

# API Documentation

## User Endpoints

### Register a New User

URL: /api/Register

Method: POST

Description: Registers a new user.

Request Body:

```
{  
  "name": "John Doe",  
  "email": "john.doe@example.com",  
  "password": "password123"  
}
```

Response:

201 Created:

```
{  
  "message": "New user created successfully"  
}
```

409 Conflict:

```
{  
  "message": "The user already exist, try logging in"  
}
```

### User Login

URL: /api/Login

Method: POST

Description: Logs in a user.

Request Body:

```
{  
  "email": "john.doe@example.com",  
}
```

```
"password": "password123"
```

```
}
```

Response:

200 OK:

```
{
```

```
  "message": "User logged In Successfully",
```

```
  "user": {
```

```
    "name": "John Doe",
```

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

```
  }
```

```
}
```

403 Forbidden:

```
{
```

```
  "message": "The specified user is not registered"
```

```
}
```

### **Book a Bus**

URL: /api/BookBus

Method: PUT

Description: Books a bus for the logged-in user.

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{
```

```
  "busId": "60d21b4667d0d8992e610c85"
```

```
}
```

Response:

200 OK:

```
{
```

```
  "message": "Booking successfully",
```

```
"bookings": ["60d21b4667d0d8992e610c85"]
}
```

### **Cancel a Bus Booking**

URL: /api/CancelBus

Method: PUT

Description: Cancels a bus booking for the logged-in user.

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{
  "busId": "60d21b4667d0d8992e610c85"
}
```

Response:

200 OK:

```
{
  "message": "Cancellation done !!!",
  "bookings": []
}
```

### **Delete User Account**

URL: /api/DeleteAcc

Method: DELETE

Description: Deletes the logged-in user's account.

Request Headers:

Authorization: Bearer <access\_token>

Response:

200 OK:

```
{
  "message": "Your account has been deleted successfully"
}
```

## **User Logout**

URL: /api/Logout

Method: GET

Description: Logs out the user.

Response:

200 OK:

```
{  
  "message": "User logged out successfully"  
}
```

## **Refresh Token**

URL: /api/refreshToken

Method: GET

Description: Refreshes the access token.

Request Headers:

Authorization: Bearer <refresh\_token>

Response:

200 OK:

```
{  
  "message": "The token refreshed successfully"  
}
```

## **Bus Endpoints**

### **Add a New Bus**

URL: /api/admin/addBus

Method: POST

Description: Adds a new bus (Admin only).

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{  
  "name": "City Express",  
  "routeNumber": "101",  
  "registerNumber": "AB1234",  
  "route": ["Stop1", "Stop2", "Stop3", "Stop4"],  
  "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]  
}
```

Response:

201 Created:

```
{  
  "message": "Bus added successfully",  
  "bus": {  
    "_id": "60d21b4667d0d8992e610c85",  
    "name": "City Express",  
    "routeNumber": "101",  
    "registerNumber": "AB1234",  
    "route": ["Stop1", "Stop2", "Stop3", "Stop4"],  
    "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]  
  }  
}
```

### **Search for Buses**

URL: /api/searchBus

Method: GET

Description: Searches for buses based on route number, name, or route.

Request Headers:

Authorization: Bearer <access\_token>

Request Query Parameters:

routeNumber (optional)

name (optional)

route (optional)

Response:

200 OK:

```
{
  "buses": [
    {
      "_id": "60d21b4667d0d8992e610c85",
      "name": "City Express",
      "routeNumber": "101",
      "registerNumber": "AB1234",
      "route": ["Stop1", "Stop2", "Stop3", "Stop4"],
      "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]
    }
  ]
}
```

### **Update Bus Information**

URL: /api/admin/updateBus

Method: PUT

Description: Updates bus information (Admin only).

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{
  "busId": "60d21b4667d0d8992e610c85",
  "name": "City Express",
  "routeNumber": "101",
  "registerNumber": "AB1234",
  "route": ["Stop1", "Stop2", "Stop3", "Stop4"],
  "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]
}
```

```
}
```

Response:

200 OK:

```
{  
  "message": "Bus updated successfully",  
  "bus": {  
    "_id": "60d21b4667d0d8992e610c85",  
    "name": "City Express",  
    "routeNumber": "101",  
    "registerNumber": "AB1234",  
    "route": ["Stop1", "Stop2", "Stop3", "Stop4"],  
    "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]  
  }  
}
```

### **Delete a Bus**

URL: /api/admin/deleteBus

Method: DELETE

Description: Deletes a bus (Admin only).

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{  
  "busId": "60d21b4667d0d8992e610c85"  
}
```

Response:

200 OK:

```
{  
  "message": "Bus deleted successfully"  
}
```

### **Add Routes to a Bus**

URL: /api/admin/addRoute

Method: PUT

Description: Adds routes to a bus (Admin only).

Request Headers:

Authorization: Bearer <access\_token>

Request Body:

```
{  
  "busId": "60d21b4667d0d8992e610c85",  
  "route": ["Stop5", "Stop6"]  
}
```

Response:

200 OK:

```
{  
  "message": "Route added successfully",  
  "bus": {  
    "_id": "60d21b4667d0d8992e610c85",  
    "name": "City Express",  
    "routeNumber": "101",  
    "registerNumber": "AB1234",  
    "route": ["Stop1", "Stop2", "Stop3", "Stop4", "Stop5", "Stop6"],  
    "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]  
  }  
}
```

### **Update Routes of a Bus**

URL: /api/admin/updateRoute

Method: PUT

Description: Updates routes of a bus (Admin only).

Request Headers:



Authorization: Bearer <access\_token>

Request Body:

```
{  
  "busId": "60d21b4667d0d8992e610c85",  
  "route": ["Stop1", "Stop2", "Stop3", "Stop4", "Stop5", "Stop6"]  
}
```

Response:

200 OK:

```
{  
  "message": "Route updated successfully",  
  "bus": {  
    "_id": "60d21b4667d0d8992e610c85",  
    "name": "City Express",  
    "routeNumber": "101",  
    "registerNumber": "AB1234",  
    "route": ["Stop1", "Stop2", "Stop3", "Stop4", "Stop5", "Stop6"],  
    "timing": ["08:00 AM", "12:00 PM", "04:00 PM"]  
  }  
}
```