**# EasyTrip Bus Booking System API Documentation**

This document provides a comprehensive guide to the backend API for the EasyTrip Bus Ticket Booking platform.

**## Base URL**

All API endpoints are prefixed with `/api`.

- **\*\*Development\*\***: `http://localhost:5000/api`

- **\*\*Production\*\***: `https://your-domain.com/api`

**## Authentication**

The system supports JWT-based authentication with HTTP-only cookies for enhanced security.

**### Authentication Headers**

For protected routes that require authentication:

```javascript

headers: {

  "Authorization": "Bearer YOUR\_JWT\_TOKEN",

  "Content-Type": "application/json"

}

```

**\*\*Note\*\***: JWT tokens are automatically managed via HTTP-only cookies for enhanced security.

---

**## 1. System Health (/api/health)**

**### 1.1 Health Check**

- **\*\*Endpoint\*\***: `GET /api/health`

- **\*\*Description\*\***: Checks if the server is running and responsive

- **\*\*Access\*\***: Public

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "ok",

  "message": "Server is running",

  "timestamp": "2025-08-01T10:00:00.000Z"

}

```

---

**## 2. Authentication (/api/auth)**

**\*\*Note\*\***: Authentication routes are available but not currently active in server.js. These endpoints provide complete JWT implementation.

**### 2.1 User Login with OTP**

- **\*\*Endpoint\*\***: `POST /api/auth/login`

- **\*\*Description\*\***: Authenticates user with phone number and OTP, returns JWT in HTTP-only cookie

- **\*\*Access\*\***: Public

- **\*\*Request Body\*\***:

```json

{

  "phone": "9876543210",

  "otp": "123456"

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "message": "Login successful",

  "user": {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "role": "user"

  }

}

```

- **\*\*Error Responses\*\***:

  - `400 Bad Request`: Missing phone or OTP

  - `404 Not Found`: User not registered

  - `500 Server Error`: Internal server error

**### 2.2 Admin Login**

- **\*\*Endpoint\*\***: `POST /api/auth/admin-login`

- **\*\*Description\*\***: Admin authentication with email and password

- **\*\*Access\*\***: Public

- **\*\*Request Body\*\***:

```json

{

  "email": "admin@easytrip.com",

  "password": "admin123"

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "message": "Admin login successful",

  "user": {

    "id": "ADMIN001",

    "name": "Admin User",

    "email": "admin@easytrip.com",

    "role": "admin"

  }

}

```

**### 2.3 User Registration**

- **\*\*Endpoint\*\***: `POST /api/auth/register`

- **\*\*Description\*\***: Creates a new user account

- **\*\*Access\*\***: Public

- **\*\*Request Body\*\***:

```json

{

  "name": "John Doe",

  "email": "john.doe@example.com",

  "phone": "9876543210",

  "gender": "male",

  "address": {

    "street": "123 Main St",

    "city": "Mumbai",

    "state": "Maharashtra",

    "pincode": "400001"

  },

  "emergencyContact": {

    "name": "Jane Doe",

    "phone": "9876543211",

    "relation": "spouse"

  }

}

```

- **\*\*Success Response (201)\*\***:

```json

{

  "status": "success",

  "message": "User registered successfully",

  "user": {

    "id": "U1235",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "role": "user"

  }

}

```

**### 2.4 Get User Profile**

- **\*\*Endpoint\*\***: `GET /api/auth/profile`

- **\*\*Description\*\***: Retrieves current authenticated user's profile

- **\*\*Access\*\***: Private (Requires JWT)

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "user": {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "role": "user",

    "address": { ... },

    "emergencyContact": { ... }

  }

}

```

**### 2.5 Refresh JWT Token**

- **\*\*Endpoint\*\***: `POST /api/auth/refresh`

- **\*\*Description\*\***: Refreshes expired JWT token

- **\*\*Access\*\***: Private (Requires valid token)

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "message": "Token refreshed successfully"

}

```

**### 2.6 Verify Token**

- **\*\*Endpoint\*\***: `GET /api/auth/verify`

- **\*\*Description\*\***: Verifies if current JWT token is valid

- **\*\*Access\*\***: Private (Requires JWT)

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "message": "Token is valid",

  "user": { ... }

}

```

**### 2.7 Check Authentication Status**

- **\*\*Endpoint\*\***: `GET /api/auth/check`

- **\*\*Description\*\***: Checks authentication status (optional auth)

- **\*\*Access\*\***: Public (but adds user info if authenticated)

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "isAuthenticated": true,

  "user": {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "role": "user"

  }

}

```

**### 2.8 User Logout**

- **\*\*Endpoint\*\***: `POST /api/auth/logout`

- **\*\*Description\*\***: Logs out user and clears JWT cookie

- **\*\*Access\*\***: Public

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "message": "Logout successful"

}

```

---

**## 3. User Management (/api/users)**

**### 3.1 Get All Users**

- **\*\*Endpoint\*\***: `GET /api/users`

- **\*\*Description\*\***: Retrieves all users with booking details and statistics

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Success Response (200)\*\***:

```json

[

  {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "gender": "male",

    "joinDate": "2025-01-15T00:00:00.000Z",

    "address": {

      "street": "123 Main St",

      "city": "Mumbai",

      "state": "Maharashtra",

      "pincode": "400001"

    },

    "emergencyContact": {

      "name": "Jane Doe",

      "phone": "9876543211",

      "relation": "spouse"

    },

    "bookings": ["BK001", "BK002"],

    "bookingDetails": [ ... ],

    "stats": {

      "totalBookings": 5,

      "cancelledBookings": 1,

      "totalSpent": 7500,

      "completedBookings": 3,

      "upcomingBookings": 1,

      "averageRating": 4.2

    }

  }

]

```

**### 3.2 Get User Details**

- **\*\*Endpoint\*\***: `GET /api/users/:id`

- **\*\*Description\*\***: Retrieves specific user details by user ID with comprehensive booking information

- **\*\*Access\*\***: Private (User can access own profile, Admin can access any)

- **\*\*URL Parameters\*\***: `id` - User ID (e.g., "U1234")

- **\*\*Success Response (200)\*\***:

```json

{

  "user": {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "stats": {

      "totalBookings": 5,

      "cancelledBookings": 1,

      "totalSpent": 7500,

      "completedBookings": 3,

      "upcomingBookings": 1,

      "averageRating": 4.2

    }

  },

  "bookings": [

    {

      "id": "BK001",

      "schedule": { ... },

      "fareDetails": { ... },

      "status": { "current": "completed" }

    }

  ]

}

```

- **\*\*Error Responses\*\***:

  - `404 Not Found`: User not found

  - `500 Server Error`: Internal server error

---

**## 4. Bus Management (/api/buses)**

**### 4.1 Get All Buses**

- **\*\*Endpoint\*\***: `GET /api/buses`

- **\*\*Description\*\***: Retrieves all buses in the fleet

- **\*\*Access\*\***: Public

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "results": 15,

  "data": {

    "buses": [

      {

        "id": "BUS001",

        "registrationNumber": "MH12AB1234",

        "type": "AC Sleeper",

        "totalSeats": 40,

        "berthStructure": {

          "lower": 20,

          "upper": 20,

          "layout": ["2x1", "single"]

        },

        "amenities": ["AC", "WiFi", "Charging Ports", "Blankets"],

        "features": {

          "hasWifi": true,

          "hasToilet": true,

          "isDoubleDecker": false,

          "isPremium": true

        },

        "status": "available",

        "isActive": true,

        "maintenanceHistory": [ ... ],

        "lastServiceDate": "2025-07-15T00:00:00.000Z",

        "nextServiceDue": "2025-10-15T00:00:00.000Z"

      }

    ]

  }

}

```

**### 4.2 Get Bus Details**

- **\*\*Endpoint\*\***: `GET /api/buses/:id`

- **\*\*Description\*\***: Retrieves specific bus details by bus ID

- **\*\*Access\*\***: Public

- **\*\*URL Parameters\*\***: `id` - Bus ID (e.g., "BUS001")

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "bus": {

      "id": "BUS001",

      "registrationNumber": "MH12AB1234",

      "type": "AC Sleeper",

      "totalSeats": 40,

      "berthStructure": {

        "lower": 20,

        "upper": 20,

        "layout": ["2x1", "single"]

      },

      "amenities": ["AC", "WiFi", "Charging Ports", "Blankets"],

      "features": {

        "hasWifi": true,

        "hasToilet": true,

        "isDoubleDecker": false,

        "isPremium": true

      },

      "status": "available",

      "isActive": true

    }

  }

}

```

- **\*\*Error Responses\*\***:

  - `404 Not Found`: Bus not found

  - `500 Server Error`: Internal server error

**### 4.3 Create Bus**

- **\*\*Endpoint\*\***: `POST /api/buses`

- **\*\*Description\*\***: Adds a new bus to the fleet

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Request Body\*\***:

```json

{

  "id": "BUS003",

  "registrationNumber": "MH12CD5678",

  "type": "AC Sleeper",

  "totalSeats": 40,

  "berthStructure": {

    "lower": 20,

    "upper": 20,

    "layout": ["2x1", "single"]

  },

  "amenities": ["AC", "WiFi", "Charging Ports", "Blankets", "Water Bottle"],

  "features": {

    "hasWifi": true,

    "hasToilet": true,

    "isDoubleDecker": false,

    "isPremium": true

  },

  "status": "available",

  "isActive": true

}

```

- **\*\*Success Response (201)\*\***:

```json

{

  "success": true,

  "data": {

    "bus": {

      "id": "BUS003",

      "registrationNumber": "MH12CD5678",

      "type": "AC Sleeper",

      "totalSeats": 40,

      "\_id": "66b234567890abcdef123459",

      "createdAt": "2025-08-01T10:00:00.000Z"

    }

  }

}

```

- **\*\*Error Responses\*\***:

  - `409 Conflict`: Duplicate registration number or bus ID

  - `400 Bad Request`: Invalid input data

  - `500 Server Error`: Internal server error

**### 4.4 Update Bus**

- **\*\*Endpoint\*\***: `PUT /api/buses/:id`

- **\*\*Description\*\***: Updates existing bus details

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Bus ID

- **\*\*Request Body\*\*** (partial update supported):

```json

{

  "type": "AC Semi Sleeper",

  "totalSeats": 45,

  "amenities": ["AC", "WiFi", "Charging Ports", "Entertainment", "Blanket", "Water Bottle"],

  "status": "maintenance"

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "bus": {

      "id": "BUS003",

      "registrationNumber": "MH12CD5678",

      "type": "AC Semi Sleeper",

      "totalSeats": 45,

      "status": "maintenance",

      "updatedAt": "2025-08-01T11:00:00.000Z"

    }

  }

}

```

- **\*\*Error Responses\*\***:

  - `404 Not Found`: Bus not found

  - `400 Bad Request`: Validation error

  - `409 Conflict`: Duplicate field values

  - `500 Server Error`: Internal server error

**### 4.5 Delete Bus**

- **\*\*Endpoint\*\***: `DELETE /api/buses/:id`

- **\*\*Description\*\***: Removes bus from the fleet

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Bus ID

- **\*\*Success Response (204)\*\***:

```json

{

  "status": "success",

  "data": null

}

```

- **\*\*Error Responses\*\***:

  - `404 Not Found`: Bus not found

  - `500 Server Error`: Internal server error

---

**## 5. Driver Management (/api/drivers)**

**### 5.1 Get All Drivers**

- **\*\*Endpoint\*\***: `GET /api/drivers`

- **\*\*Description\*\***: Retrieves all drivers with their schedules and availability

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Success Response (200)\*\***:

```json

[

  {

    "driverId": "DRV001",

    "name": "Rajesh Kumar",

    "mobile": "9876543214",

    "licenseNumber": "MH1234567890",

    "licenseExpiry": "2027-12-31T00:00:00.000Z",

    "gender": "male",

    "address": {

      "street": "321 Driver Colony",

      "city": "Mumbai",

      "state": "Maharashtra",

      "pincode": "400050"

    },

    "dateOfBirth": "1985-03-15T00:00:00.000Z",

    "bloodGroup": "B+",

    "emergencyContact": {

      "name": "Sunita Kumar",

      "relation": "wife",

      "mobile": "9876543215"

    },

    "joiningDate": "2020-01-15T00:00:00.000Z",

    "experienceYears": 8,

    "status": "active",

    "availableTimings": {

      "weekdays": ["06:00-14:00", "22:00-06:00"],

      "weekends": ["08:00-20:00"]

    },

    "rating": 4.5,

    "totalTripsCompleted": 245,

    "schedules": [

      {

        "id": "SCH001",

        "route": { ... },

        "timings": { ... }

      }

    ]

  }

]

```

**### 5.2 Get Driver Details**

- **\*\*Endpoint\*\***: `GET /api/drivers/:id`

- **\*\*Description\*\***: Retrieves specific driver details by driver ID

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Driver ID (e.g., "DRV001")

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "driver": {

      "driverId": "DRV001",

      "name": "Rajesh Kumar",

      "mobile": "9876543214",

      "licenseNumber": "MH1234567890",

      "experienceYears": 8,

      "status": "active",

      "assignedSchedules": [ ... ]

    }

  }

}

```

**### 5.3 Create Driver**

- **\*\*Endpoint\*\***: `POST /api/drivers`

- **\*\*Description\*\***: Creates a new driver profile

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Request Body\*\***:

```json

{

  "driverId": "DRV003",

  "name": "Rajesh Kumar",

  "mobile": "9876543214",

  "licenseNumber": "MH1234567890",

  "licenseExpiry": "2027-12-31",

  "gender": "male",

  "address": {

    "street": "321 Driver Colony",

    "city": "Mumbai",

    "state": "Maharashtra",

    "pincode": "400050"

  },

  "dateOfBirth": "1985-03-15",

  "bloodGroup": "B+",

  "emergencyContact": {

    "name": "Sunita Kumar",

    "relation": "wife",

    "mobile": "9876543215"

  },

  "joiningDate": "2020-01-15",

  "experienceYears": 8,

  "status": "active",

  "availableTimings": {

    "weekdays": ["06:00-14:00", "22:00-06:00"],

    "weekends": ["08:00-20:00"]

  }

}

```

- **\*\*Success Response (201)\*\***:

```json

{

  "status": "success",

  "data": {

    "driver": {

      "driverId": "DRV003",

      "name": "Rajesh Kumar",

      "mobile": "9876543214",

      "licenseNumber": "MH1234567890",

      "\_id": "66b234567890abcdef12345a",

      "createdAt": "2025-08-01T10:00:00.000Z"

    }

  }

}

```

**### 5.4 Update Driver**

- **\*\*Endpoint\*\***: `PUT /api/drivers/:id`

- **\*\*Description\*\***: Updates driver information

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Driver ID

- **\*\*Request Body\*\*** (partial update supported):

```json

{

  "mobile": "9876543215",

  "experienceYears": 9,

  "status": "on\_duty"

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "driver": {

      "driverId": "DRV003",

      "mobile": "9876543215",

      "experienceYears": 9,

      "status": "on\_duty",

      "updatedAt": "2025-08-01T11:00:00.000Z"

    }

  }

}

```

**### 5.5 Delete Driver**

- **\*\*Endpoint\*\***: `DELETE /api/drivers/:id`

- **\*\*Description\*\***: Removes driver from the system

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Driver ID

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "driver": null

  }

}

```

---

**## 6. Schedule Management (/api/schedules)**

**### 6.1 Get All Schedules**

- **\*\*Endpoint\*\***: `GET /api/schedules`

- **\*\*Description\*\***: Retrieves all bus schedules with route information

- **\*\*Access\*\***: Public

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "results": 25,

  "data": {

    "schedules": [

      {

        "id": "SCH001",

        "route": {

          "from": "Mumbai",

          "to": "Pune",

          "distance": 150,

          "duration": "3h 30m",

          "description": "Express route via highway"

        },

        "timings": {

          "departure": "2025-08-15T06:00:00.000Z",

          "arrival": "2025-08-15T09:30:00.000Z"

        },

        "bus": {

          "id": "BUS001",

          "type": "AC Sleeper",

          "registrationNumber": "MH12AB1234",

          "amenities": ["AC", "WiFi", "Charging Point"],

          "features": {

            "hasWifi": true,

            "hasToilet": true,

            "isDoubleDecker": false

          }

        },

        "stops": {

          "via": [

            { "city": "Lonavala", "duration": "15 min" }

          ],

          "pickupPoints": [

            {

              "location": "Mumbai Central",

              "time": "06:00",

              "address": "Mumbai Central Railway Station",

              "coordinates": { "lat": 19.0330, "lng": 72.8570 }

            }

          ],

          "dropPoints": [

            {

              "location": "Pune Station",

              "time": "09:30",

              "address": "Pune Railway Station",

              "coordinates": { "lat": 18.5284, "lng": 73.8748 }

            }

          ]

        },

        "pricing": {

          "baseFare": 500,

          "acCharge": 200,

          "serviceTax": 50,

          "totalFare": 750

        },

        "availability": {

          "totalSeats": 40,

          "availableSeats": 35,

          "bookedSeats": ["A1", "A2", "B1", "B2", "C1"]

        },

        "status": "active"

      }

    ]

  }

}

```

**### 6.2 Get Schedule Details**

- **\*\*Endpoint\*\***: `GET /api/schedules/:id`

- **\*\*Description\*\***: Retrieves specific schedule details by schedule ID

- **\*\*Access\*\***: Public

- **\*\*URL Parameters\*\***: `id` - Schedule ID (e.g., "SCH001")

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "schedule": {

      "id": "SCH001",

      "route": { ... },

      "timings": { ... },

      "bus": { ... },

      "stops": { ... },

      "pricing": { ... },

      "availability": { ... },

      "status": "active"

    }

  }

}

```

**### 6.3 Create Schedule**

- **\*\*Endpoint\*\***: `POST /api/schedules`

- **\*\*Description\*\***: Creates a new bus schedule

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Request Body\*\***:

```json

{

  "id": "SCH003",

  "bus": {

    "id": "BUS001"

  },

  "driver": {

    "driverId": "DRV001"

  },

  "route": {

    "from": "Mumbai",

    "to": "Pune",

    "distance": 150,

    "duration": "3h 30m",

    "description": "Express route via highway"

  },

  "timings": {

    "departure": "2025-08-15T06:00:00.000Z",

    "arrival": "2025-08-15T09:30:00.000Z"

  },

  "stops": {

    "via": [

      { "city": "Lonavala", "duration": "15 min" }

    ],

    "pickupPoints": [

      {

        "location": "Mumbai Central",

        "time": "06:00",

        "address": "Mumbai Central Railway Station",

        "coordinates": { "lat": 19.0330, "lng": 72.8570 }

      }

    ],

    "dropPoints": [

      {

        "location": "Pune Station",

        "time": "09:30",

        "address": "Pune Railway Station",

        "coordinates": { "lat": 18.5284, "lng": 73.8748 }

      }

    ]

  },

  "pricing": {

    "baseFare": 500,

    "acCharge": 200,

    "serviceTax": 50,

    "totalFare": 750

  },

  "availability": {

    "totalSeats": 40,

    "availableSeats": 40,

    "bookedSeats": []

  },

  "status": "active"

}

```

- **\*\*Success Response (201)\*\***:

```json

{

  "status": "success",

  "data": {

    "schedule": {

      "id": "SCH003",

      "\_id": "66b234567890abcdef12345b",

      "route": { ... },

      "timings": { ... },

      "createdAt": "2025-08-01T10:00:00.000Z"

    }

  }

}

```

**### 6.4 Update Schedule**

- **\*\*Endpoint\*\***: `PUT /api/schedules/:id`

- **\*\*Description\*\***: Updates schedule information

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Schedule ID

- **\*\*Request Body\*\*** (partial update supported):

```json

{

  "pricing": {

    "baseFare": 550,

    "acCharge": 250,

    "serviceTax": 60,

    "totalFare": 860

  },

  "timings": {

    "departure": "2025-08-15T06:30:00.000Z",

    "arrival": "2025-08-15T10:00:00.000Z"

  }

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "schedule": {

      "id": "SCH003",

      "pricing": { ... },

      "timings": { ... },

      "updatedAt": "2025-08-01T11:00:00.000Z"

    }

  }

}

```

**### 6.5 Delete Schedule**

- **\*\*Endpoint\*\***: `DELETE /api/schedules/:id`

- **\*\*Description\*\***: Removes schedule from the system

- **\*\*Access\*\***: Private (Admin only)

- **\*\*URL Parameters\*\***: `id` - Schedule ID

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "data": {

    "schedule": null

  }

}

```

---

**## 7. Booking Management (/api/bookings)**

**### 7.1 Get All Bookings**

- **\*\*Endpoint\*\***: `GET /api/bookings`

- **\*\*Description\*\***: Retrieves all bookings with passenger and journey details

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Success Response (200)\*\***:

```json

[

  {

    "id": "BK12345",

    "user": {

      "id": "U1234",

      "name": "John Doe",

      "email": "john.doe@example.com",

      "phone": "9876543210",

      "address": {

        "street": "123 Main St",

        "city": "Mumbai",

        "state": "Maharashtra",

        "pincode": "400001",

        "coordinates": { "lat": 19.0760, "lng": 72.8777 }

      }

    },

    "schedule": {

      "id": "SCH001",

      "from": { "city": "Mumbai", "terminal": "Mumbai Central", "departure": "06:00" },

      "to": { "city": "Pune", "terminal": "Pune Station", "arrival": "09:30" },

      "bus": {

        "id": "BUS001",

        "type": "AC Sleeper",

        "amenities": ["AC", "WiFi", "Charging Point"]

      },

      "via": [

        { "city": "Lonavala", "stopDuration": 15 }

      ],

      "distance": 150,

      "duration": "3h 30m"

    },

    "seat": ["A1", "A2"],

    "passengerDetails": [

      {

        "name": "John Doe",

        "age": 30,

        "gender": "male"

      },

      {

        "name": "Jane Doe",

        "age": 28,

        "gender": "female"

      }

    ],

    "bookingDetails": {

      "date": "2025-08-01T10:00:00.000Z",

      "travelDate": "2025-08-15T00:00:00.000Z",

      "confirmationNumber": "CNF123456"

    },

    "fareDetails": {

      "baseFare": 1400,

      "tax": 70,

      "discount": {

        "code": "FIRST10",

        "amount": 140

      },

      "surcharges": [

        { "type": "Service Fee", "amount": 30 }

      ],

      "total": 1360

    },

    "status": {

      "current": "confirmed"

    }

  }

]

```

**### 7.2 Create Booking**

- **\*\*Endpoint\*\***: `POST /api/bookings`

- **\*\*Description\*\***: Creates a new booking with passenger details

- **\*\*Access\*\***: Private (Authenticated user)

- **\*\*Request Body\*\***:

```json

{

  "id": "BK12346",

  "user": {

    "id": "U1234",

    "name": "John Doe",

    "email": "john.doe@example.com",

    "phone": "9876543210",

    "address": {

      "street": "123 Main St",

      "city": "Mumbai",

      "state": "Maharashtra",

      "pincode": "400001"

    }

  },

  "schedule": {

    "id": "SCH001"

  },

  "seat": ["A3", "A4"],

  "passengerDetails": [

    {

      "name": "John Doe",

      "age": 30,

      "gender": "male"

    },

    {

      "name": "Jane Doe",

      "age": 28,

      "gender": "female"

    }

  ],

  "bookingDetails": {

    "date": "2025-08-01T10:00:00.000Z",

    "travelDate": "2025-08-15T00:00:00.000Z"

  },

  "fareDetails": {

    "baseFare": 1400,

    "tax": 70,

    "total": 1470

  },

  "status": {

    "current": "confirmed"

  }

}

```

- **\*\*Success Response (201)\*\***:

```json

{

  "message": "Booking created successfully",

  "booking": {

    "id": "BK12346",

    "\_id": "66b234567890abcdef12345c",

    "user": { ... },

    "schedule": { ... },

    "createdAt": "2025-08-01T10:00:00.000Z"

  }

}

```

**### 7.3 Update Booking**

- **\*\*Endpoint\*\***: `PUT /api/bookings/:id`

- **\*\*Description\*\***: Updates booking information

- **\*\*Access\*\***: Private (User can update own booking, Admin can update any)

- **\*\*URL Parameters\*\***: `id` - Booking ID

- **\*\*Request Body\*\*** (partial update supported):

```json

{

  "status": {

    "current": "completed"

  },

  "passengerDetails": [

    {

      "name": "John Smith",

      "age": 31,

      "gender": "male"

    }

  ]

}

```

- **\*\*Success Response (200)\*\***:

```json

{

  "success": true,

  "message": "Booking updated successfully",

  "data": {

    "id": "BK12346",

    "status": { "current": "completed" },

    "updatedAt": "2025-08-01T11:00:00.000Z"

  }

}

```

---

**## 8. Dashboard Analytics (/api/dashboard)**

**### 8.1 Get Dashboard Data**

- **\*\*Endpoint\*\***: `GET /api/dashboard`

- **\*\*Description\*\***: Retrieves comprehensive dashboard analytics with real-time statistics

- **\*\*Access\*\***: Private (Admin only)

- **\*\*Success Response (200)\*\***:

```json

{

  "totalBuses": 15,

  "totalRevenue": 125000,

  "monthlyBookings": 48,

  "totalCustomers": 234,

  "cancelledTickets": 5,

  "recentBookings": [

    {

      "id": "BK12345",

      "passengerName": "John Doe, Jane Doe",

      "from": "Mumbai",

      "to": "Pune",

      "travelDate": "2025-08-15",

      "bookingDate": "2025-08-01",

      "fare": 1470,

      "status": "confirmed",

      "busType": "AC Sleeper",

      "passengerCount": 2

    }

  ]

}

```

- **\*\*Error Responses\*\***:

  - `500 Server Error`: Database connection or query error

---

**## 9. Search & Filter Operations**

**### 9.1 Search Buses by Route**

- **\*\*Endpoint\*\***: `GET /api/buses/search`

- **\*\*Description\*\***: Search for available buses on specific route and date

- **\*\*Access\*\***: Public

- **\*\*Query Parameters\*\***:

  - `from` (required): Origin city

  - `to` (required): Destination city

  - `date` (required): Travel date (YYYY-MM-DD)

  - `type` (optional): Bus type filter

- **\*\*Example\*\***: `GET /api/buses/search?from=Mumbai&to=Pune&date=2025-08-15&type=AC Sleeper`

- **\*\*Success Response (200)\*\***:

```json

{

  "status": "success",

  "results": 5,

  "data": {

    "buses": [

      {

        "id": "BUS001",

        "type": "AC Sleeper",

        "departure": "06:00",

        "arrival": "09:30",

        "duration": "3h 30m",

        "fare": 750,

        "availableSeats": 35

      }

    ]

  }

}

```

**### 9.2 Filter Buses by Type and Status**

- **\*\*Endpoint\*\***: `GET /api/buses`

- **\*\*Description\*\***: Filter buses by type and operational status

- **\*\*Access\*\***: Public

- **\*\*Query Parameters\*\***:

  - `type` (optional): Bus type (e.g., "AC Sleeper", "Non AC Seater")

  - `status` (optional): Bus status (e.g., "available", "maintenance")

- **\*\*Example\*\***: `GET /api/buses?type=AC Sleeper&status=available`

**### 9.3 Search Schedules by Route**

- **\*\*Endpoint\*\***: `GET /api/schedules`

- **\*\*Description\*\***: Search schedules for specific route

- **\*\*Access\*\***: Public

- **\*\*Query Parameters\*\***:

  - `from` (optional): Origin city

  - `to` (optional): Destination city

- **\*\*Example\*\***: `GET /api/schedules?from=Mumbai&to=Pune`

---

**## 10. Ticket Operations**

**### 10.1 Track Ticket by Booking ID**

- **\*\*Endpoint\*\***: `GET /api/bookings/:id`

- **\*\*Description\*\***: Track ticket status using booking ID

- **\*\*Access\*\***: Public (with booking ID and contact verification)

- **\*\*URL Parameters\*\***: `id` - Booking ID

**### 10.2 Get User Bookings**

- **\*\*Endpoint\*\***: `GET /api/users/:id`

- **\*\*Description\*\***: Get all bookings for specific user with statistics

- **\*\*Access\*\***: Private (User can access own bookings, Admin can access any)

- **\*\*URL Parameters\*\***: `id` - User ID

---

**## 11. Admin Operations**

**### 11.1 Get System Status**

- **\*\*Endpoint\*\***: `GET /api/health`

- **\*\*Description\*\***: Check overall system health and operational status

- **\*\*Access\*\***: Public

**### 11.2 Get Fleet Management Data**

- **\*\*Endpoint\*\***: `GET /api/buses`

- **\*\*Description\*\***: Get complete fleet management information including maintenance schedules

- **\*\*Access\*\***: Private (Admin only)

**### 11.3 Get Driver Operations Data**

- **\*\*Endpoint\*\***: `GET /api/drivers`

- **\*\*Description\*\***: Get all driver operations, availability, and performance metrics

- **\*\*Access\*\***: Private (Admin only)

**### 11.4 Get Complete Dashboard Analytics**

- **\*\*Endpoint\*\***: `GET /api/dashboard`

- **\*\*Description\*\***: Get comprehensive business analytics and operational metrics

- **\*\*Access\*\***: Private (Admin only)

---

**## Error Handling**

All API endpoints follow a consistent error response format:

**### Success Responses**

- **\*\*200 OK\*\***: Successful GET, PUT requests

- **\*\*201 Created\*\***: Successful POST requests

- **\*\*204 No Content\*\***: Successful DELETE requests

**### Error Responses**

- **\*\*400 Bad Request\*\***: Invalid input data or missing required fields

- **\*\*401 Unauthorized\*\***: Authentication required or invalid credentials

- **\*\*403 Forbidden\*\***: Insufficient permissions

- **\*\*404 Not Found\*\***: Resource not found

- **\*\*409 Conflict\*\***: Duplicate data (e.g., registration number already exists)

- **\*\*500 Internal Server Error\*\***: Server-side error

**### Error Response Format**

```json

{

  "status": "error" | "fail",

  "message": "Human-readable error message",

  "error": "Detailed error information (development only)",

  "field": "fieldName (for validation errors)"

}

```

---

**## Data Models**

**### Bus Model**

```typescript

interface Bus {

  id: string;

  registrationNumber: string;

  type: 'AC Sleeper' | 'Non AC Seater' | 'AC Seater' | 'Non AC Sleeper' | 'Premium';

  totalSeats: number;

  berthStructure?: {

    lower: number;

    upper?: number;

    layout?: string[];

  };

  amenities?: string[];

  features?: {

    hasWifi?: boolean;

    hasToilet?: boolean;

    isDoubleDecker?: boolean;

    isPremium?: boolean;

  };

  status: 'available' | 'maintenance' | 'unavailable';

  maintenanceHistory?: MaintenanceRecord[];

  lastServiceDate?: Date;

  nextServiceDue?: Date;

  isActive: boolean;

}

```

**### User Model**

```typescript

interface User {

  id: string;

  name: string;

  email: string;

  phone: string;

  gender: 'male' | 'female' | 'other';

  joinDate: Date;

  address?: {

    street?: string;

    city?: string;

    state?: string;

    pincode?: string;

  };

  emergencyContact?: {

    name?: string;

    phone?: string;

    relation?: string;

  };

  bookings: string[];

  stats?: {

    totalBookings: number;

    cancelledBookings: number;

    totalSpent: number;

    completedBookings: number;

    upcomingBookings: number;

    averageRating?: number;

  };

}

```

**### Booking Model**

```typescript

interface Booking {

  id: string;

  user: {

    id: string;

    name: string;

    email: string;

    phone: string;

    address?: Address;

  };

  schedule: {

    id: string;

    from?: RoutePoint;

    to?: RoutePoint;

    bus?: BusInfo;

    via?: StopInfo[];

    distance?: number;

    duration?: string;

  };

  seat: string[];

  passengerDetails: PassengerInfo[];

  bookingDetails: {

    date: Date;

    travelDate: Date;

    confirmationNumber?: string;

  };

  fareDetails: {

    baseFare: number;

    tax?: number;

    discount?: DiscountInfo;

    surcharges?: SurchargeInfo[];

    total: number;

  };

  status: {

    current: string;

  };

}

```

---

**## Rate Limiting & Security**

**### Rate Limiting**

- **\*\*General API calls\*\***: 100 requests per 15 minutes per IP

- **\*\*Authentication endpoints\*\***: 5 requests per 15 minutes per IP

- **\*\*Search endpoints\*\***: 30 requests per minute per IP

**### Security Features**

- JWT authentication with HTTP-only cookies

- CORS enabled for specified origins

- Request validation and sanitization

- SQL injection protection via Mongoose ODM

- Password hashing with bcrypt

- Environment-based configuration

---

**## Testing & Development**

**### Test Data**

The API includes test endpoints for creating sample data:

- `POST /api/buses` - Test bus creation

- `POST /api/drivers` - Test driver creation

- `POST /api/schedules` - Test schedule creation

- `POST /api/bookings` - Test booking creation

**### Environment Variables**

```env

NODE\_ENV=development

PORT=5000

MONGODB\_URI=mongodb://localhost:27017/easytrip

JWT\_SECRET=your-secret-key

JWT\_EXPIRE=7d

```

---

**## Support & Contact**

For API support and technical assistance:

- **\*\*Email\*\***: support@easytrip.com

- **\*\*Phone\*\***: 1800-123-4567

- **\*\*Documentation\*\***: This document

- **\*\*Repository\*\***: Your project repository

---

*\*Last Updated: August 1, 2025\**

*\*API Version: 1.0.0\**

*\*Documentation Version: 1.0.0\**