

BusBox List Of APIs

1. Login (Jwt Tokens)

Method : GET

Route : */auth/login*

This API will be used to Authenticate the user

it will return a accessToken(valid for 2 hours) and set a refreshToken(valid for 1 day) in a httpOnly cookie

```
//Sample Request
{
  email : "String",
  password : "String"
}
```

```
// Sample Response
{
  accessToken : "String"
}
```

2. Sign UP

Method : POST

Route : */auth/signup*

This API will be used for registering a new user

After successful registration will return the same result as login API

```
//Sample Request
{
  email : "String",
  password : "String"
}
```

```
// Sample Resposne
{
  accessToken : "String"
}
```

3. Log Out (No Content)

Method : DELETE

route : */auth/logout*

This API will be used to log out the user from current session,

it will invalidate the current accessToken user has.

```
// Sample Request
{
  email : "String",
  accessToken : "String"
}
```

```
// Sample Resposne
{
  // No Content 204
}
```

4. List Buses

Method : GET

route : /buses/list

This API will be used to List the buses based on the sourceCity, destinationCity and the Date provided by user.

Unauthenticated users can also call this API, hence it doesn't need an accessToken.

```
// Sample Request
{
  sourceCity : "String",
  destinationCity : "String",
  date : "Date"
}
```

```
// Sample Response
{
  list : [
    {
      busId : "String",
      busType : "String",
      ticketFare : "Number",
      departureTime : "Date",
      journeyTime : "Date",
      rating : {
        outOfFive : "Number",
        numberOfRatings : "Number"
      },
      amenities : ["String"]
    }
  ]
}
```

5. Get Bus Info (pickup & drop points)

Method : GET

route : /buses/:busId

This API will be used to get detailed information about a selected bus(seatLayout statuses of all the seats of bus),

Unauthenticated users can call this api, hence it doesn't need any accessToken

```
//Sample Request
{
  busId : "String",
}
```

```
// Sample Response
{
  numberOfSeats : "Number",
  seatLayout : "String",
  seatStatus : {
    "<seatNumber>" : {
      status : "String", // available/ booked/
      reservedFor : "String", // optional
      bookingId : "String"
    }
  }
}
```

6. Book Tickets

Method : POST

route : /bookings/new

This api will be used to actually book a ticket,

only authenticated users can call this api, hence an accessToken is mandatory

```
// Sample Request
{
  busId : "String",
  busType : "String",
  seatsSelected : [{seatType : "String", seatNumber : "Number"}],
  totalFare : "Number",
  sourceCity : "String",
  destinationCity : "String",
  journeyDate : "Date",
  pickUpPoint : {
    address : "String",
    departureTime : "Date",
  },
  dropPoint : {
    address : "String",
    arrivalTime : "Date"
  },
  passengerDetails : [
    {
      name : "String",
      age : "Number",
      gender : "String"
    }
  ],
  contactDetails : {
    email : "Email",
    phoneNumber : "Phone"
  }
}
```

```
// Sample Resposne
{
  bookingStatus : "String",
  bookingId : "String",
  busId : "String",
  busType : "String",
  seatsSelected : ["Number"],
  sourceCity : "String",
  destinationCity : "String",
  journeyDate : "Date",
  pickUpPoint : {
    address : "String",
    departureTime : "Date",
  },
  dropPoint : {
    address : "String",
    arrivalTime : "Date"
  },
  passengerDetails : [
    {
      name : "String",
      age : "Number",
      gender : "String"
    }
  ],
  contactDetails : {
    email : "Email",
    phoneNumber : "Phone"
  }
}
```

7. List Bookings

Method : GET

route : /booking/list

This api will be used to list all the bookings a user has made,

user must be authenticated and must send a accessToken to this api

```
// Sample Request
{
  email : "String"
}
```

```
// Sample Response
{
  list : [
    {
      bookingId : "String",
      sourceCity : "String",
      destinationCity : "String",
      journeyDate : "Date",
      busType : "String",
      bookingStatus : "String"
    }
  ]
}
```

8. Get Booking Info

Method : GET

route : /bookings/:bookingId

This api will be used to get more information about a booking.

can only be called by authenticated users, hence needs authenticationToken.

```
// Sample Request
{
  bookingId : "String"
}
```

```
// Sample Resposne
{
  bookingStatus : "String",
  bookingId : "String",
  busId : "String",
  busType : "String",
  seatsSelected : ["Number"],
  sourceCity : "String",
  destinationCity : "String",
  journeyDate : "Date",
  pickUpPoint : {
    address : "String",
    departureTime : "Date",
  },
  dropPoint : {
    address : "String",
    arrivalTime : "Date"
  },
  passengerDetails : [
    {
      name : "String",
      age : "Number",
      gender : "String"
    }
  ],
  contactDetails : {
    email : "Email",
    phoneNumber : "Phone"
  }
}
```

9. Email Ticket

Method : POST

route : /bookings/email/:bookingId

This api will be used for send the ticket to the user's email address.

user has to be authenticated with accessToken.

```
//Sample Request
{
  bookingId : "String"
}
```

```
// Sample Resposne
{
  status : "String"
}
```

10. Initiate Cancel Booking (will send otp on email)

Method : POST

route : /booking/cancel/initiate

This api will be used to initiate a booking cancellation, the booking has to be in InProgress state.

and should only be called before the departure time for a successful cancellation.

It will send an otp on users email which has to confirmed by user.

User needs to be authenticated with accessToken.

```
// Sample Request
{
  bookingId : "String"
}
```

```
// Sample Resposne
{
  status : "String"
}
```

11. Confirm Cancel Booking (will actually cancel booking, and refund the amount)

Method : POST

route : /booking/cancel

this api will be used for actually cancelling a booking. User has to send a confirmation Code

this code will be verified and then only the cancellation will succeed.

User needs to be authenticated with accessToken.

```
// Sample Request
{
  bookingId : "String",
  confirmationCode : "Number"
}
```



```
// Sample Resposne
{
  status : "String"
}
```

Bus Schema

```
//bus Schema
{
  busId : "String",
  busType : "String",
  numberOfStops : "Number",
  runningDays : [""]
  priceFromOriginToEnd : "Number",
  rating : {
    outOfFive : "Number",
    numberOfRatings : "Number"
  },
  busRoute : {
    "<cityname>" : {
      "<stopNumber>": "Number",
      departurTime : "String"
    }
  },
  aminities : ["String"],
  numberOfSeats : "Number",
  seatLayout : "String", // (2+2 / 2+1)
  seatStatus : {
    "<seatNumber>" : {
      status : {
        value : "String",
        from : "<stopNumber :: Number>",
        to : "<stopNumber :: Number>"
      },
      reservedFor : "String" // "anything like,
female/ senior citizen/ physically challenged"
    }
  }
}
```