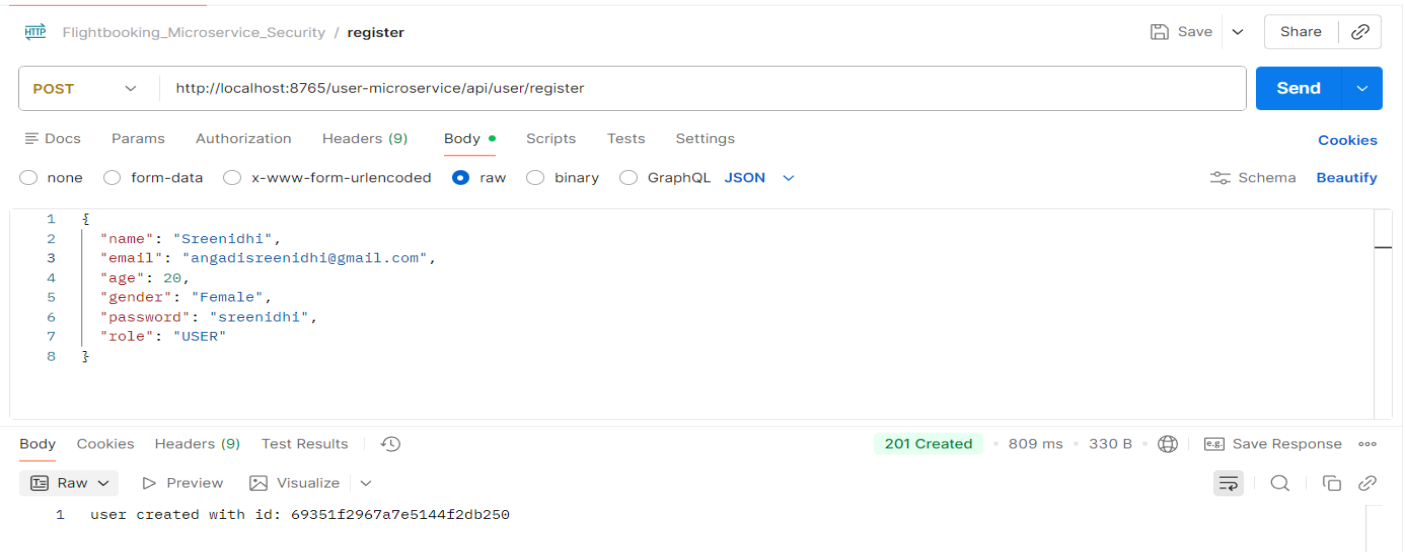


Postman implementation after Api gateway Security and Dockerization

1)user register



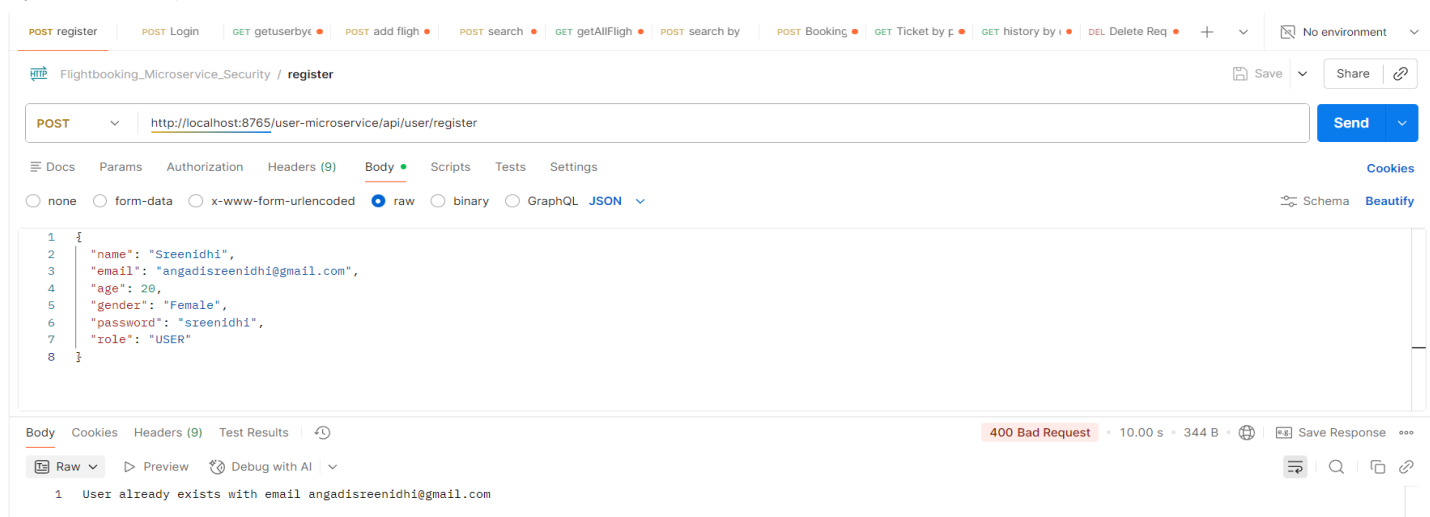
The screenshot shows a Postman client interface for a POST request to `http://localhost:8765/user-microservice/api/user/register`. The request body is a JSON object with user details. The response is a 201 Created status with a message indicating the user was created with a specific ID.

```
1 {
2   "name": "Sreenidhi",
3   "email": "angadisreenidhi@gmail.com",
4   "age": 20,
5   "gender": "Female",
6   "password": "sreenidhi",
7   "role": "USER"
8 }
```

201 Created • 809 ms • 330 B • Save Response

1 user created with id: 69351f2967a7e5144f2db250

2)user already exists validation



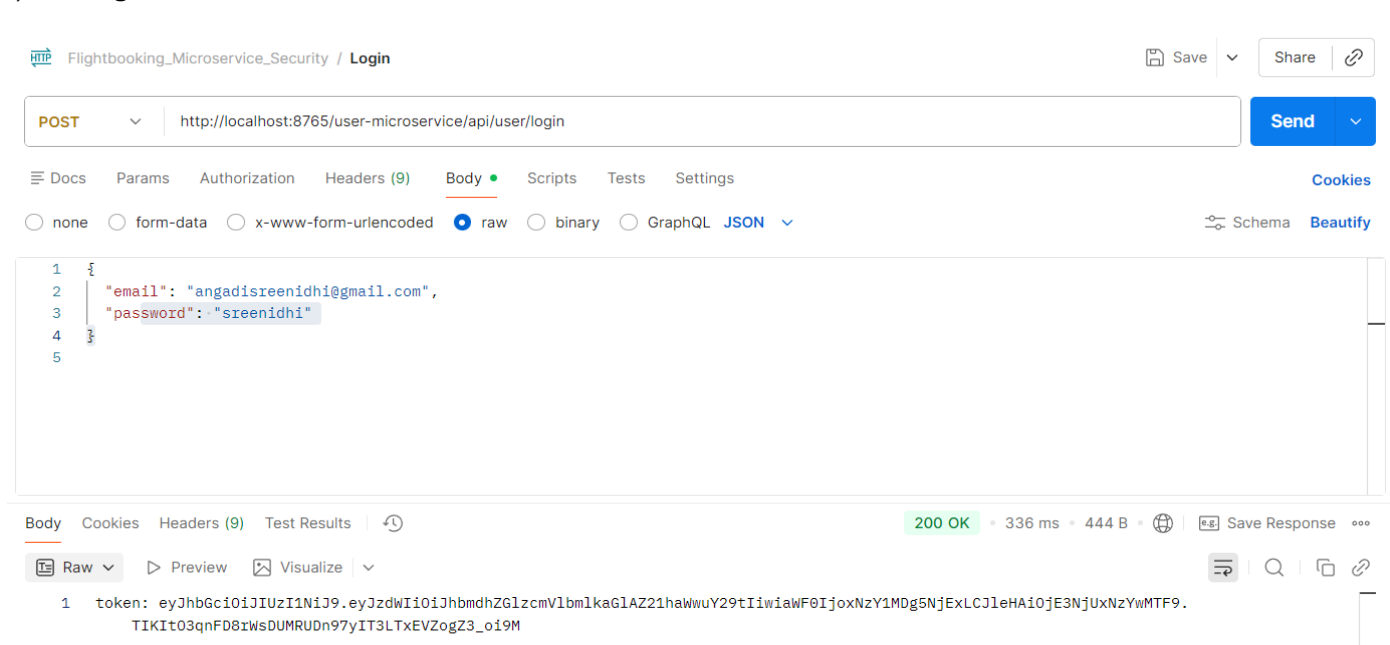
The screenshot shows a Postman client interface for a POST request to `http://localhost:8765/user-microservice/api/user/register`. The request body is the same JSON object as in the first screenshot. The response is a 400 Bad Request status with a message indicating the user already exists with the provided email.

```
1 {
2   "name": "Sreenidhi",
3   "email": "angadisreenidhi@gmail.com",
4   "age": 20,
5   "gender": "Female",
6   "password": "sreenidhi",
7   "role": "USER"
8 }
```

400 Bad Request • 10.00 s • 344 B • Save Response

1 User already exists with email angadisreenidhi@gmail.com

3)User login



The screenshot shows a Postman client interface for a POST request to `http://localhost:8765/user-microservice/api/user/login`. The request body is a JSON object with email and password. The response is a 200 OK status with a token.

```
1 {
2   "email": "angadisreenidhi@gmail.com",
3   "password": "sreenidhi"
4 }
5
```

200 OK • 336 ms • 444 B • Save Response

1 token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhbmRmdmZ6LzcmVlbnlkaGlAZ21haWwY29tIiwiaWF0IjoxNzY1MDg5NjExLCJleHAiOjE3NjUxNzYwMTF9.TIKIt03qnFD8rwsDUMRUDn97yIT3LTxEVZogZ3_o19M

4) get user by email

The screenshot shows the Postman interface for the endpoint `http://localhost:8765/user-microservice/api/user/angadisreenidhi@gmail.com` with the method `GET`. The `Authorization` tab is selected, showing `Auth Type` as `Bearer Token`. The token field contains a masked token. The `Body` tab is also visible, showing a JSON response:

```
1 {
2   "id": "69351f2967a7e5144f2db250",
3   "name": "Sreenidhi",
4   "gender": "Female",
5   "age": 20,
6   "password": "$2a$10$JuZrqBfuJEkA8r5mqZDMA.H.lpXxahFcFchg244AESsAJhyoVSpIe",
7   "email": "angadisreenidhi@gmail.com",
8   "role": "USER"
9 }
```

The response status is `200 OK` with a response time of `1.28 s` and a size of `475 B`.

5) Adding inventory without authorization (showing one example)

The screenshot shows the Postman interface for the endpoint `http://localhost:8765/flight-microservice/api/flight/airline/inventory/add` with the method `POST`. The `Body` tab is selected, showing a JSON request:

```
1 {
2   "airline": "AirAsia",
3   "fromPlace": "Mumbai",
4   "toPlace": "Hyderabad",
5   "departureTime": "2025-12-04T06:00:00",
6   "arrivalTime": "2025-12-04T09:00:00",
7   "price": 4000,
8   "totalSeats": 160,
9   "availableSeats": 160
10 }
11
```

The response status is `401 Unauthorized` with a response time of `180 ms` and a size of `274 B`.

6) adding inventory with authorization

The screenshot shows the Postman interface for the endpoint `http://localhost:8765/flight-microservice/api/flight/airline/inventory/add` with the method `POST`. The `Authorization` tab is selected, showing `Auth Type` as `Bearer Token`. The token field contains a masked token. The `Body` tab is also visible, showing a raw text response:

```
1 Flight added successfully with id: 69352313ace9642c67553a1d
```

The response status is `201 Created` with a response time of `2.18 s` and a size of `343 B`.

7)Search flight

Flightbooking_Microservice_Security / search

POST http://localhost:8765/flight-microservice/api/flight/search

Send

Docs Params Authorization Headers (9) Body Scripts Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Schema Beautify

```
1 {
2   {
3     "fromPlace": "Mumbai",
4     "toPlace": "Hyderabad",
5     "startTime": "2025-11-28T00:00:00",
6     "endTime": "2025-11-28T23:59:59"
7   }
8 }
9
```

Body Cookies Headers (9) Test Results 200 OK • 423 ms • 504 B Save Response

{ } JSON Preview Visualize

```
1 [
2   {
3     "id": "69310ee6e5b8ddb69f9ec145",
4     "airline": "AirAsia",
5     "fromPlace": "Mumbai",
6     "toPlace": "Hyderabad",
7     "departureTime": "2025-11-28T06:00:00",
8     "arrivalTime": "2025-11-28T09:00:00",
9     "price": 4800.0,
10    "totalSeats": 160,
11    "availableSeats": 158
12  }
13 ]
```

8)get all flights

Flightbooking_Microservice_Security / getAllFlights

GET http://localhost:8765/flight-microservice/api/flight/getallflights

Send

Docs Params Authorization Headers (9) Body Scripts Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Text


```
1 {
2   {
3     "email": "angadisreenidhi@gmail.com",
4     "password": "sreenidhi"
5   }
6 }
```

Body Cookies Headers (9) Test Results 200 OK • 55 ms • 1.37 KB Save Response

{ } JSON Preview Visualize

```
1 [
2   {
3     "id": "692b3a4a6444e2b421e44d2b",
4     "airline": "AirAsia",
5     "fromPlace": "Banglore",
6     "toPlace": "Hyderabad",
7     "departureTime": "2025-11-28T06:00:00",
8     "arrivalTime": "2025-11-28T09:00:00",
9     "price": 4800.0,
10    "totalSeats": 160,
11    "availableSeats": 141
12  },
13 ]
```

9)Search by airline

 Flightbooking_Microservice_Security / **search by airline**

POST

http://localhost:8765/flight-microservice/api/flight/search/airline

Send

Docs

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Schema

Beautify




```
1 {
2   "fromPlace": "Mumbai",
3   "toPlace": "Hyderabad",
4   "airline": "AirAsia"
5 }
```

Body

Cookies

Headers (9)

Test Results

200 OK • 71 ms • 952 B •   Save Response • 


{ } JSON

Preview

Visualize

```
3   "id": "69310ee6e5b8ddb69f9ec145",
4   "airline": "AirAsia",
5   "fromPlace": "Mumbai",
6   "toPlace": "Hyderabad",
7   "departureTime": "2025-11-28T06:00:00",
8   "arrivalTime": "2025-11-28T09:00:00",
9   "price": 4800.0,
10  "totalSeats": 160,
11  "availableSeats": 158
12 },
13 {
14   "id": "69314b89ec05baf4c110c11a",
15   "airline": "AirAsia",
16   "fromPlace": "Mumbai",
17   "toPlace": "Hyderabad",
18   "departureTime": "2025-12-04T06:00:00",
19   "arrivalTime": "2025-12-04T09:00:00",
20   "price": 4800.0,
21   "totalSeats": 160,
```

10)Booking

 Flightbooking_Microservice_Security / **Booking**

POST

http://localhost:8765/booking-microservice/api/flight/booking/69314b89ec05baf4c110c11a

Send

Docs

Params

Authorization

Headers (10)

Body

Scripts

Tests

Settings

Cookies

Auth Type

Bearer Token

Token

.....




The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body

Cookies

Headers (9)

Test Results

200 OK • 5.10 s • 292 B •   Save Response • 

Raw

Preview

Visualize

```
1 PNR: dbeac786
```

11) Ticket by PNR

Flightbooking_Microservice_Security / Ticket by pnr

GET http://localhost:8765/booking-microservice/api/flight/ticket/dbeac786

Auth Type: Bearer Token

Token:

The authorization header will be automatically generated when you send the

Body: 200 OK • 1.27 s • 748 B

```
{
  "id": "69353007d5c49615a6429808",
  "pnr": "dbeac786",
  "userId": null,
  "departureFlightId": "69314b89ec05baf4c110c11a",
  "returnFlightId": null,
  "tripType": "ONE_WAY",
  "bookingTime": "2025-12-07T13:13:03.506",
  "seatsBooked": "12A",
  "userEmail": "angadisreenidhi@gmail.com",
  "mealType": null,
  "totalPrice": 4800.0,
  "canceled": true,
  "passengers": [
    {
      "id": "69353008d5c49615a6429809",
      "name": "Sreenidhi",
      "gender": null,
      "age": 20,
      "seatNumber": "12A",
      "mealPreference": null,
    }
  ]
}
```

12) get booking history

Flightbooking_Microservice_Security / history by email

GET http://localhost:8765/booking-microservice/api/flight/booking/history/angadisreenidhi@gmail.com

Auth Type: Be...

Token:

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body: 200 OK • 104 ms • 6 KB

```
{
  {
    "id": "692c06db78a5e26842e5da35",
    "pnr": "d66c32a6",
    "userId": null,
    "departureFlightId": "692b3a4a6444e2b421e44d2b",
    "returnFlightId": null,
    "tripType": "ONE_WAY",
    "bookingTime": "2025-11-30T14:26:59.446",
    "seatsBooked": "12A",
    "userEmail": "angadisreenidhi@gmail.com",
    "mealType": null,
    "totalPrice": 4800.0,
    "canceled": false,
    "passengers": null
  },
  {
    "id": "692c3a017d98e412aefb2478",
    "pnr": "71c85860",
  }
}
```

13)Cancel Ticket

Flightbooking_Microservice_Security / Delete Request

SaveShare

DELETEhttp://localhost:8082/api/flight/booking/cancel/dbeac786

Send

DocsParamsAuthorizationHeaders (8)BodyScriptsTestsSettings

Cookies

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

BodyCookiesHeaders (9)Test Results

200 OK530 ms301 BSave Response

RawPreviewVisualize

1Cancelled Successfully

14)cancel ticket validation

Flightbooking_Microservice_Security / Delete Request

SaveShare

DELETEhttp://localhost:8765/booking-microservice/api/flight/booking/cancel/9cb8b16a

Send

DocsParamsAuthorizationHeaders (8)BodyScriptsTestsSettings

Cookies

Auth Type

Be...

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

BodyCookiesHeaders (9)Test Results

200 OK2.42 s328 BSave Response

RawPreviewVisualize

1Cannot cancel ticket within 24 hours of departure