

## API Gateway with security

Actuator:

The screenshot shows the Postman interface with a collection named "My Workspace". A specific request is selected: "GET check health" from the "security-flight-microservices" collection. The request URL is `http://localhost:9090/actuator/health`. The response status is 200 OK, with a response time of 258 ms and a size of 1.39 KB. The response body is a JSON object representing the健康检查结果 (check health result). The JSON structure includes fields like "status": "UP", "components": {}, and "discoveryComposite": {}.

```
1 {
2   "status": "UP",
3   "components": {
4     "clientConfigServer": {
5       "status": "UP",
6       "details": {
7         "propertySources": [
8           "configserver:https://github.com/yme754/flight-config-server/flight-api-gateway.properties",
9           "configClient"
10      ]
11    }
12  },
13  "discoveryComposite": {
14    "status": "UP",
15    "components": {
16      "discoveryClient": {
17        "status": "UP",
18        "details": {
19          "services": [
20            "booking-service"
21          ]
22        }
23      }
24    }
25  }
26}
```

Signup as admin:

The screenshot shows the Postman interface with a collection named "My Workspace". A specific request is selected: "POST signup as admin" from the "security-flight-microservices" collection. The request URL is `http://localhost:9090/api/auth/signup`. The request method is POST. The response status is 200 OK, with a response time of 287 ms and a size of 315 B. The response body is a JSON object with a message: "Admin registered successfully!".

```
1 {
2   "message": "Admin registered successfully!"
3 }
```

## Signup as admin (with already registered email):

The screenshot shows the Postman interface with a collection named "security-flight-microservices". A POST request is made to `http://localhost:9090/api/auth/signup`. The request body is:

```
1 {
2   "username": "admin_user1",
3   "email": "admin@flight.com",
4   "password": "password123",
5   "roles": ["admin"]
6 }
```

The response status is **400 Bad Request** with a message: "Error: Email is already in use!"

## Signup as user:

The screenshot shows the Postman interface with the same collection. A POST request is made to `http://localhost:9090/api/auth/signup`. The request body is:

```
1 {
2   "username": "normal_user2",
3   "email": "user@flight.com",
4   "password": "password123",
5   "roles": ["user"]
6 }
```

The response status is **200 OK** with a message: "User registered successfully!"

## Signup as user(with already registered user name):

The screenshot shows the Postman interface with a collection named "security-flight-microservices". A POST request is made to `http://localhost:9090/api/auth/signup` with the following raw JSON body:

```
1 {
2   "username": "normal_user2",
3   "email": "user6@flight.com",
4   "password": "password123",
5   "roles": ["user"]
6 }
```

The response status is 400 Bad Request, with the message: "Error: Username is already taken!"

## Signin as admin:

The screenshot shows the Postman interface with the same collection. A POST request is made to `http://localhost:9090/api/auth/signin` with the following raw JSON body:

```
1 {
2   "username": "admin_user5",
3   "password": "password123"
4 }
```

The response status is 200 OK, returning a JSON object containing user information and a JWT token:

```
1 {
2   "id": "6936a2182312f7ae9805d143",
3   "username": "admin_user5",
4   "email": "admin5@flight.com",
5   "roles": [
6     "ROLE_ADMIN"
7   ],
8   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbW1cL2VyNzIsInJvbGViIjpbIlJPTEVfQURNSU4iXSwiaWF0IjoxNzY1MTg5Nzc2LCJleHAiOjE3NjUyNzYxNzZ9.GtBRXuq0GZBeRuTjoC-y67A04P2AtDq0S4PL423Qxo",
9   "type": "Bearer"
10 }
```

## Signin as user:

The screenshot shows the Postman interface with a collection named "security-flight-microservices". A POST request is made to `http://localhost:9090/api/auth/signin` with the following body:

```
1 {
2   "username": "normal_user2",
3   "password": "password123"
4 }
```

The response is a 200 OK status with a JSON payload containing a token and user details:

```
1 {
2   "id": "6936b79d24ce549c4c6f1488",
3   "username": "normal_user2",
4   "email": "user5@flight.com",
5   "roles": [
6     "ROLE_USER"
7   ],
8   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJu3j3tYwxfXNlcjIiLCJyb2xlcI6WyJST0xFX1VTRViiXSwiaWF0IjoxNzY1MTkzOTg0LCJleHAiOjE3NjUyODA2ODR9.5GMXTijwFSl1oiHWPsdKOhm_ujF32E_b0kjCNFs1Eyg",
9   "type": "Bearer"
10 }
```

## Verify the generated token by get flights:

The screenshot shows the Postman interface with the same collection. A GET request is made to `http://localhost:9090/api/flight/flights` using the "Bearer Token" auth type. The token is pasted into the "Token" field.

The response is a 200 OK status with a JSON array of flight objects:

```
1 [
2   {
3     "id": "6932a50ffb27eee2f27397b8",
4     "fromPlace": "BLR",
5     "toPlace": "DEL",
6     "arrivalTime": "2025-12-10T10:00:01",
7     "departureTime": "2025-12-10T08:00:00",
8     "availableSeats": 50,
9     "price": {
10       "oneWay": 5000.0,
11       "roundTrip": 9000.0
12     },
13     "airlineId": "AL123",
14     "flightNumber": "FL123"
15   },
16   {
17     "id": "6935118751cdfc5e6328b8eb",
18     "fromPlace": "HYD",
19     "toPlace": "DEL",
20     "arrivalTime": "2025-12-25T14:00:00",
21     "departureTime": "2025-12-25T10:00:00"
22   }
23 ]
```

## Add airline(as admin):

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** http://localhost:9090/api/flight/airline/add
- Method:** POST
- Headers:** (9)
- Body:** raw JSON

```
1 {
2   "name": "XYZ Airlines",
3   "logoUrl": "https://example.com/logo.png",
4   "flightIds": []
5 }
```

- Test Results:** 201 Created, 75 ms, 388 B

## Add airline(as admin - airline duplicate restriction):

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** http://localhost:9090/api/flight/airline/add
- Method:** POST
- Headers:** (9)
- Body:** raw JSON

```
1 {
2   "name": "XYZ Airlines",
3   "logoUrl": "https://example.com/logo.png",
4   "flightIds": []
5 }
```

- Test Results:** 409 Conflict, 46 ms, 338 B
- Body:** JSON

```
1 {
2   "message": "Airline with name 'XYZ Airlines' already exists"
3 }
```

## Add airline(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > security-flight-microservices
- Request Type:** POST
- URL:** http://localhost:9090/api/flight/airline/add
- Authorization:** Bearer Token (Token field is populated with a token)
- Body:** Raw (empty)
- Response Status:** 401 Unauthorized
- Response Body:** 1

## Get all airlines(both admin and user are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > security-flight-microservices
- Request Type:** GET
- URL:** http://localhost:9090/api/flight/airline/get
- Authorization:** Bearer Token (Token field is populated with a token)
- Body:** JSON (empty)
- Response Status:** 200 OK
- Response Body:** A JSON array of three airline objects, each with fields: id, name, logoUrl, and flightIds.

```
1 [  
2 {  
3   "id": "69350efb51cdfc5e6328b8e9",  
4   "name": "ABC Airlines",  
5   "logoUrl": "https://example.com/logo.png",  
6   "flightIds": []  
7 },  
8 {  
9   "id": "693510fe51cdfc5e6328b8ea",  
10  "name": "ABC Airlines",  
11  "logoUrl": "https://example.com/logo.png",  
12  "flightIds": []  
13 },  
14 {  
15   "id": "6935130651cdfc5e6328b8fc",  
16   "name": "ABC Airlines",  
17   "logoUrl": "https://example.com/logo.png",  
18   "flightIds": []  
19 },  
20 ]
```

## Get airline by ID(both admin and user are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** GET
- URL:** <http://localhost:9090/api/flight/airline/get/69350efb51cd5e6328b8e9>
- Authorization:** Bearer Token (Token field is filled with a long token)
- Body:** JSON response (200 OK) containing the following JSON:

```
1 {
2   "id": "69350efb51cd5e6328b8e9",
3   "name": "ABC Airlines",
4   "logoUrl": "https://example.com/logo.png",
5   "flightIds": []
6 }
```

## Add flight(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** POST
- URL:** <http://localhost:9090/api/flight/flights/add>
- Authorization:** Bearer Token (Token field is filled with a long token)
- Body:** Raw JSON (201 Created) containing the following JSON:

```
1 {
2   "fromPlace": "HYD",
3   "toPlace": "DEL",
4   "arrivalTime": "2025-12-25T14:00:00",
5   "departureTime": "2025-12-25T10:00:00",
6   "availableSeats": 100,
7   "price": {
8     "oneWay": 5000,
9     "roundTrip": 9000
10   }
11 }
```

## Add flight(as admin - airline duplicate restriction):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Method:** POST
- Request URL:** http://localhost:9090/api/flight/flights/add
- Request Body (raw JSON):**

```
        ...
4   "arrivalTime": "2025-12-25T14:00:00",
5   "departureTime": "2025-12-25T10:00:00",
6   "availableSeats": 100,
7   "price": {
8     "oneWay": 5000,
9     "roundTrip": 9000
10 },
11  "airlineId": "6935238a8c951a7f21103",
12  "flightNumber": "SH-6"
13 }
```
- Response Status:** 409 Conflict
- Response Body:**

```
1 {
2   "message": "Flight with number SH-6 already exists"
3 }
```

## Add flight(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Method:** POST
- Request URL:** http://localhost:9090/api/flight/flights/add
- Authorization Type:** Bearer Token
- Request Body (Raw JSON):**

```
1
```
- Response Status:** 401 Unauthorized

## Get all flights(both admin and user are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > security-flight-microservices
- Request:** GET /api/flight/flights
- Authorization:** Bearer Token (Token value is masked)
- Response:** 200 OK (64 ms, 2.88 KB)  
The response body is a JSON array of flight objects, with one object partially visible:

```
[{"id": "6932a50ff27eee2f27397b8", "fromPlace": "BLR", "toPlace": "DEL", "arrivalTime": "2025-12-10T10:00:01", "departureTime": "2025-12-10T08:00:00", "availableSeats": 50, "price": {"oneWay": 5000.0, "roundTrip": 9000.0}, "airlineId": "AL123", "flightNumber": "FL123"}, {"id": "6935118751cd5c5e6328b8eb"}]
```

## Get flight by ID(both admin and user are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > security-flight-microservices
- Request:** GET /api/flight/flights/693523ec8e98c951a7f21104
- Authorization:** Bearer Token (Token value is masked)
- Response:** 200 OK (49 ms, 541 B)  
The response body is a JSON object of a flight, with one object partially visible:

```
{ "id": "693523ec8e98c951a7f21104", "fromPlace": "GUR", "toPlace": "BLR", "arrivalTime": "2025-12-25T14:00:00", "departureTime": "2025-12-25T10:00:00", "availableSeats": 98, "price": {"oneWay": 5500.0, "roundTrip": 9000.0}, "airlineId": "6935130651cd5c5e6328b8hh", "flightNumber": "SH-101"}
```

## Update flight(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Method:** PUT
- Request URL:** <http://localhost:9090/api/flight/flights/693523ec8e98c951a7f21104>
- Body (raw JSON):**

```
1 {
2   "fromPlace": "GUR",
3   "toPlace": "BLR",
4   "arrivalTime": "2025-12-25T14:00:00",
5   "departureTime": "2025-12-25T10:00:00",
6   "availableSeats": 100,
7   "price": {
8     "oneWay": 5500,
9     "roundTrip": 9000
10 },
11   "airlineId": "6935130651cdfc5e6328b8hh",
12   "flightNumber": "SH-101"
```
- Response Status:** 200 OK
- Response Time:** 52 ms
- Response Size:** 238 B

## Update flight(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Method:** PUT
- Request URL:** <http://localhost:9090/api/flight/flights/693523ec8e98c951a7f21104>
- Authorization Type:** Bearer Token
- Authorization Token:** A placeholder token is shown in the input field.
- Response Status:** 401 Unauthorized
- Response Time:** 23 ms
- Response Size:** 48 B

search flights by source, destination(both admin and user are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** POST
- URL:** <http://localhost:9090/api/flight/flights/search>
- Body (raw JSON):**

```
1 "from": "HYD",
2 "to": "DEL"
```
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 [
2   {
3     "id": "6935118751cd5c5e6328b8eb",
4     "fromPlace": "HYD",
5     "toPlace": "DEL",
6     "arrivalTime": "2025-12-25T14:00:00",
7     "departureTime": "2025-12-25T10:00:00",
8     "availableSeats": 100,
9     "price": {
10       "oneWay": 5000.0,
11       "roundTrip": 9000.0
12     },
13     "airlineId": "693510fe51cd5c5e6328b8ea",
14     "flightNumber": "SH-101"
15   },
16   {
17     "id": "6935130951cd5c5e6328b8fd",
18     "fromPlace": "HYD",
19     "toPlace": "DEL".
20 }
```

Get seats(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** GET
- URL:** <http://localhost:9090/api/flight/seats/693523ec8e98c951a7f21104>
- Authorization:** Bearer Token
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 [
2   {
3     "id": "693546c869b7dc7f561e1f1b",
4     "seatNumber": "1A",
5     "available": false,
6     "flightId": "693523ec8e98c951a7f21104"
7   },
8   {
9     "id": "693546c869b7dc7f561e1f1c",
10    "seatNumber": "1B",
11    "available": false,
12    "flightId": "693523ec8e98c951a7f21104"
13  },
14  {
15    "id": "6935482369b7dc7f561e1f36",
16    "seatNumber": "1F",
17    "available": false.
18 }
```

## Get seats(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET http://localhost:9090/api/flight/seats/693523ec8e98c951a7f21104
- Authorization:** Bearer Token (Token field contains a redacted token)
- Status:** 401 Unauthorized
- Body:** Raw JSON response: [ { "seatNumber": "1E", "available": true }, { "seatNumber": "1F", "available": true } ]

## Add seats(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** POST http://localhost:9090/api/flight/seats/693546c869b7dc7f561e1f1c/add
- Authorization:** Bearer Token (Token field contains a redacted token)
- Body:** Raw JSON data: [ { "seatNumber": "1E", "available": true }, { "seatNumber": "1F", "available": true } ]
- Status:** 200 OK
- Body:** Raw JSON response: [ { "seatNumber": "1E", "available": true }, { "seatNumber": "1F", "available": true } ]

## Add seats(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** POST
- URL:** http://localhost:9090/api/flight/seats/693546c869b7dc7f561e1f1c/add
- Authorization:** Bearer Token (Token field is empty)
- Body:** Raw JSON (empty)
- Response Status:** 401 Unauthorized

## Update seats(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** PUT
- URL:** http://localhost:9090/api/flight/seats/693546c869b7dc7f561e1f1c/update
- Authorization:** Bearer Token (Token field is empty)
- Body:** Raw JSON (contains seat update data)
- Response Status:** 200 OK

```
1 [ 2   { 3     "seatNumber": "1F", 4     "available": true, 5     "flightId": "693546c869b7dc7f561e1f1c" 6   }, 7   { 8     "seatNumber": "1E", 9     "available": true, 10    "flightId": "693546c869b7dc7f561e1f1c" 11   } 12 ]
```

## Update seats(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** PUT
- URL:** http://localhost:9090/api/flight/seats/693546c869b7dc7f561e1fc/update
- Authorization:** Bearer Token (Token field is filled with a long string of asterisks)
- Body:** Raw JSON (Content: 1)
- Status:** 401 Unauthorized

## Book flight(both user and admin are allowed):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request Type:** POST
- URL:** http://localhost:9090/api/flight/bookings/book
- Authorization:** Bearer Token (Token field is filled with a long string of asterisks)
- Body:** Raw JSON (Content: 1)
- Status:** 201 Created

## Get bookings(as admin):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request:** GET /api/flight/bookings
- Authorization:** Bearer Token (with a placeholder token)
- Response Status:** 200 OK
- Response Body (JSON):**

```
[{"id": "14d4c753-860e-442a-830c-997440ae434a", "email": "vaswithasakhamuri05@gmail.com", "flightId": "6935130951cdcf5e6328b8fd", "seatCount": 2, "seatNumbers": ["1A", "1B"], "pnr": "PNR-1404C7", "totalAmount": 0.0, "passengerIds": ["P1", "P2"]}]
```

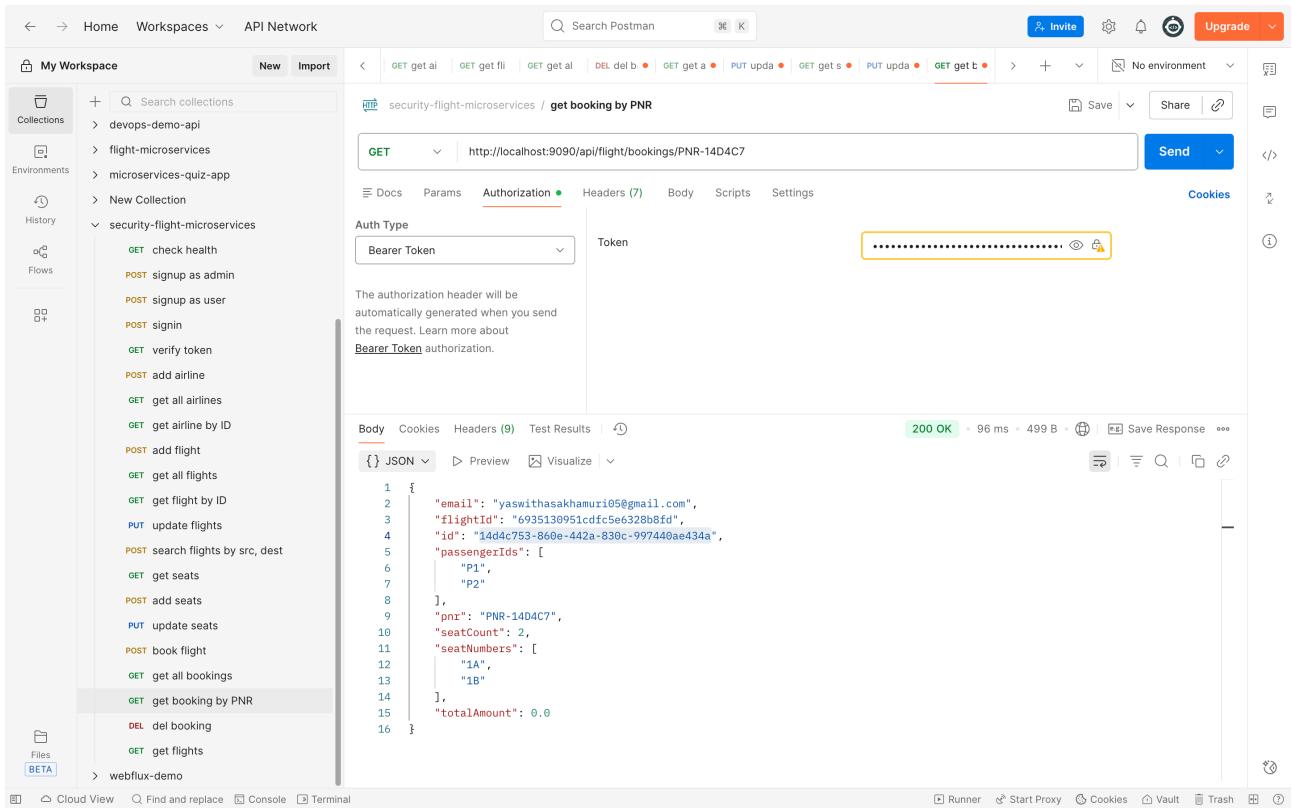
## Get bookings(as user - restricted):

The screenshot shows the Postman interface with the following details:

- Collection:** security-flight-microservices
- Request:** GET /api/flight/bookings
- Authorization:** Bearer Token (with a placeholder token)
- Response Status:** 401 Unauthorized
- Response Body (Raw):**

```
1
```

## Get booking by PNR(as admin):

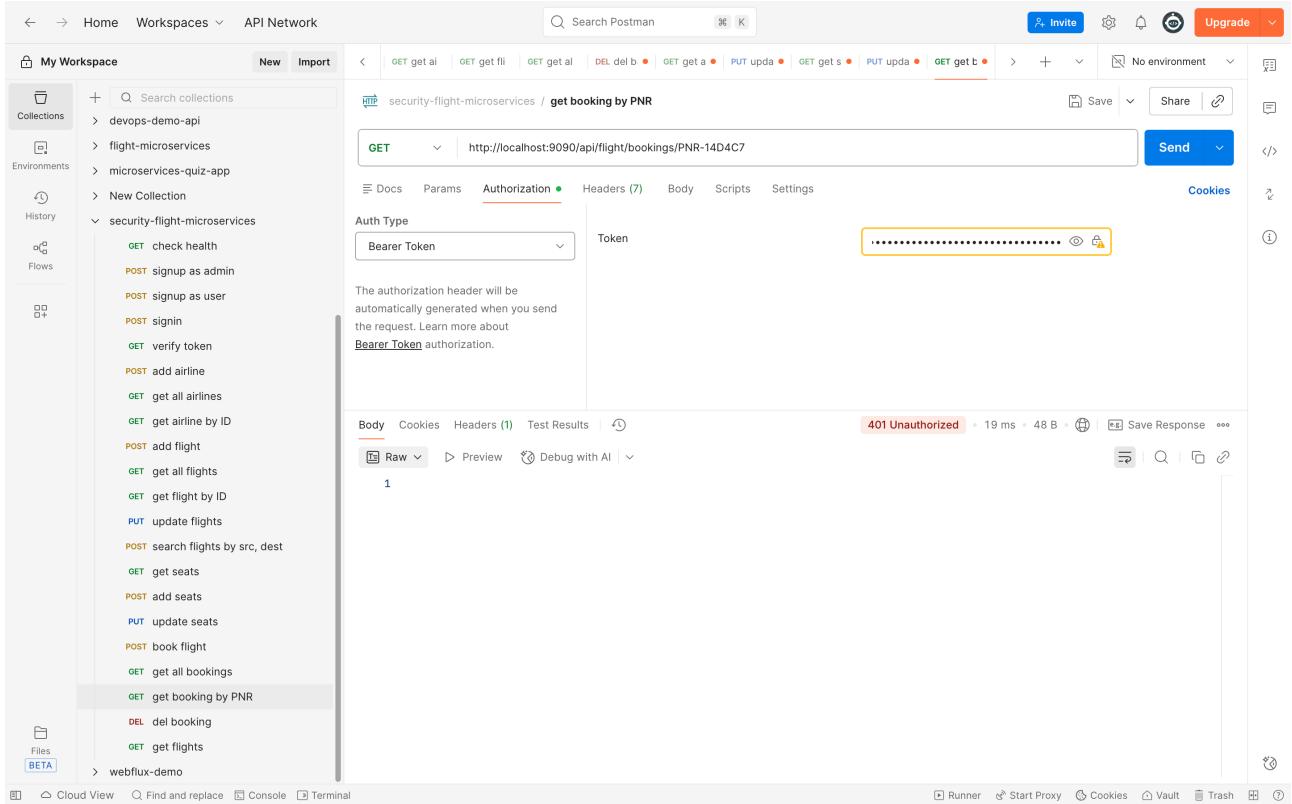


The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/flight/bookings/PNR-14D4C7
- Authorization:** Bearer Token (Token value is masked)
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
2   "email": "yaswithasadhamuri05@gmail.com",
3   "flightId": "6935130951cdcf5e6328b8fd",
4   "id": "14d4c753-860e-442a-830c-997440ae434a",
5   "passengerIds": [
6     "P1",
7     "P2"
8   ],
9   "pnr": "PNR-14D4C7",
10  "seatCount": 2,
11  "seatNumbers": [
12    "1A",
13    "1B"
14  ],
15  "totalAmount": 0.0
16 }
```

## Get booking by PNR(as user - restricted):



The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/flight/bookings/PNR-14D4C7
- Authorization:** Bearer Token (Token value is masked)
- Response Status:** 401 Unauthorized
- Response Body (Raw JSON):**

```
1
```

## Delete booking(both admin and user are allowed):

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with collections like 'devops-demo-api', 'flight-microservices', 'microservices-quiz-app', and 'security-flight-microservices'. The 'security-flight-microservices' collection is expanded, showing various API endpoints such as 'check health', 'signup as admin', 'signup as user', 'signin', 'verify token', 'add airline', 'get all airlines', 'get airline by ID', 'add flight', 'get all flights', 'get flight by ID', 'update flights', 'search flights by src, dest', 'get seats', 'add seats', 'update seats', 'book flight', 'get all bookings', 'get booking by PNR', 'del booking', 'get flights', and 'webflux-demo'. The 'Files' section is marked as BETA.

The main workspace shows a DELETE request to `http://localhost:9090/api/flight/bookings/956385b4-3653-4475-b34a-5b7545a29889`. The 'Authorization' tab is selected, showing 'Bearer Token' and a placeholder 'Token' with a yellow border. Below it, a note explains that the authorization header will be automatically generated when sending the request. The 'Headers' tab shows 7 headers. The 'Body' tab is empty. The 'Test Results' tab shows a successful response: `200 OK`, `583 ms`, `308 B`, and a message `Booking deleted successfully!`.