

Adding flight when user is admin

The screenshot shows a REST client interface for a flight booking system. The request is a POST to `http://localhost:8765/flight-service/api/flight/add`. The JSON body contains flight details: fromPlace: "Delhi", toPlace: "Mumbai", departureTime: "2025-12-08T10:30:00", arrivalTime: "2025-12-08T10:00:00", availableSeats: 120, and price: { (truncated). The response is a 201 Created status with a 165 ms response time and 340 B body. The response body is "Sucessfully created".

```
POST http://localhost:8765/flight-service/api/flight/add

{
  "fromPlace": "Delhi",
  "toPlace": "Mumbai",
  "departureTime": "2025-12-08T10:30:00",
  "arrivalTime": "2025-12-08T10:00:00",
  "availableSeats": 120,
  "price": {

```

Body: 201 Created • 165 ms • 340 B • Save Response

Raw: 1 Sucessfully created

Getting all flights

The screenshot shows a REST client interface for a flight booking system. The request is a GET to `http://localhost:8765/flight-service/api/flight/airline/get`. The Auth Type is set to Bearer Token, and a token is provided. The response is a 200 OK status with a 41 ms response time and 1000 B body. The response body is a JSON array of flight details for Air India and IndiGo.

```
GET http://localhost:8765/flight-service/api/flight/airline/get

[
  {
    "id": "AI001",
    "name": "Air India",
    "logoUrl": "https://upload.wikimedia.org/airindia/logo.png",
    "flightIds": null
  },
  {
    "id": "IN002",
    "name": "IndiGo",
    "logoUrl": "https://upload.wikimedia.org/indigo/logo.png",
    "flightIds": null
  }
]
```

Body: 200 OK • 41 ms • 1000 B • Save Response

JSON: 1 [2 { 3 "id": "AI001", 4 "name": "Air India", 5 "logoUrl": "https://upload.wikimedia.org/airindia/logo.png", 6 "flightIds": null 7 }, 8 9 "id": "IN002", 10 "name": "IndiGo", 11 "logoUrl": "https://upload.wikimedia.org/indigo/logo.png", 12 "flightIds": null 13 }

Debug for flight

The screenshot shows the Postman interface for a request named "Debug for flight". The request is a GET to `http://localhost:8765/flight-service/api/flight/debug`. The "Auth" tab is selected, showing a "Bearer Token" type with a token field containing 15 dots. The response is a 200 OK status, 30 ms, 471 B. The "Raw" tab is selected, showing the response body as a JSON object: `{ "Auth": "UsernamePasswordAuthenticationToken", "Principal": "pranitha", "Credentials": "[PROTECTED]", "Authenticated": true, "Details": null, "Granted Authorities": "[ROLE_USER]" }`.

Getting all flights

The screenshot shows the Postman interface for a new request. The request is a GET to `http://localhost:8765/flight-service/api/flight/get/flights`. The "Auth" tab is selected, showing a "Bearer Token" type with a token field containing 15 dots. The response is a 200 OK status, 54 ms, 1.47 KB. The "JSON" tab is selected, showing the response body as a JSON array:

```
1 [
2   {
3     "id": "692b2874e0335bee85bc7dd9",
4     "fromPlace": "Delhi",
5     "toPlace": "Mumbai",
6     "arrivalTime": "2025-12-10T10:00:00",
7     "departureTime": "2025-12-10T07:30:00",
8     "availableSeats": 120,
9     "price": {
10      "oneWay": 3200.0,
11      "roundTrip": 6000.0
12    },
13     "airlineId": "SJ003"
14  },
15 ]
```

Flight by flight Id

The screenshot shows a REST client interface for a flight booking system. The request is a GET to `http://localhost:8765/flight-service/api/flight/get/69329ec5...`. The authentication is set to Bearer Token. The response is a 200 OK with a JSON body containing flight details.

Request:

- Method: GET
- URL: `http://localhost:8765/flight-service/api/flight/get/69329ec5...`
- Auth Type: Bearer Token

Response:

```
1 {
2   "id": "69329ec54804b49cb6a76b2e",
3   "fromPlace": "Delhi",
4   "toPlace": "Mumbai",
5   "arrivalTime": "2025-12-06T10:00:00",
6   "departureTime": "2025-12-06T10:30:00",
7   "availableSeats": 120,
8   "price": {
9     "oneWay": 3200.0,
10    "roundTrip": 6000.0
11  },
12  "airlineId": "SJ003"
13 }
```

Booking service debug

The screenshot shows a REST client interface for a flight booking system. The request is a GET to `http://localhost:8765/booking-service/api/debug`. The authentication is set to Bearer Token. The response is a 200 OK with a raw text body showing authentication details.

Request:

- Method: GET
- URL: `http://localhost:8765/booking-service/api/debug`
- Auth Type: Bearer Token

Response:

```
1 Auth: UsernamePasswordAuthenticationToken [Principal=pranithareddy,
  Credentials=[PROTECTED], Authenticated=true, Details=null, Granted
  Authorities=[ROLE_USER]]
```

Get user by id

The screenshot shows a REST client interface for a flight booking system. The request is a GET to `http://localhost:8765/user-service-auth/auth/get/69352cb8...`. The response is a 200 OK with a JSON body containing user details.

Request:

- Method: GET
- URL: `http://localhost:8765/user-service-auth/auth/get/69352cb8...`

Auth Type: Bearer Token

Token: [Redacted]

Response: 200 OK, 106 ms, 518 B

JSON Body:

```
1 {
2   "id": "69352cb85e43cd58ac3c8aa3",
3   "email": "pranitha@example.com",
4   "username": "pranitha",
5   "password":
6     "$2a$10$BbJvXmmgHR2AFZKyhFtrv0NA4YSYctu0DiqLnhWeyRCHoVo6X2psG",
7   "role": "ADMIN",
8   "age": 22,
9   "gender": "FEMALE"
}
```

Get user by email

The screenshot shows a REST client interface for a flight booking system. The request is a GET to `http://localhost:8765/user-service-auth/auth/get/email/pran...`. The response is a 200 OK with a JSON body containing user details.

Request:

- Method: GET
- URL: `http://localhost:8765/user-service-auth/auth/get/email/pran...`

Auth Type: Bearer Token

Token: [Redacted]

Response: 200 OK, 68 ms, 571 B

JSON Body:

```
1 {
2   "id": "693509438be3079da655af07",
3   "email": "pranithareddy@example.com",
4   "username": "pranithareddy",
5   "password": "$2a$10$grtj7mdhxjordNXS/
6     7zcTeFRHj4qm5fK4rUlaMf6RsBbtTBIzLvmS",
7   "role": "USER",
8   "age": 22,
9   "gender": "FEMALE"
}
```

Add Booking

A screenshot of the Postman application interface. At the top, the URL bar shows 'flightbookingsystem-jwt / Add Booking'. The request method is 'POST' and the URL is 'http://localhost:8765/booking-service/api/booking/693537e...'. The 'Body' tab is selected, showing a JSON payload:

```
{
  "email": "pranithareddy@gmail.com",
  "seatCount": 1,
  "tripType": "ROUND_TRIP",
  "mealPreference": "VEG",
  "passengerIds": [
    "693509438be3079da655af07"
  ]
}
```

. The status bar at the bottom indicates a '201 Created' response with a 255 ms duration and 397 B size. The response body shows: '1 Booking created successfully with PNR: 693537e128d4e4694162add4-251207145014'.

Trying to book the same already booked seat

A screenshot of the Postman application interface. The request is identical to the previous one, with the same JSON payload. However, the status bar at the bottom indicates a '409 Conflict' response with a 150 ms duration and 347 B size. The response body shows: '1 Seat 4B is already booked'.

Sign up as admin but email already present

A screenshot of the Postman application showing a POST request to `http://localhost:8083/auth/signup`. The request body is a JSON object with the following fields: `username: "pranitha"`, `email: "pranitha@example.com"`, `password: "secure123"`, `role: "ADMIN"`, `gender: "FEMALE"`, and `age: 22`. The response status is **400 Bad Request** with a message: `Account with this email already present`. The response time is 50 ms and the size is 353 B.

Sign up as admin

A screenshot of the Postman application showing a POST request to `http://localhost:8083/auth/signup`. The request body is a JSON object with the following fields: `username: "pranitha varakantham"`, `email: "pranithareddy123@example.com"`, `password: "secure123"`, `role: "ADMIN"`, `gender: "FEMALE"`, and `age: 22`. The response status is **200 OK** with a message: `Created account`. The response time is 180 ms and the size is 349 B.

Login when user not found

A screenshot of the Postman application showing a POST request to `http://localhost:8765/user-service-auth/auth/login`. The request body is a JSON object with the following fields: `username: "pranith"` and `password: "secure123"`. The response status is **404 Not Found** with a message: `User not found`. The response time is 40 ms and the size is 337 B.

Login when user found

flightbookingsystem-jwt / login

POST http://localhost:8765/user-service-auth/auth/login

Body

```
1 {
2   "username": "pranithareddy",
3   "password": "secure123"
4 }
```

200 OK • 253 ms • 753 B

Body

```
1 Login successful
2 eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJwcmFuaXRoYXJlZGR5IiwiaWF0IjoxNzY1MTE3ODMxLCJleHAiOjE3NjUxMjE0MzEsInJvbGVzIjoiaVVMFuIj9.
   nQlyGbFnRbyN3CJxeG7Pouv6kmHsBpCCtuzzvnXUfek
```

Sign up as User

flightbookingsystem-jwt / Signup

POST http://localhost:8083/auth/signup

Body

```
1 {
2   "username": "pranitha varakantham",
3   "email": "pranithareddy123@example.com",
4   "password": "secure123",
5   "role": "USER",
6   "gender": "FEMALE",
7   "age": 22
8 }
```

200 OK • 195 ms • 349 B

Body

```
1 Created account
```