**Guvi Capstone Project**

**( Back End Documentation )**

## Bus Ticket Booking System

**BusController API Documentation**

**Endpoint: /bus/all**

**Description**

This endpoint retrieves a list of all buses.

**HTTP Method**

GET

**Request URL**

http://{base\_url}/bus/all

**Response**

* **Status Code:** 200 OK
* **Content-Type:** application/json

Response Body:

[

{

"id": 1,

"name": "Bus Name 1",

"capacity": 50,

"route": "Route 1"

},

{

"id": 2,

"name": "Bus Name 2",

"capacity": 40,

"route": "Route 2"

},

...

]

#### Response Fields

* id: The unique identifier of the bus.
* name: The name of the bus.
* capacity: The seating capacity of the bus.
* route: The route the bus is assigned to.

### Error Responses

* **Status Code:** 500 Internal Server Error
  + **Description:** If there is an error while processing the request.
  + **Response Body:**

{

"timestamp": "2024-07-09T12:34:56.789+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "An error occurred while retrieving buses.",

"path": "/bus/all"

}

### Notes

* Ensure the base URL is replaced with the actual domain or IP address of your server.
* This API does not require any authentication for access. Depending on your application's requirements, you might need to secure this endpoint.

## BookingController API Documentation

### Endpoint: /save/booking

#### Description

This endpoint saves a new booking.

#### HTTP Method

POST

#### Request URL

http://{base\_url}/save/booking

#### Request Body

* **Content-Type:** application/json
* **Body:**

[

{

"userId": 123,

"busId": 1,

"seatNumber": "A1",

"bookingDate": "2024-07-09"

},

{

"userId": 124,

"busId": 1,

"seatNumber": "A2",

"bookingDate": "2024-07-09"

}

]

#### Response

* **Status Code:** 200 OK
* **Content-Type:** application/json
* **Response Body:**

{

"bookingId": 1001,

"userId": 123,

"busId": 1,

"seatNumber": "A1",

"bookingDate": "2024-07-09",

"status": "CONFIRMED"

}

Example Response

{

"bookingId": 1001,

"userId": 123,

"busId": 1,

"seatNumber": "A1",

"bookingDate": "2024-07-09",

"status": "CONFIRMED"

}

#### Response Fields

* bookingId: The unique identifier of the booking.
* userId: The unique identifier of the user.
* busId: The unique identifier of the bus.
* seatNumber: The seat number assigned in the booking.
* bookingDate: The date of the booking.
* status: The status of the booking (e.g., "CONFIRMED").

### Error Responses

* **Status Code:** 400 Bad Request
  + **Description:** If the request body is invalid or missing required fields.
  + **Response Body:**

{

"timestamp": "2024-07-09T12:34:56.789+00:00",

"status": 400,

"error": "Bad Request",

"message": "Invalid booking request.",

"path": "/save/booking"

}

**Status Code:** 500 Internal Server Error

* **Description:** If there is an error while processing the request.
* **Response Body:**

### Notes

* Ensure the base URL is replaced with the actual domain or IP address of your server.
* This API assumes that the booking request contains a list of BookingRequest objects.
* This API does not require any authentication for access. Depending on your application's requirements, you might need to secure this endpoint.

## AuthenticationController API Documentation

### Endpoint: /api/v1/auth/register

#### Description

This endpoint registers a new user.

#### HTTP Method

POST

#### Request URL

http://{base\_url}/api/v1/auth/register

#### Request Body

* **Content-Type:** application/json
* **Body:**

{

"username": "john\_doe",

"password": "password123",

}

**Response**

* **Status Code:** 200 OK
* **Content-Type:** application/json
* **Response Body:**

{

"userId": 1,

"username": "john\_doe",

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

"token": "jwt\_token\_here"

}

**Response Fields**

* userId: The unique identifier of the authenticated user.
* username: The username of the authenticated user.
* email: The email address of the authenticated user.
* token: The JWT token for authentication.

**Error Responses**

* **Status Code:** 400 Bad Request
  + **Description:** If the request body is invalid or missing required fields.
  + **Response Body:**

{

"timestamp": "2024-07-09T12:34:56.789+00:00",

"status": 400,

"error": "Bad Request",

"message": "Invalid authentication request.",

"path": "/api/v1/auth/authenticate"

}

**Status Code:** 401 Unauthorized

* **Description:** If the authentication credentials are incorrect.
* **Response Body:**

{

"timestamp": "2024-07-09T12:34:56.789+00:00",

"status": 401,

"error": "Unauthorized",

"message": "Invalid username or password.",

"path": "/api/v1/auth/authenticate"

}

**Status Code:** 500 Internal Server Error

* **Description:** If there is an error while processing the request.
* **Response Body:**

{

"timestamp": "2024-07-09T12:34:56.789+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "An error occurred while authenticating the user.",

"path": "/api/v1/auth/authenticate"

}

### Notes

* Ensure the base URL is replaced with the actual domain or IP address of your server.
* This API assumes that the register and authentication requests contain appropriate fields.
* Depending on your application's requirements, you might need to secure these endpoints and handle JWT token management.