

School Transport Management API - Complete Documentation

Table of Contents

1. [Overview](#overview)
 2. [Authentication APIs](#authentication-apis)
 3. [Admin APIs](#admin-apis)
 4. [Parent APIs](#parent-apis)
 5. [Driver APIs](#driver-apis)
 6. [Route APIs](#route-apis)
 7. [Bus APIs](#bus-apis)
 8. [Route Stop APIs](#route-stop-apis)
 9. [Student APIs](#student-apis)
 10. [Trip APIs](#trip-apis)
 11. [Database Table Models](#database-table-models)
 12. [Entity Relationships](#entity-relationships)
 13. [Security & Integration](#security--integration)
-

Overview

****Base URL**:** `http://72.61.250.191:8080/api/v1`

****API Version**:** 1.0.0

****Authentication**:** JWT Bearer Token

Authentication APIs

1. Login

****Endpoint**:** `POST /auth/login`

****Description**:** Universal login for admins, parents, and drivers.

****Request Body**:**

```
{  
  "phone": 9876543210,  
  "password": "your_password"  
}
```

2. Get Profile

****Endpoint**:** `GET /auth/profile?phone={phone_number}`

****Description**:** Get user profile by phone number.

Admin APIs

1. Create Admin

****Endpoint**:** `POST /admins`

2. Get All Admins

****Endpoint**:** `GET /admins`

3. Update Admin Status

****Endpoint**:** `PUT /admins/{admin_id}/status`

****Status Options**:** `ACTIVE`, `INACTIVE`

Parent APIs

1. Create Parent

****Endpoint**:** `POST /parents`

2. Update Parent FCM Token

****Endpoint**:** `PATCH /parents/{parent_id}/fcm-token`

Driver APIs

1. Create Driver

****Endpoint**:** `POST /drivers`

2. Update Driver Status

****Endpoint**:** `PUT /drivers/{driver_id}/status`

****Available Status Values**:** `ACTIVE`, `INACTIVE`, `SUSPENDED` (Use for suspended drivers)

Bus APIs

1. Create Bus

****Endpoint**:** `POST /buses`

2. Update Bus Status

****Endpoint**:** `PUT /buses/{bus_id}/status`

****Available Status Values**:** `ACTIVE`, `INACTIVE`, `MAINTENANCE` (Use for buses in maintenance)

Student APIs

1. Create Student

****Endpoint**:** `POST /students`

****Description**:** Create a new student. Provides specific error messages for invalid references.

****Error Handling**:** Returns detailed 400 errors if parent_id, route_id, or stop_id is invalid.

2. Update Student Transport Status

****Endpoint**:** `PUT /students/{student_id}/status`

****Transport Status Options**:** `ACTIVE`, `TEMP_STOP`, `CANCELLED`

Trip APIs

1. Create Trip

****Endpoint**:** `POST /trips`

****Trip Type**:** `MORNING`, `EVENING`

2. Update Trip Status

****Endpoint**:** `PUT /trips/{trip_id}/status`

****Trip Status Options**:** `NOT_STARTED`, `ONGOING`, `PAUSED`, `COMPLETED`, `CANCELED`

Database Table Models

1. Admins Table

Column	Type	Description
admin_id	VARCHAR(36)	Primary Key
phone	BIGINT	Unique Phone
status	VARCHAR(20)	ACTIVE, INACTIVE

2. Drivers Table

Column	Type	Description
--------	------	-------------

Column	Type	Description
driver_id	VARCHAR(36)	Primary Key
status	VARCHAR(20)	ACTIVE, INACTIVE, SUSPENDED

3. Buses Table

Column	Type	Description
bus_id	VARCHAR(36)	Primary Key
status	VARCHAR(20)	ACTIVE, INACTIVE, MAINTENANCE

Entity Relationships

```

Parents (has) Students
  Students (assigned to) Routes
    Routes (has) Route Stops
      Routes (used by) Buses
        Buses (driven by) Drivers

Trips (combines): Bus, Driver, Route
  
```

Security Features

1. **JWT Authentication**: Secure token-based auth for all protected endpoints.
2. **Password Hashing**: Bcrypt encryption for all user passwords.
3. **Foreign Key Integrity**: Strict database constraints with clear API error reporting.

Interactive Documentation:

- [Swagger UI](http://72.61.250.191:8080/docs)
- [ReDoc](http://72.61.250.191:8080/redoc)