

API Documentation of
Ticket Management System

Authentication APIs

API – User Registration

Method – **POST**

Endpoint – <http://localhost:4001/auth/register>

Payload Description –

```
{
  "fullName" : Takes full name from the user
  "userName" : Takes an unique username,
  "email" : Takes the email from the user,
  "role" : Takes role from the user; "admin"/"user",
  "password" : Takes the user password
}
```

Successful response –

```
{
  "message" : "User Created"
}
```

Unsuccessful response –

```
{
  "message" : "Must Enter all the Info!"
}
```

Sample Payload (for successful response) –

```
{
  "fullName" : "Nazmus Sakib",
  "userName" : "sak12345",
  "email" : "nazmussakib@gmail.com",
  "role" : "admin",
  "password" : "0099hello"
}
```

Sample Payload (for unsuccessful response) –

```
{
  "fullName" : "Nazmus Sakib",
  "userName" : "",
}
```

```
"email" : "nazmussakib@gmail.com",  
"role" : "",  
"password" : "0099hello"  
}
```

API – User Login

Method – **POST**

Endpoint – <http://localhost:4001/auth/login>

Payload Description –

```
{  
  "userName" : Takes username from the user,  
  "password" : Takes password from the user  
}
```

Successful response –

```
{  
  "message": "Login Successful",  
  "token":  
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3NWVmMTgyNTk5YTQ2ODY3ZTNkZGY0YiIsInJvbmUiOiJhZG1pbilzImIhdCI6MTczNDA2NDQzNiwiZXhwIjoxNzM0MDY4MDM2fQ.WFizF_fDzk4Zc  
  mwvjT5L8VKCS_2771cM6u-ZelinKeA"  
}
```

Unsuccessful response – (If username is wrong)

```
{  
  "message" : "No user found!"  
}
```

Unsuccessful response – (if password is wrong)

```
{  
  "message" : "Wrong Password"  
}
```

Sample Payload (for successful response) –

```
{
```

```
"userName" : "saif9795",  
"password" : "12345asd"  
}
```

Sample Payload (for unsuccessful response; wrong username) –

```
{  
  "userName" : "saif9790",  
  "password" : "12345asd"  
}
```

Sample Payload (for unsuccessful response; wrong password) –

```
{  
  "userName" : "saif9790",  
  "password" : "12345add"  
}
```

API – User Logout

Method – **POST**

Endpoint – <http://localhost:4001/auth/logout>

Successful response –

```
{  
  "message": "Logged out successfully"  
}
```

Unsuccessful response – (if no token found in the header)

```
{  
  "message" : "Unauthorized"  
}
```

Admin APIs

API – Adds a new bus

Method – **POST**

Endpoint – <http://localhost:4001/admin/bus>

Payload Description –

```
{
  "coachNumber" : Takes the coach number of the bus,
  "from" : Takes the starting position of the route,
  "destination" : Takes the destination of the route
  "departureDay" : Takes the departure day; value between 1 to 31,
  "departureMonth" : Takes the departure month; value between 1 to 12 ,
  "departureHour" : Takes the departure hour; value between 0 to 23,
  "departureMinute" : Takes the departure minute; value between 0 to 59,
  "fare" : Takes the fare
}
```

Successful response –

```
{
  "message": "New Bus Information Created."
}
```

Unsuccessful response – (if departure date is invalid)

```
{
  "message" : "Invalid Departure Date"
}
```

Unsuccessful response – (if departure time is invalid)

```
{
  "message" : "Invalid Departure Time"
}
```

Sample payload – (for successful response)

```
{
  "coachNumber" : "DH-101",
  "from" : "Dhaka",
```

```
"destination" : "Jessore",
"departureDay" : 15,
"departureMonth" : 12,
"departureHour" : 12,
"departureMinute" : 0,
"fare" : 500
}
```

Sample payload – (for unsuccessful response; invalid departure date)

```
{
"coachNumber" : "JS-101",
"from" : "Jessore",
"destination" : "Dhaka",
"departureDay" : 0,
"departureMonth" : 13,
"departureHour" : 12,
"departureMinute" : 0,
"fare" : 700
}
```

Sample payload – (for unsuccessful response; invalid departure time)

```
{
"coachNumber" : "CH-101",
"from" : "Dhaka",
"destination" : "Chittagang",
"departureDay" : 15,
"departureMonth" : 12,
"departureHour" : 25,
"departureMinute" : 65,
"fare" : 1200
}
```

API – Updates bus information

Method – PUT

Endpoint – <http://localhost:4001/admin/bus/:id>

Payload Description –

```
{
```

```
"key": Takes the key
"data": Takes the updated value
}
```

Successful response – (in case of updating “fare”)

```
{
  "message": "fare Updated"
}
```

Unsuccessful response – (in case of invalid key)

```
{
  "message": "No such key found."
}
```

Unsuccessful response – (in case of invalid updated value)

```
{
  "message": "Invalid Departure Day input"
}
```

Sample payload – (for successful response)

```
{
  "key": "fare",
  "data": "900"
}
```

Sample payload – (for unsuccessful response; invalid key)

```
{
  "key": "farer",
  "data": "900"
}
```

Sample payload – (for unsuccessful response; invalid updated value)

```
{
  "key": "departureDay",
  "data": "32"
}
```

API – Deletes bus information

Method – **DELETE**

Endpoint – <http://localhost:4001/admin/bus/:id>

Successful response –

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

Unsuccessful response – (if no id found)

```
{
  "message": "No Bus found"
}
```

API – Adds a ticket

Method – **POST**

Endpoint – <http://localhost:4001/admin/ticket>

Payload Description –

```
{
  "busId" : Takes Bus id from the Bus model of the database,
  "seatNumber" : Takes the seat number
}
```

Successful response –

```
{
  "message" : "New Ticket Information Created."
}
```

Unsuccessful response – (if bus id is not found)

```
{
  "message" : "No Bus found"
}
```


Unsuccessful response – (if seat already exists)

```
{  
  "message" : "Seat already exists"  
}
```

Unsuccessful response – (if invalid seat number is given)

```
{  
  "message" : "Invalid Seat "  
}
```

Unsuccessful response – (if ticket count is more than capacity)

```
{  
  "message" : "No ticket available"  
}
```

Sample payload – (for successful response)

```
{  
  "busId" : "675c1bb62e6e9a426c96392e",  
  "seatNumber" : "E1"  
}
```

Sample payload – (for unsuccessful response; invalid seat number)

```
{  
  "busId" : "675c1bb62e6e9a426c96392e",  
  "seatNumber" : "Z1"  
}
```

Sample payload – (for unsuccessful response; seat already exists)

```
{  
  "busId" : "675c1bb62e6e9a426c96392e",  
  "seatNumber" : "E1"  
}
```

API – Updates ticket information

Method – **PUT**

Endpoint – <http://localhost:4001/admin/ticket/:id>

Payload Description –

```
{  
  "busId" : Takes Bus id from the Bus model of the database,  
  "key": "Takes a key,  
  "data": Takes the updated value  
}
```

Successful response – (in case of updating “seatNumber”)

```
{  
  "message": "seatNumber Updated"  
}
```

Unsuccessful response – (when no ticket id is given)

```
{  
  "message": 'Enter Ticket Id!'  
}
```

Unsuccessful response – (in case of any of the value is not given)

```
{  
  "message": 'Enter all the information!'  
}
```

Unsuccessful response – (in case of no bus found with given bus id)

```
{  
  "message": "No Bus found"  
}
```

Unsuccessful response – (in case of no ticket found with given bus id)

```
{  
  "message": 'Ticket not found'  
}
```

Unsuccessful response – (in case of invalid key)

```
{  
  "message": "No such key found."  
}
```

Sample payload – (for successful response)

```
{
  "busId" : "675c1bb62e6e9a426c96392e",
  "key": "seatNumber",
  "data": "D1"
}
```

Sample payload – (for unsuccessful response; invalid bus id)

```
{
  "busId" : "675c1bb62e6e9a426c96392a",
  "key": "seatNumber",
  "data": "D1"
}
```

Sample payload – (for unsuccessful response; invalid key)

```
{
  "busId" : "675c1bb62e6e9a426c96392e",
  "key": "seatNumberrrr",
  "data": "D1"
}
```

API – Deletes Ticket information

Method – **DELETE**

Endpoint – <http://localhost:4001/admin/ticket/:id>

Successful response –

```
{
  "message": "Ticket deleted"
}
```

Unsuccessful response – (if no id found)

```
{
  "message": "No ticket found"
}
```

User APIs

API – View all available buses

Method – GET

Endpoint – <http://localhost:4001/buses>

Successful response –

```
{
  "busData": [
    {
      "coachNumber": "DH-101",
      "route": "Dhaka to Jessore",
      "departureDay": 15,
      "departureMonth": 12,
      "departureHour": 18,
      "departureMinute": 0,
      "fare": 500,
      "availableSeats": 12
    }
  ]
}
```

Unsuccessful response –

```
{
  "message": "No Bus available."
}
```

API – View tickets of specific bus and time

Method – GET

Endpoint – <http://localhost:4001/tickets>

Payload Description –

```
{
  "key": "Takes a key,
  "data": Takes the updated value
}
```

Successful response –

```
"data": [
  [
    {
      "_id": "675c3919ec4673870a0be790",
      "busId": "675c1bb62e6e9a426c96392e",
      "coachNumber": "DH-101",
      "seatNumber": "A1",
      "status": "booked",
      "bookingDate": "2024-12-13T13:39:37.277Z",
      "__v": 0
    },
  ]
]
```

Unsuccessful response – (when no ticket is available)

```
{
  "message" : "No ticket found."
}
```

Unsuccessful response – (in case of invalid time)

```
{
  "message" : "Invalid Time"
}
```

Unsuccessful response – (in case of invalid key)

```
{
  "message" : "Invalid key"
}
```

Sample payload – (for successful response)

```
{
  "key" : "departureHour",
  "data": 12
}
```

Sample payload – (for unsuccessful response; invalid key)

```
{
  "key" : "departureHourrrrrr",
  "data": 12
}
```

Sample payload – (for unsuccessful response; invalid key)

```
{
  "key" : "departureHour",
  "data": 25
}
```

API – Purchase tickets of specific bus and time

Method – **POST**

Endpoint – <http://localhost:4001/tickets/purchase>

Payload Description –

```
{
  "ticketId": "Takes a id of a ticket,"
}
```

Successful response –

```
{
  "message": "Ticket Purchased"
}
```

Unsuccessful response – (when no ticket is found)

```
{
  "message": "No ticket found."
}
```