

# Final Project of Backend Developer Bootcamp

Eda Özyurt

## Contents

1. Purpose of Project.....	3
1.1 Project Summary.....	3
1.2 Technologies.....	3
1.3 Requirements of Project .....	3
2. Service Documentation and JSON Examples for API Endpoints.....	4
2.1 Endpoints .....	4
2.2 DTOs (Data Transfer Objects) .....	5
2.2 JSON Examples .....	7

## 1. Purpose of Project

**Project Name** : Airline Ticket Booking System

### 1.1 Project Summary

The "Airline Ticket Booking System Backend Service Application" project addresses the needs of modern airline operations by providing a comprehensive backend solution.

### 1.2 Technologies

Technology	Version
Java	17
Spring Boot	3
MySQL	
Database Restful Web Service (JSON)	
modelmapper	3.1.1
openapi	2.1.0
lombok	
IntelliJ IDEA	

### 1.3 Requirements of Project

- **Streamlined Operations:** The project aims to facilitate efficient airline management through functionalities like adding, searching, and defining airline companies, airports, and flight routes. This ensures smoother operations for both airlines and passengers.
- **Secure Ticket Purchasing:** With a strong emphasis on security, the project implements a secure ticket purchasing process. Sensitive credit card information is masked during the transaction, accommodating various card number formats to ensure a secure and hassle-free experience.
- **Enhanced Ticket Management:** The system enables users to easily search for tickets using ticket numbers and provides the capability to cancel tickets.
- **Error Management and Consistent Responses:** The project establishes a robust global exception handling structure to handle errors effectively. API responses are structured uniformly, providing clarity and consistency to users interacting with the system.

## 2. Service Documentation and JSON Examples for API Endpoints

### 2.1 Endpoints

ticket-controller ^	
PUT	/tickets/cancelTicket={ticketNumber} v
POST	/tickets/buyTicket/flightId={flightId} v
GET	/tickets v
GET	/tickets/ticketNumber={ticketNumber} v
route-controller ^	
POST	/routes/addRoute v
GET	/routes v
GET	/routes/routeId={id} v
flight-controller ^	
POST	/flights/airlineId={airlineId}/routeId={routeId} v
GET	/flights v
airport-controller ^	
POST	/airports/addAirport v
GET	/airports v
GET	/airports/airportName={name} v
airline-controller ^	
POST	/airlines/addAirline v
GET	/airlines v
GET	/airlines/airlineName={name} v

## 2.2 DTOs (Data Transfer Objects)

```
TicketRequestDTO {  
    price          number($double)  
    passengerName  string  
    seatNumber     string  
    cc             string  
}
```

```
AirlineResponseDTO {  
    name    string  
    country string  
}
```

```
BuyTicketResponseDTO {  
    ticketNumber string  
    flight        FlightResponseDTO {  
        route  
        flightNumber  
        airline  
        departureTime  
        arrivalTime  
        capacity  
    }  
    price          number($double)  
    passengerName  string  
    seatNumber     string  
    cc             string  
}
```

```
FlightResponseDTO {  
    route  
    flightNumber  
    airline  
    departureTime  
    arrivalTime  
    capacity  
}
```

```
RouteResponseDTO {  
    departureAirport string  
    arrivalAirport   string  
}  
AirlineResponseDTO {  
    name    string  
    country string  
}
```

```
RouteResponseDTO {  
    departureAirport string  
    arrivalAirport   string  
}
```

```
RouteRequestDTO {  
    routeId      integer($int64)  
    departureAirport string  
    arrivalAirport string  
}
```

```
FlightRequestDTO {  
    departureTime string  
    arrivalTime   string  
    capacity      integer($int32)  
}
```

```
AirportRequestDTO {  
    name    string  
    city    string  
    country string  
}
```

```
AirportResponseDTO {  
    name    string  
    city    string  
    country string  
}
```

```
AirlineRequestDTO {
  airlineId integer($int64)
  name      string
  country   string
}
```

```
TicketResponseDTO {
  ticketNumber string
  flight       FlightResponseDTO {
    route
    flightNumber
    airline
    departureTime
    arrivalTime
    capacity
  }
  price          number($double)
  passengerName string
  seatNumber     string
}
```

```
RouteResponseDTO {
  departureAirport string
  arrivalAirport   string
}
string
AirlineResponseDTO {
  name      string
  country   string
}
string
string
integer($int32)
```

## 2.2 JSON Examples

### Add Airpot

**POST** /airports/addAirport

Parameters

No parameters

Request body required

application/json

```
{
  "name": "Esenboğa Havalimanı",
  "city": "Ankara",
  "country": "Türkiye"
}
```

Execute

### Response Body – Add Airport

Server response

Code

Details

200

Response body

```
{
  "name": "Esenboğa Havalimanı",
  "city": "Ankara",
  "country": "Türkiye"
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 14 Aug 2023 17:11:09 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

### Get All Airports

Server response

Code

Details

200

Response body

```
[
  {
    "name": "Sabiha Gökçen Havalimanı",
    "city": "İstanbul",
    "country": "Türkiye"
  },
  {
    "name": "Esenboğa Havalimanı",
    "city": "Ankara",
    "country": "Türkiye"
  }
]
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 14 Aug 2023 17:13:21 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Buy Ticket

POST /tickets/buyTicket/flightId={flightId}

Parameters

Name

Description

flightId required

integer (\$int64)

1

(path)

Request body required

application/json

```
{
  "price": 1500,
  "passengerName": "Ayse Yilmaz",
  "seatNumber": "1-B",
  "cc": "3216-5498-7032-1654"
}
```

Response Body – Buy Ticket

Server response

Code

Details

200

Response body

```
{
  "ticketNumber": "I2W31",
  "flight": {
    "route": {
      "departureAirport": "Sabiha Gökçen Havalimanı",
      "arrivalAirport": "Ordu/Giresun Havalimanı"
    },
    "flightNumber": "KUEAG",
    "airline": {
      "name": "Türk Hava Yolları",
      "country": "Türkiye"
    },
    "departureTime": "12.00",
    "arrivalTime": "13.15",
    "capacity": 150
  },
  "price": 1500,
  "passengerName": "Ayse Yilmaz",
  "seatNumber": "1-B",
  "cc": "321654*****1654"
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 14 Aug 2023 17:16:34 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Buy Ticket For Unavailable Seat with Same Seat Number

Server response

Code

Details

500

Undocumented

Error: response status is 500

Response body

Unavailable Seat!

Download

Response headers

```
connection: close
content-length: 17
content-type: text/plain; charset=UTF-8
date: Mon, 14 Aug 2023 17:18:22 GMT
```



## Cancel Ticket – Request

PUT

/tickets/cancelTicket={ticketNumber}

Parameters

Cancel

Name	Description
ticketNumber * required	
string	
(path)	

Execute

## Cancel Ticket – Response Body

Server response

Code

Details

200

Response body

Ticket has been canceled!

Download

Response headers

connection: keep-alive  
content-length: 25  
content-type: text/plain; charset=UTF-8  
date: Mon, 14 Aug 2023 17:30:49 GMT  
keep-alive: timeout=60

	ticket_id	available	passenger_name	price	seat_number	ticket_number	flight_id
▶	2	1	Ayşe Yılmaz	1500	1-B	I2M31	1

## Get All Tickets

Server response

Code

Details

200

Response body

```
[
  {
    "ticketNumber": "GUI55",
    "flight": {
      "route": {
        "departureAirport": "Sabiha Gökçen Havalimanı",
        "arrivalAirport": "Ordu/Giresun Havalimanı"
      },
      "flightNumber": "KUEAG",
      "airline": {
        "name": "Türk Hava Yolları",
        "country": "Türkiye"
      },
      "departureTime": "12.00",
      "arrivalTime": "13.15",
      "capacity": 150
    },
    "price": 1000,
    "passengerName": "Eda Özyurt",
    "seatNumber": "1-A"
  },
  {
    "ticketNumber": "I2M31",
    "flight": {
      "route": {
        "departureAirport": "Sabiha Gökçen Havalimanı",
        "arrivalAirport": "Ordu/Giresun Havalimanı"
      }
    }
  }
]
```

Download

## Get Ticket By Ticket Number – Request

GET

/tickets/ticketNumber={ticketNumber}

⌵

Parameters

Cancel

Name	Description
<b>ticketNumber</b> * required	
string	GUI55
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/tickets/ticketNumber=GUI55' \
  -H 'accept: */*'
```

⌵

Request URL

```
http://localhost:8080/tickets/ticketNumber=GUI55
```

## Get Ticket By Ticket Number – Response Body

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "ticketNumber": "GUI55",   "flight": {     "route": {       "departureAirport": "Sabiha Gökçen Havalimanı",       "arrivalAirport": "Ordu/Giresun Havalimanı"     },     "flightNumber": "KUEAG",     "airline": {       "name": "Türk Hava Yolları",       "country": "Türkiye"     },     "departureTime": "12.00",     "arrivalTime": "13.15",     "capacity": 150   },   "price": 1000,   "passengerName": "Eda Özyurt",   "seatNumber": "1-A" }</pre></div> <div>⌵ Download</div>

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 14 Aug 2023 17:35:03 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

## Get Ticket By Ticket Number with Unexisting Ticket Number

Request URL

```
http://localhost:8080/tickets/ticketNumber=GUI58
```

Server response

Code	Details
500	<div>Error: response status is 500</div> <div>Response body</div> <div><pre>There is no such ticket number!</pre></div> <div>⌵ Download</div>