

Adding flight when user is admin

The screenshot shows a Postman interface for a POST request to `http://localhost:8765/flight-service/api/flight/add`. The request body is a JSON object:

```
1  {
2      "fromPlace": "Delhi",
3          "toPlace": "Mumbai",
4              "departureTime": "2025-12-08T10:30:00",
5                  "arrivalTime": "2025-12-08T10:00:00",
6                      "availableSeats": 120,
7                          "price": {
```

The response is a `201 Created` status with a duration of 165 ms and a size of 340 B. The response body is: `1 Successfully created`.

Getting all flights

The screenshot shows a Postman interface for a GET request to `http://localhost:8765/flight-service/api/flight/airline/get`. The request includes an `Auth Type: Bearer Token` with a placeholder token. A note explains that the authorization header will be automatically generated.

The response is a `200 OK` status with a duration of 41 ms and a size of 1000 B. The response body is a JSON array of airline objects:

```
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 application interface. At the top, the URL is `flightbookingsystem-jwt / Debug for flight`. Below the URL, a `GET` request is selected with the endpoint `http://localhost:8765/flight-service/api/flight/debug`. A `Send` button is visible. The `Auth` tab is active, showing `Bearer Token` selected. The token field contains a redacted string. The response status is `200 OK`, with a response time of `30 ms` and a size of `471 B`. The response body is displayed in raw JSON:

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

Getting all flights

The screenshot shows the Postman application interface. At the top, the URL is `flightbookingsystem-jwt / New Request`. Below the URL, a `GET` request is selected with the endpoint `http://localhost:8765/flight-service/api/flight/get/flights`. A `Send` button is visible. The `Auth` tab is active, showing `Bearer Token` selected. The token field contains a redacted string. The response status is `200 OK`, with a response time of `54 ms` and a size of `1.47 KB`. The response body is displayed in JSON format:

```
[{"id": "692b2874e0335bee85bc7dd9", "fromPlace": "Delhi", "toPlace": "Mumbai", "arrivalTime": "2025-12-10T10:00:00", "departureTime": "2025-12-10T07:30:00", "availableSeats": 120, "price": {"oneWay": 3200.0, "roundTrip": 6000.0}, "airlineId": "SJ003"}]
```

Flight by flight Id

The screenshot shows a Postman request for the endpoint `http://localhost:8765/flight-service/api/flight/get/69329ec5...`. The request method is `GET`. The response status is `200 OK` with a response time of `66 ms` and a size of `548 B`. The response body is a JSON object representing a flight:

```
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 Postman request for the endpoint `http://localhost:8765/booking-service/api/debug`. The request method is `GET`. The response status is `200 OK` with a response time of `50 ms` and a size of `476 B`. The response body is a JSON object representing an authentication token:

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

Get user by id

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8765/user-service-auth/auth/get/69352cb8...
- Auth Type:** Bearer Token
- Response Status:** 200 OK
- Body (JSON Preview):**

```
1 {
2     "id": "69352cb85e43cd58ac3c8aa3",
3     "email": "pranitha@example.com",
4     "username": "pranitha",
5     "password": "$2a$10$BbJvXmmgHR2AFZKyhFtrv0NA4YSYCTuODiqLnhWeyRCh0Vo6X2psG",
6     "role": "ADMIN",
7     "age": 22,
8     "gender": "FEMALE"
9 }
```

Get user by email

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8765/user-service-auth/auth/get/email/pran...
- Auth Type:** Bearer Token
- Response Status:** 200 OK
- Body (JSON Preview):**

```
1 {
2     "id": "693509438be3079da655af07",
3     "email": "pranithareddy@example.com",
4     "username": "pranithareddy",
5     "password": "$2a$10$grtj7mdhxjordNXS/_7zcTeFRHj4qm5fK4iUlaMf6RsBbtTBIZLvmS",
6     "role": "USER",
7     "age": 22,
8     "gender": "FEMALE"
9 }
```

Add Booking

The screenshot shows a POST request to `http://localhost:8765/booking-service/api/booking/693537e...`. The request body is a JSON object:

```
1 {  
2   "email": "pranithareddy@gmail.com",  
3   "seatCount": 1,  
4   "tripType": "ROUND_TRIP",  
5   "mealPrefernce": "VEG",  
6   "passengerIds": [  
7     "693509438be3079da655af07"  
]
```

The response is a **201 Created** status with a PNR of `693537e128d4e4694162add4-251207145014`.

Trying to book the same already booked seat

The screenshot shows a POST request to `http://localhost:8765/booking-service/api/booking/693537e...`. The request body is identical to the previous one.

The response is a **409 Conflict** status with the message `Seat 4B is already booked`.

Sign up as admin but email already present

The screenshot shows a Postman request to `http://localhost:8083/auth/signup` using a POST method. The request body is a JSON object with fields: `username: "pranitha"`, `email: "pranitha@example.com"`, `password: "secure123"`, `role: "ADMIN"`, `gender: "FEMALE"`, and `age: 22`. The response status is `400 Bad Request` with the message `Account with this email already present`.

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

Sign up as admin

The screenshot shows a Postman request to `http://localhost:8083/auth/signup` using a POST method. The request body is a JSON object with fields: `username: "pranitha varakantham"`, `email: "pranithareddy123@example.com"`, `password: "secure123"`, `role: "ADMIN"`, `gender: "FEMALE"`, and `age: 22`. The response status is `200 OK` with the message `Created account`.

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

Login when user not found

The screenshot shows a Postman request to `http://localhost:8765/user-service-auth/auth/login` using a POST method. The request body is a JSON object with fields: `username: "pranith"` and `password: "secure123"`. The response status is `404 Not Found` with the message `User not found`.

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

Login when user found

The screenshot shows a Postman request to `http://localhost:8765/user-service-auth/auth/login`. The request method is POST. The body contains the following JSON:

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

The response status is 200 OK, with a response time of 253 ms and a response size of 753 B. The response body is:

```
1 Login successful
2 eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJwcmFuaXRoYXJlZGR5IiwiaWF0IjoxNzY1MTE3ODMxLCJleHAiOjE3NjUxMjE0MzEsInJvbGVzIjoiVVNFUiJ9.
nQlyGbFnRbyN3CJxeG7Pouv6kmHsBpCCtuzzvnXufek
```

Sign up as User

The screenshot shows a Postman request to `http://localhost:8083/auth/signup`. The request method is POST. The body contains the following JSON:

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

The response status is 200 OK, with a response time of 195 ms and a response size of 349 B. The response body is:

```
1 Created account
```