }  
  ]  
}
```

GET /payment/transactions

Get driver transaction history (paginated).

Authentication: Required

Roles: DRIVER

Query Parameters:

- `page` (number, default: 1)
- `limit` (number, default: 20)
- `type` (string, optional): DEPOSIT, RENTAL, TRIP_PAYMENT, INCENTIVE, PENALTY, PAYOUT
- `status` (string, optional): PENDING, SUCCESS, FAILED
- `startDate` (string, optional): ISO date
- `endDate` (string, optional): ISO date

Response:

```
{  
  "transactions": [  
    {  
      "id": "uuid",  
      "type": "DEPOSIT",  
      "amount": 5000,  
      "status": "SUCCESS",  
      "paymentMethod": "PG_UPI",  
      "description": "Security deposit - ₹5000",  
    }  
  ]  
}
```

```
      "createdAt": "2025-12-23T10:00:00.000Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 45,
    "totalPages": 3
  }
}
```

GET `/payment/incentives`

Get driver incentives.

Authentication: Required

Roles: DRIVER

Query Parameters:

- `isPaid` (boolean, optional)

Response:

```
{
  "incentives": [
    {
      "id": "uuid",
      "amount": 500,
      "reason": "Completed 50 trips",
      "category": "MILESTONE",
      "isPaid": false,
      "createdAt": "2025-12-25T00:00:00.000Z"
    }
  ],
  "summary": {
    "totalIncentives": 10,
    "paidIncentives": 7,
    "unpaidIncentives": 3,
    "totalAmount": 5000,
    "paidAmount": 3500,
    "unpaidAmount": 1500
  }
}
```

GET `/payment/penalties`

Get driver penalties.

Authentication: Required

Roles: DRIVER

Query Parameters:

- `type` (string, optional): MONETARY, WARNING, SUSPENSION, BLACKLIST

- `isPaid` (boolean, optional)
- `isWaived` (boolean, optional)

Response:

```
{
  "penalties": [
    {
      "id": "uuid",
      "type": "MONETARY",
      "amount": 200,
      "reason": "Late for pickup",
      "isPaid": true,
      "isWaived": false,
      "deductedFromDeposit": true,
      "depositDeductionAmount": 200,
      "createdAt": "2025-12-24T00:00:00.000Z"
    }
  ]
}
```

GET `/payment/collections`

Get daily collections (for payout model drivers).

Authentication: Required

Roles: DRIVER

Query Parameters:

- `startDate` (string, optional): ISO date
- `endDate` (string, optional): ISO date
- `isReconciled` (boolean, optional)
- `isPaid` (boolean, optional)

Response:

```
{
  "collections": [
    {
      "id": "uuid",
      "date": "2025-12-29T00:00:00.000Z",
      "qrCollectionAmount": 3000,
      "cashCollectionAmount": 2000,
      "totalCollection": 5000,
      "revShareAmount": 3500,
      "incentiveAmount": 500,
      "penaltyAmount": 200,
      "netPayout": 3800,
      "isReconciled": true,
      "isPaid": false
    }
  ],
  "summary": {
```

```
    "totalDays": 30,  
    "totalCollections": 150000,  
    "totalRevShare": 105000,  
    "totalIncentives": 5000,  
    "totalPenalties": 2000,  
    "totalPayout": 108000,  
    "paidAmount": 75000,  
    "unpaidAmount": 33000  
  }  
}
```

POST `/payment/deposit`

Initiate deposit payment.

Authentication: Required

Roles: DRIVER

Request Body:

```
{  
  "amount": 5000  
}
```

Response:

```
{  
  "transactionId": "uuid",  
  "paymentUrl": "https://testpay.easebuzz.in/pay/...",  
  "txnId": "TXN_1735123456_ABC123"  
}
```

POST `/payment/rental`

Initiate rental payment.

Authentication: Required

Roles: DRIVER

Request Body:

```
{  
  "rentalPlanId": "uuid"  
}
```

Response:

```
{
  "transactionId": "uuid",
  "paymentUrl": "https://testpay.easebuzz.in/pay/...",
  "txnId": "TXN_1735123456_XYZ789"
}
```

Admin Endpoints

POST /payment/admin/rental-plans

Create a rental plan.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "fleetId": "uuid",
  "name": "Weekly Plan",
  "rentalAmount": 3500,
  "depositAmount": 5000,
  "validityDays": 7
}
```

Response:

```
{
  "id": "uuid",
  "fleetId": "uuid",
  "name": "Weekly Plan",
  "rentalAmount": 3500,
  "depositAmount": 5000,
  "validityDays": 7,
  "isActive": true
}
```

GET /payment/admin/rental-plans/:fleetId

Get rental plans for a fleet.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Query Parameters:

- `activeOnly` (boolean, default: true)

POST /payment/admin/penalty

Create a penalty for a driver.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "driverId": "uuid",
  "type": "MONETARY",
  "amount": 500,
  "reason": "Customer complaint",
  "category": "BEHAVIOR"
}
```

Penalty Types:

- MONETARY - Financial penalty (auto-deducted from deposit for rental model)
- WARNING - Verbal/written warning
- SUSPENSION - Temporary suspension (requires suspensionStartDate and suspensionEndDate)
- BLACKLIST - Permanent ban

POST /payment/admin/penalty/:id/waive

Waive a penalty.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "waiverReason": "First-time offense, driver apologized"
}
```

POST /payment/admin/incentive

Create an incentive for a driver.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Request Body:

```
{
  "driverId": "uuid",
  "amount": 500,
  "reason": "Completed 50 trips this month",
  "category": "MILESTONE"
}
```

POST /payment/admin/incentive/:id/payout

Process incentive payout.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "success": true,
  "txnId": "TXN_1735123456_PAY123",
  "status": "PENDING",
  "utr": "UTR123456789"
}
```

POST /payment/admin/collection/:id/reconcile

Reconcile daily collection.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

Request Body:

```
{
  "expectedRevenue": 5000,
  "reconciliationNotes": "All collections verified"
}
```

POST /payment/admin/collection/:id/payout

DEPRECATED: Use Bulk Payout instead.

Process daily payout (Legacy).

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS

POST /payment/admin/bulk-payout

Upload CSV for automated bulk payouts.

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS

Request: multipart/form-data

- file : CSV File (phone, amount or accountNumber, amount)

Response:

```
{
  "total": 10,
  "success": 9,
  "failed": 1,
  "amountDisbursed": 45000,
  "details": [
```

```
{ "phone": "9876543210", "status": "SUCCESS", "amount": 5000, "utr": "UTR..." }
]
}
```

GET `/payment/admin/reconciliations/pending`

Get pending reconciliations.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

GET `/payment/admin/payouts/pending`

Get pending payouts.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

POST `/payment/admin/vehicle/:id/qr`

Generate virtual QR code for a vehicle.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS

Response:

```
{
  "id": "uuid",
  "vehicleId": "uuid",
  "virtualAccountId": "VA123456",
  "virtualAccountNumber": "1234567890123456",
  "ifscCode": "HDFC0000001",
  "qrCodeBase64": "data:image/png;base64,...",
  "upiId": "driversklub.va123456@easebuzz",
  "isActive": true
}
```

GET `/payment/admin/vehicle/:id/qr`

Get vehicle QR code.

Authentication: Required

Roles: SUPER_ADMIN , OPERATIONS , MANAGER

InstaCollect Orders (Dynamic QR)

These endpoints manage ad-hoc payment orders where a customer can pay via a dynamic QR code, supporting partial payments.

POST `/payment/orders`

Create a new payment order. This generates a unique virtual account and QR code.

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS , MANAGER

Request Body:

```
{
  "customerName": "John Doe",
  "customerPhone": "9876543210",
  "customerEmail": "john@example.com",
  "description": "Invoice #1234",
  "amount": 5000
}
```

Response:

```
{
  "id": "uuid",
  "totalAmount": 5000,
  "collectedAmount": 0,
  "remainingAmount": 5000,
  "status": "PENDING",
  "virtualAccountId": "VA123456",
  "virtualAccountNumber": "1234567890",
  "ifscCode": "HDFC0000001",
  "qrCodeBase64": "data:image/png;base64,...",
  "upiId": "driversklub.va123456@easebuzz",
  "paymentLink": "https://testpay.easebuzz.in/pay/..."
}
```

GET `/payment/orders`

List payment orders.

Authentication: Required **Roles:** `SUPER_ADMIN` , `OPERATIONS` , `MANAGER`

Query Parameters:

- `status` (string, optional): `PENDING` , `PARTIAL` , `COMPLETED` , `FAILED`
- `search` (string, optional): Search by customer name or phone

Response:

```
{
  "orders": [
    {
      "id": "uuid",
      "customerName": "John Doe",
      "totalAmount": 5000,
      "collectedAmount": 2000,
      "remainingAmount": 3000,
      "status": "PARTIAL",
      "createdAt": "2025-12-30T10:00:00Z"
    }
  ]
}
```

```
]
}
```

GET /payment/orders/:id

Get detailed order information including transaction history.

Authentication: Required **Roles:** SUPER_ADMIN , OPERATIONS , MANAGER

Response:

```
{
  "id": "uuid",
  "customerName": "John Doe",
  "totalAmount": 5000,
  "collectedAmount": 2000,
  "remainingAmount": 3000,
  "status": "PARTIAL",
  "transactions": [
    {
      "id": "tx_1",
      "amount": 2000,
      "status": "SUCCESS",
      "createdAt": "2025-12-30T10:05:00Z",
      "easebuzzTxnId": "TXN123"
    }
  ]
}
```

Rapido (Fleet)

Webhook callbacks from Rapido Fleet API.

POST /rapido/webhook/order-status

Callback for Rapido order status updates.

Authentication: None (Public/Webhook)

Request Body:

```
{
  "orderId": "rapido_order_123",
  "status": "started",
  "captainId": "cap_123",
  "timestamp": "2025-12-30T10:00:00Z"
}
```

Response:

```
{
  "success": true,
  "message": "Order status updated"
}
```

POST /rapido/webhook/captain-status

Callback for Captain login/availability status.

Authentication: None (Public/Webhook)

Request Body:

```
{
  "captainId": "cap_123",
  "status": "online",
  "location": { "lat": 28.5, "lng": 77.1 }
}
```

Response:

```
{
  "success": true,
  "message": "Captain status updated"
}
```
