RS-Express Bus Booking System - API Documentation

Authentication

Login

```
POST /api/login
```

Authenticates a user and returns a token.

Request Body:

```
{
  "email": "user@example.com",
  "password": "password"
}
```

Response:

```
{
  "status": "success",
  "user": {
    "id": 1,
    "name": "John Doe",
    "email": "user@example.com",
    "role": "user",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  },
  "permissions": ["view_dashboard", "make_booking"],
  "token": "6|laravel_sanctum_token..."
}
```

Register

```
POST /api/register
```

Creates a new user account.

Request Body:

```json
{
  "name": "John Doe",
  "email": "user@example.com",
  "password": "password",
  "password_confirmation": "password"
}
```

Response:

```json
{
  "status": "success",
  "message": "User registered successfully",
  "user": {
    "id": 1,
    "name": "John Doe",
    "email": "user@example.com",
    "role": "user",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  }
}
```

Logout

```
POST /api/logout
```

Invalidates the current user's token.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "status": "success",
  "message": "Logged out successfully"
}
```

Bus Management

List All Buses

```
GET /api/buses
```

Returns a list of all buses.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": [
    {
      "id": 1,
      "bus_no": "ABC-1234",
      "bus_type": "luxury",
      "seat_capacity": 40,
      "status": "active",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    },
    {
      "id": 2,
      "bus_no": "XYZ-5678",
      "bus_type": "standard",
      "seat_capacity": 50,
      "status": "active",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    }
  ],
  "links": {
    "first": "http://example.com/api/buses?page=1",
    "last": "http://example.com/api/buses?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "http://example.com/api/buses",
    "per_page": 15,
    "to": 2,
    "total": 2
  }
}
```

Get Bus Details

```
GET /api/buses/{id}
```

Returns details for a specific bus.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": {
    "id": 1,
    "bus_no": "ABC-1234",
    "bus_type": "luxury",
    "seat_capacity": 40,
    "status": "active",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  }
}
```

Create Bus

```
POST /api/buses
```

Creates a new bus record.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "bus_no": "DEF-9012",
  "bus_type": "luxury",
  "seat_capacity": 40,
  "status": "active"
}
```

Response:

```json
{
  "status": "success",
  "message": "Bus created successfully",
  "data": {
    "id": 3,
    "bus_no": "DEF-9012",
    "bus_type": "luxury",
    "seat_capacity": 40,
    "status": "active",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  }
}
```

Update Bus

```
PUT /api/buses/{id}
```

Updates an existing bus record.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "bus_no": "DEF-9012",
  "bus_type": "standard",
  "seat_capacity": 45,
  "status": "active"
}
```

Response:

```json
{
  "status": "success",
  "message": "Bus updated successfully",
  "data": {
    "id": 3,
    "bus_no": "DEF-9012",
    "bus_type": "standard",
    "seat_capacity": 45,
    "status": "active",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T11:15:00.000000Z"
  }
}
```

Delete Bus

```
DELETE /api/buses/{id}
```

Deletes a bus record.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "status": "success",
  "message": "Bus deleted successfully"
}
```

Route Management

List All Routes

```
GET /api/routes
```

Returns a list of all routes.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": [
    {
      "id": 1,
      "route_name": "Colombo - Kandy",
      "start_point": "Colombo",
      "end_point": "Kandy",
      "distance": 115,
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    },
    {
      "id": 2,
      "route_name": "Colombo - Galle",
      "start_point": "Colombo",
      "end_point": "Galle",
      "distance": 125,
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    }
  ],
  "links": {
    "first": "http://example.com/api/routes?page=1",
    "last": "http://example.com/api/routes?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "http://example.com/api/routes",
    "per_page": 15,
    "to": 2,
    "total": 2
  }
}
```

Create Route

```
POST /api/routes
```

Creates a new route.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```
{
  "route_name": "Colombo - Jaffna",
  "start_point": "Colombo",
  "end_point": "Jaffna",
  "distance": 398
}
```

Response:

```
{
  "status": "success",
  "message": "Route created successfully",
  "data": {
    "id": 3,
    "route_name": "Colombo - Jaffna",
    "start_point": "Colombo",
    "end_point": "Jaffna",
    "distance": 398,
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  }
}
```

Trip Management

List All Trips

```
GET /api/trips
```

Returns a list of all trips.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": [
    {
      "id": 1,
      "bus_id": 1,
      "route_id": 1,
      "departure_date": "2023-06-15",
      "departure_time": "08:00:00",
      "arrival_time": "11:30:00",
      "price": 1200,
      "available_seats": 40,
      "status": "scheduled",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z",
      "bus": {
        "id": 1,
        "bus_no": "ABC-1234",
        "bus_type": "luxury",
        "seat_capacity": 40
      },
      "route": {
        "id": 1,
        "route_name": "Colombo - Kandy",
        "start_point": "Colombo",
        "end_point": "Kandy",
        "distance": 115
      }
    }
  ],
  "links": {
    "first": "http://example.com/api/trips?page=1",
    "last": "http://example.com/api/trips?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "http://example.com/api/trips",
    "per_page": 15,
    "to": 1,
    "total": 1
  }
}
```

Search Trips

```
GET /api/trips/search?start_point=Colombo&end_point=Kandy&date=2023-06-15
```

Searches for trips by route and date.

Parameters:

- start_point (string): Departure location
- end_point (string): Destination location
- date (date): Travel date (YYYY-MM-DD)

Response:

```json
{
  "data": [
    {
      "id": 1,
      "bus_id": 1,
      "route_id": 1,
      "departure_date": "2023-06-15",
      "departure_time": "08:00:00",
      "arrival_time": "11:30:00",
      "price": 1200,
      "available_seats": 40,
      "status": "scheduled",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z",
      "bus": {
        "id": 1,
        "bus_no": "ABC-1234",
        "bus_type": "luxury",
        "seat_capacity": 40
      },
      "route": {
        "id": 1,
        "route_name": "Colombo - Kandy",
        "start_point": "Colombo",
        "end_point": "Kandy",
        "distance": 115
      }
    }
  ]
}
```

Create Trip

```
POST /api/trips
```

Creates a new trip.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "bus_id": 1,
  "route_id": 1,
  "departure_date": "2023-06-16",
  "departure_time": "08:00:00",
  "arrival_time": "11:30:00",
  "price": 1200
}
```

Response:

```json
{
  "status": "success",
  "message": "Trip created successfully",
  "data": {
    "id": 2,
    "bus_id": 1,
    "route_id": 1,
    "departure_date": "2023-06-16",
    "departure_time": "08:00:00",
    "arrival_time": "11:30:00",
    "price": 1200,
    "available_seats": 40,
    "status": "scheduled",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z"
  }
}
```

Booking Management

List All Bookings

```
GET /api/bookings                                    14 / 24
```

Returns a list of all bookings.

Headers:

```
Authorization: Bearer {token}
```

Booking Management

List All Bookings

Response:

```json
{
  "data": [
    {
      "id": 1,
      "user_id": 1,
      "bus_id": 1,
      "seat_no": "A1",
      "price": 1200,
      "status": "confirmed",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z",
      "trip": {
        "id": 1,
        "departure_date": "2023-06-15",
        "departure_time": "08:00:00",
        "route": {
          "id": 1,
          "route_name": "Colombo - Kandy",
          "start_point": "Colombo",
          "end_point": "Kandy"
        }
      },
      "user": {
        "id": 1,
        "name": "John Doe",
        "email": "user@example.com"
      }
    }
  ],
  "links": {
    "first": "http://example.com/api/bookings?page=1",
    "last": "http://example.com/api/bookings?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "http://example.com/api/bookings",
    "per_page": 15,
    "to": 1,
    "total": 1
  }
}
```

Create Booking

```
POST /api/bookings
```

Creates a new booking.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "trip_id": 1,
  "seat_no": "B5",
  "passenger_name": "John Doe",
  "passenger_email": "user@example.com",
  "passenger_phone": "1234567890"
}
```

Response:

```json
{
  "status": "success",
  "message": "Booking created successfully",
  "data": {
    "id": 2,
    "user_id": 1,
    "bus_id": 1,
    "trip_id": 1,
    "seat_no": "B5",
    "price": 1200,
    "status": "confirmed",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z",
    "passenger": {
      "name": "John Doe",
      "email": "user@example.com",
      "phone": "1234567890"
    },
    "booking_reference": "RS-12345"
  }
}
```

Get Booking Details

```
GET /api/bookings/{id}
```

Returns details for a specific booking.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": {
    "id": 1,
    "user_id": 1,
    "bus_id": 1,
    "trip_id": 1,
    "seat_no": "A1",
    "price": 1200,
    "status": "confirmed",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T10:30:00.000000Z",
    "passenger": {
      "name": "John Doe",
      "email": "user@example.com",
      "phone": "1234567890"
    },
    "trip": {
      "id": 1,
      "departure_date": "2023-06-15",
      "departure_time": "08:00:00",
      "bus": {
        "id": 1,
        "bus_no": "ABC-1234",
        "bus_type": "luxury"
      },
      "route": {
        "id": 1,
        "route_name": "Colombo - Kandy",
        "start_point": "Colombo",
        "end_point": "Kandy"
      }
    },
    "booking_reference": "RS-12345"
  }
}
```

Cancel Booking

```
POST /api/bookings/{id}/cancel
```

Cancels a booking.

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "reason": "Change of plans"
}
```

Response:

```json
{
  "status": "success",
  "message": "Booking cancelled successfully",
  "data": {
    "id": 1,
    "status": "cancelled",
    "cancellation": {
      "reason": "Change of plans",
      "refund_amount": 960,
      "cancellation_date": "2023-05-10T15:45:00.000000Z"
    }
  }
}
```

User Management

List All Users

```
GET /api/users
```

Returns a list of all users (admin only).

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "data": [
    {
      "id": 1,
      "name": "Admin User",
      "email": "admin@example.com",
      "role": "admin",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    },
    {
      "id": 2,
      "name": "John Doe",
      "email": "user@example.com",
      "role": "user",
      "created_at": "2023-05-10T10:30:00.000000Z",
      "updated_at": "2023-05-10T10:30:00.000000Z"
    }
  ],
  "links": {
    "first": "http://example.com/api/users?page=1",
    "last": "http://example.com/api/users?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "http://example.com/api/users",
    "per_page": 15,
    "to": 2,
    "total": 2
  }
}
```

Update User

```
PUT /api/users/{id}
```

Updates a user (admin only).

Headers:

```
Authorization: Bearer {token}
```

Request Body:

```json
{
  "name": "Updated Name",
  "email": "updated@example.com",
  "role": "staff"
}
```

Response:

```json
{
  "status": "success",
  "message": "User updated successfully",
  "data": {
    "id": 2,
    "name": "Updated Name",
    "email": "updated@example.com",
    "role": "staff",
    "created_at": "2023-05-10T10:30:00.000000Z",
    "updated_at": "2023-05-10T15:45:00.000000Z"
  }
}
```

Dashboard

Get Dashboard Stats

```
GET /api/dashboard/stats
```

Returns statistics for the admin dashboard.

Headers:

```
Authorization: Bearer {token}
```

Response:

```json
{
  "total_revenue": 24000,
  "total_bookings": 20,
  "recent_bookings": [
    {
      "id": 1,
      "user": "John Doe",
      "route": "Colombo - Kandy",
      "departure_date": "2023-06-15",
      "price": 1200,
      "created_at": "2023-05-10T10:30:00.000000Z"
    }
  ],
  "upcoming_trips": [
    {
      "id": 1,
      "route": "Colombo - Kandy",
      "departure_date": "2023-06-15",
      "booked_seats": 1,
      "available_seats": 39
    }
  ],
  "booking_stats": {
    "weekly": [120, 150, 140, 160, 170, 180, 200],
    "weekly_labels": ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
  },
  "revenue_stats": {
    "monthly": [120000, 150000, 140000, 160000, 170000, 180000],
    "monthly_labels": ["Jan", "Feb", "Mar", "Apr", "May", "Jun"]
  }
}
```

Error Responses

Validation Error

```
Status: 422 Unprocessable Entity
```

```
{
  "message": "The given data was invalid.",
  "errors": {
    "email": [
      "The email field is required."
    ],
    "password": [
      "The password field is required."
    ]
  }
}
```

Authentication Error

```
Status: 401 Unauthorized
```

```
{
  "message": "Unauthenticated."
}
```

Authorization Error

```
Status: 403 Forbidden
```

```
{
  "message": "You do not have permission to access this resource."
}
```

Resource Not Found

```
Status: 404 Not Found
```

```
{
  "message": "Resource not found."
}
```

Server Error

```
Status: 500 Internal Server Error
```

```
{
  "message": "An unexpected error occurred."
}
```