Flight Management System

MicroServices & Security and Docker Implementation

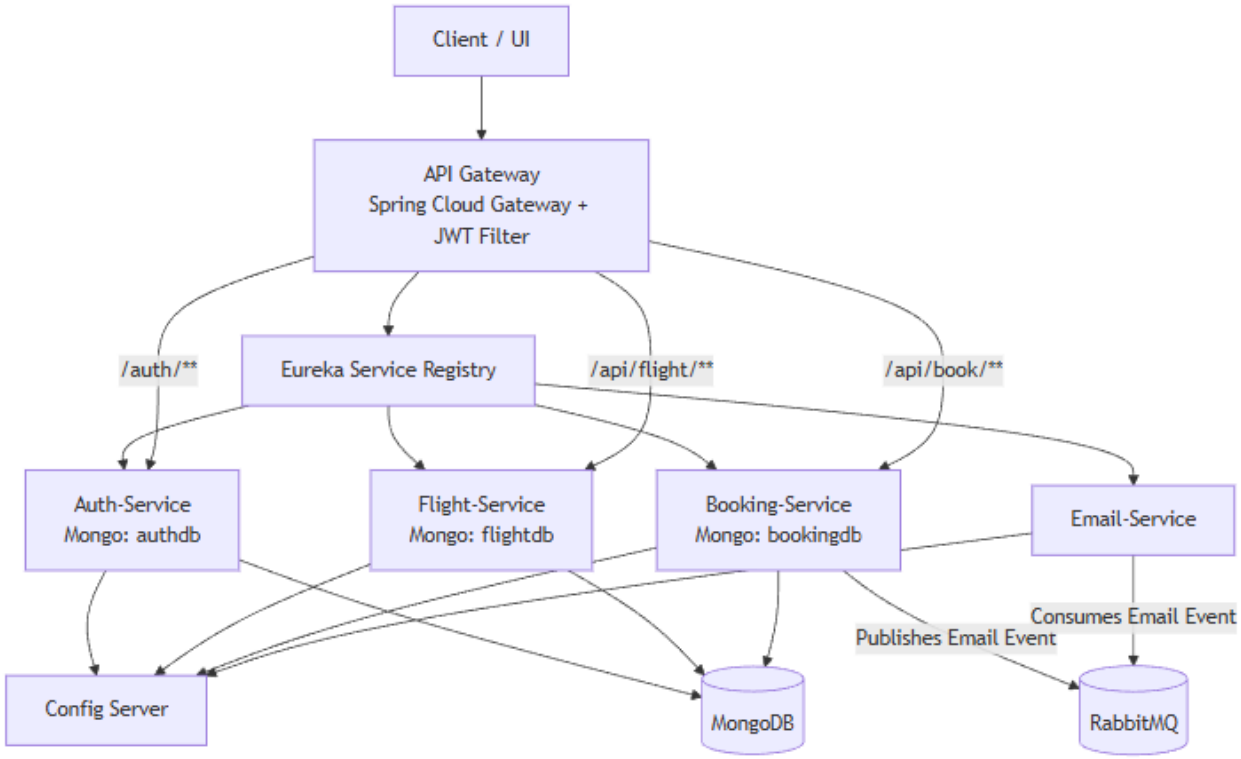
Week - 7

Assignment

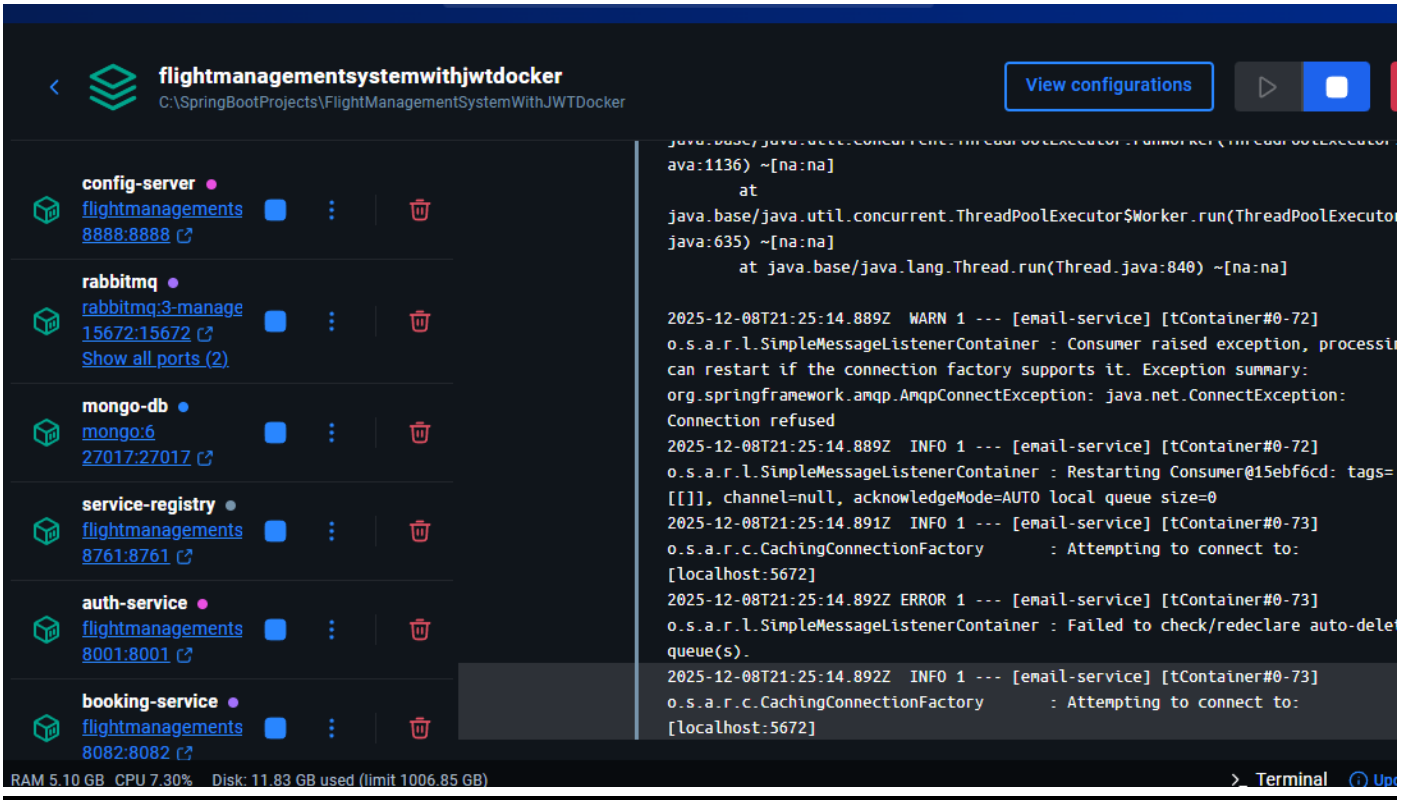
Project Report

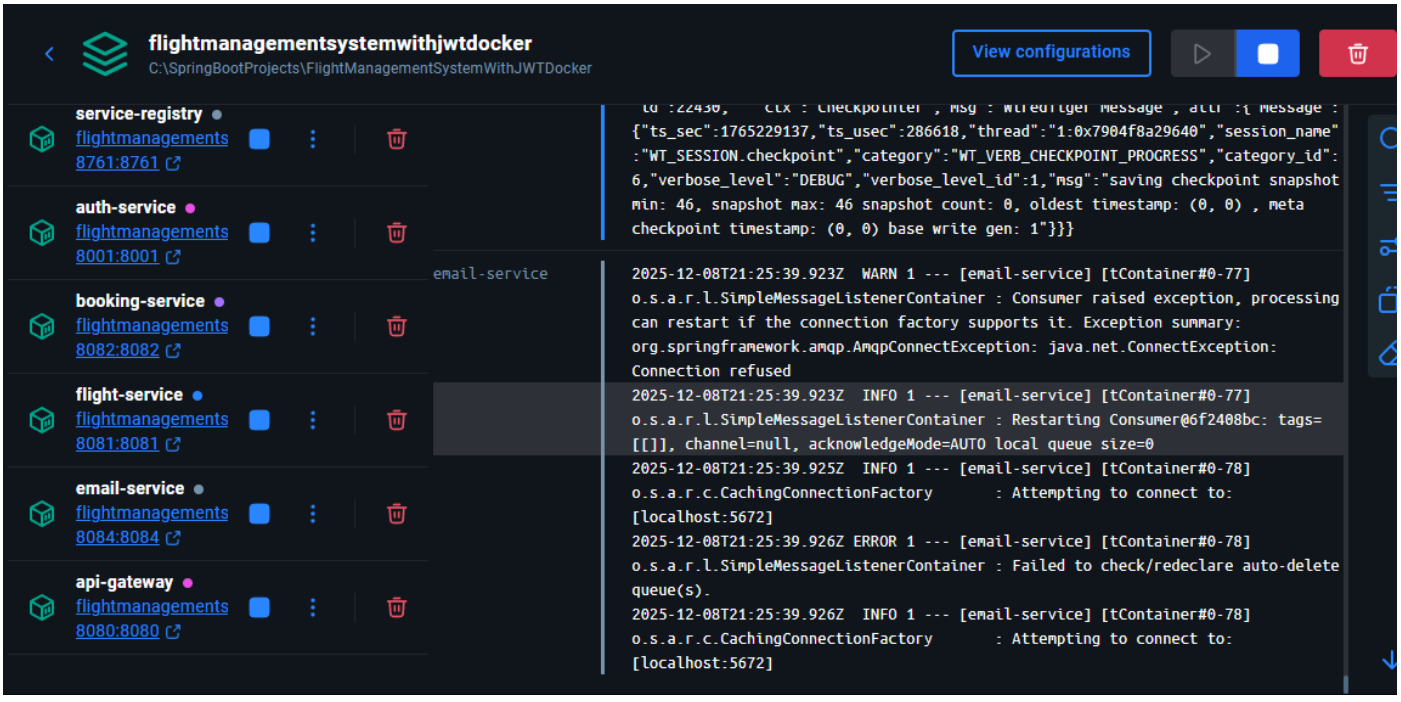
Important Screenshots

MicroServices Architecture



All my microservices are deployed in docker and running from there:-





These are my services which are registered in service registry

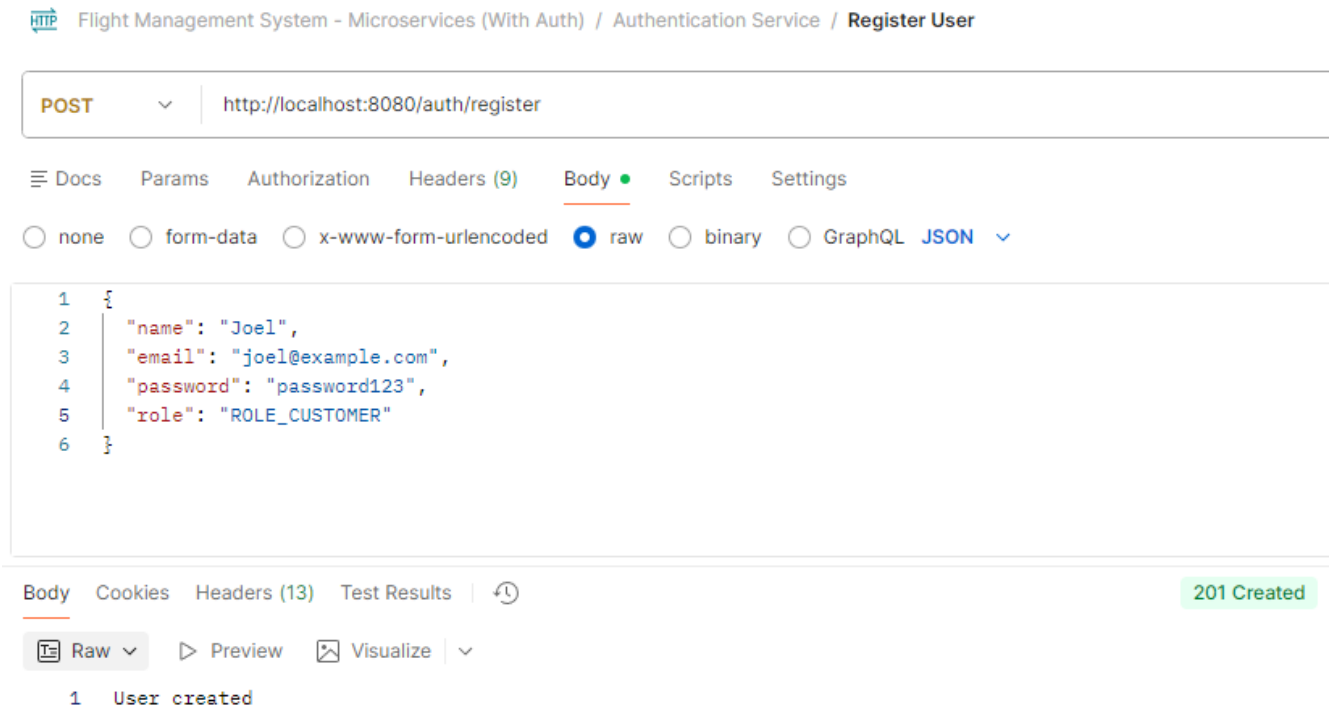
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:api-gateway:8080
AUTH-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:auth-service:8001
BOOKING-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:booking-service:8082
EMAIL-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:email-service:8084
FLIGHT-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:flight-service:8081
FMS-CONFIG-SERVER	n/a (1)	(1)	UP (1) - DESKTOP-U2KN9DC:fms-config-server:8888

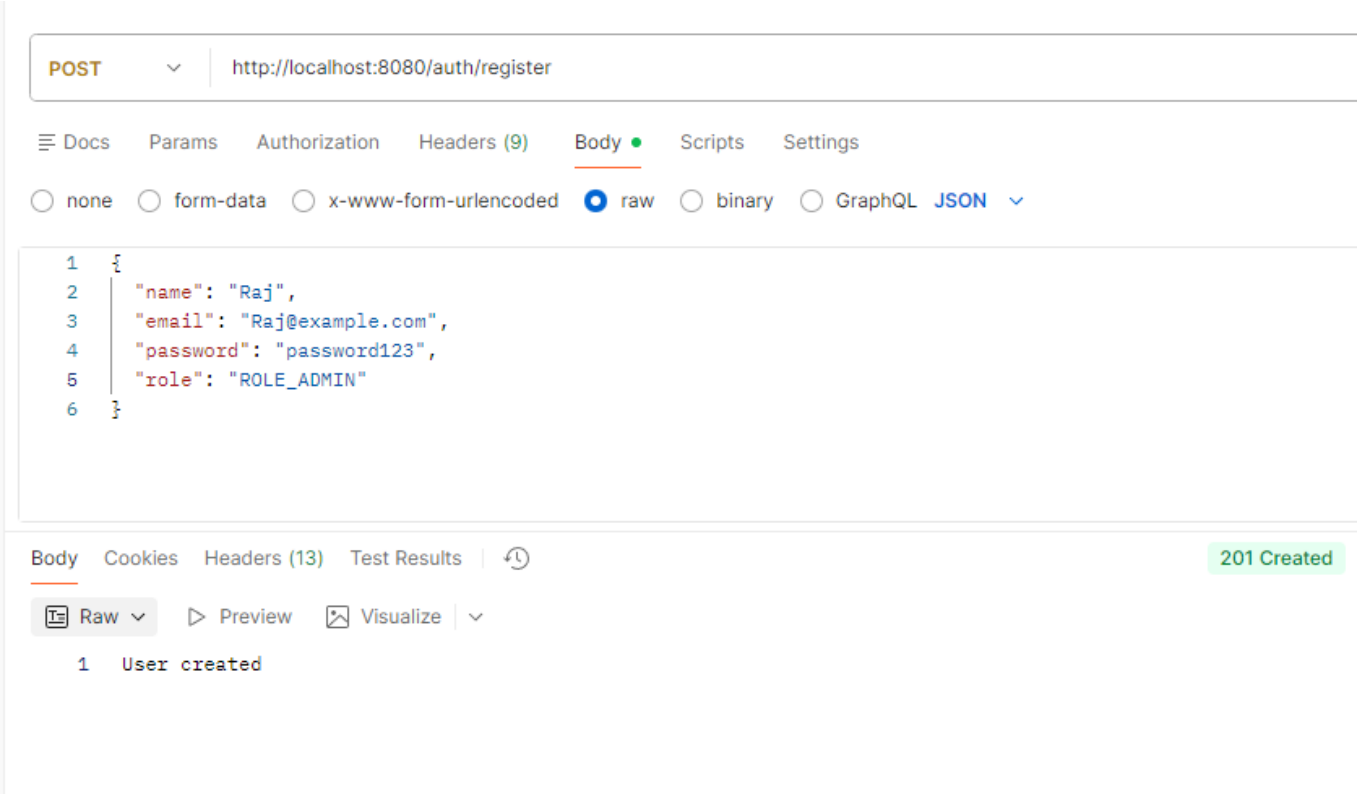
General Info

Name	Value
total-avail-memory	72mb
num-of-cpus	12
current-memory-usage	48mb (66%)

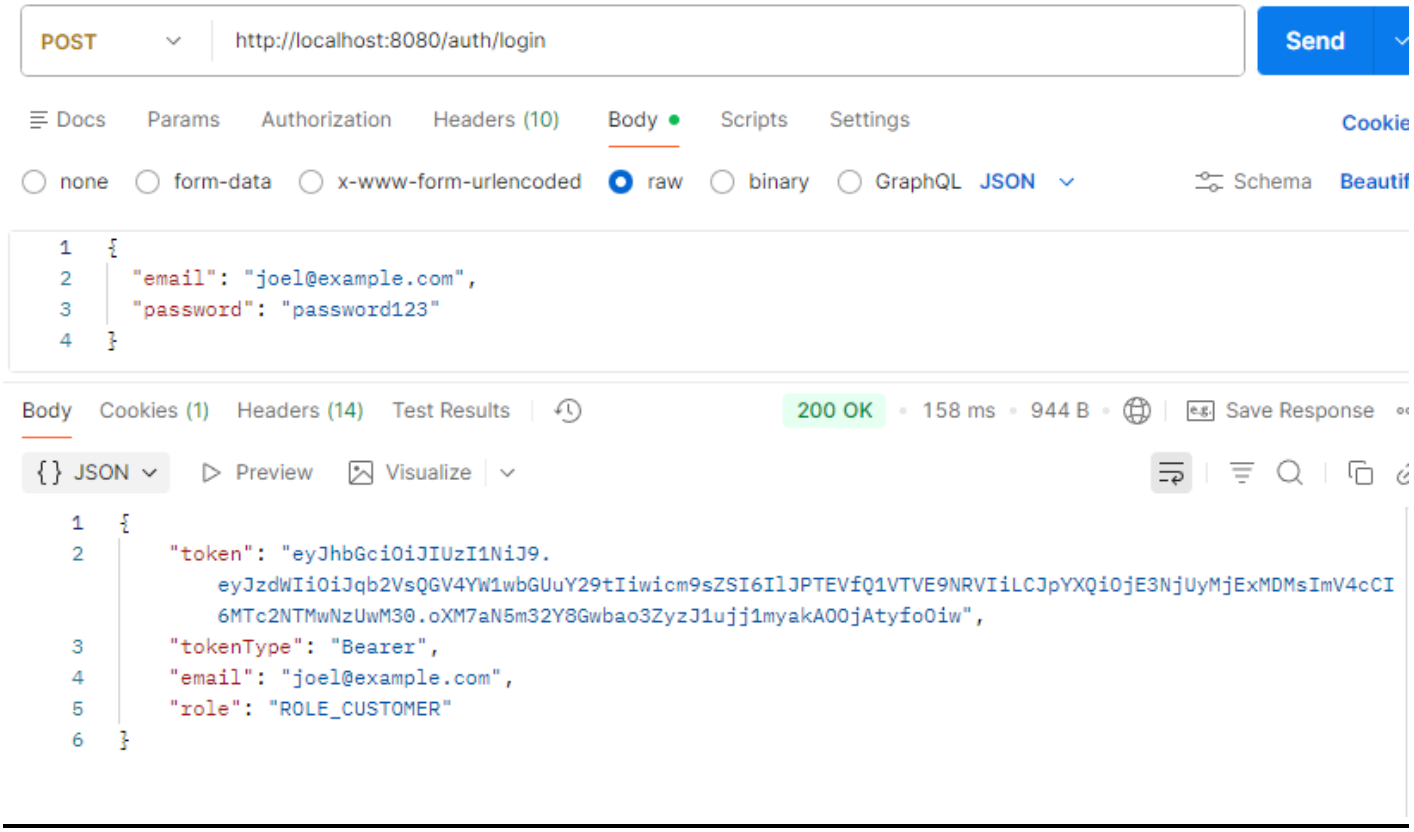
I have added Authentication Endpoints such as :- Register User – Customer (Books Flights)



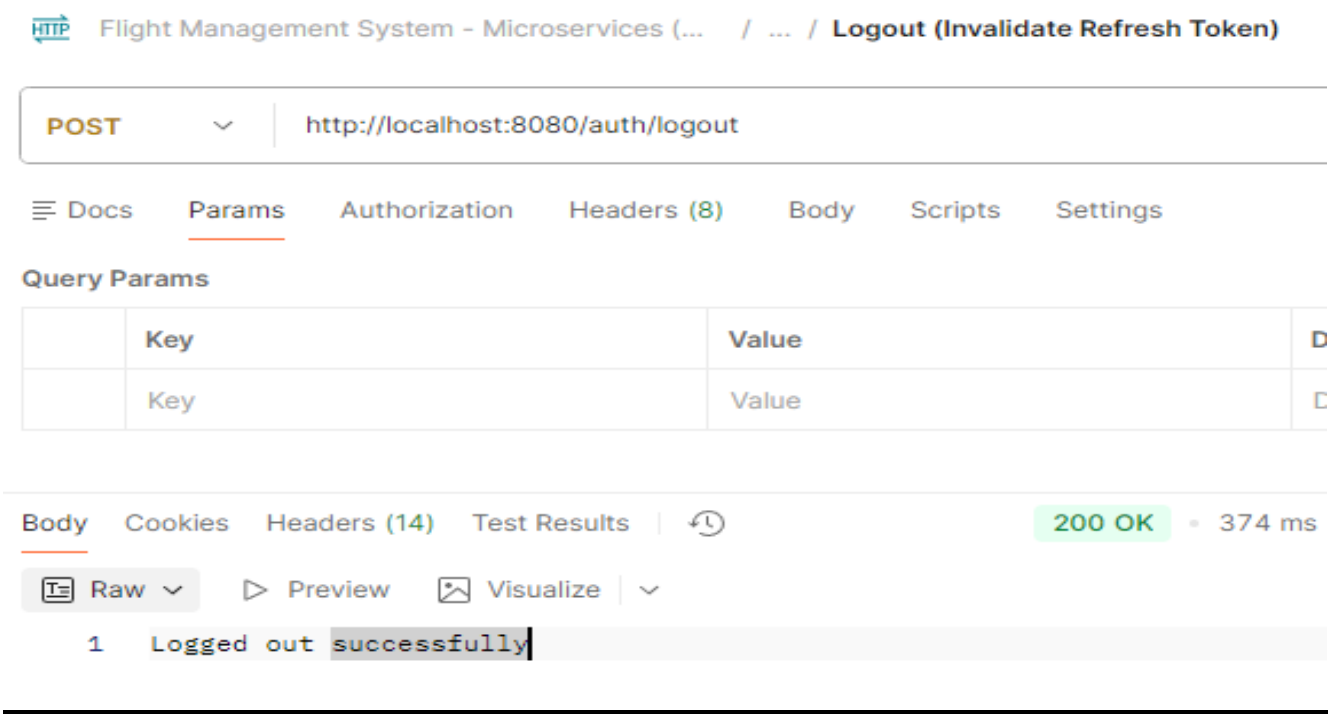
Register Admin (Add Flights and Airlines)



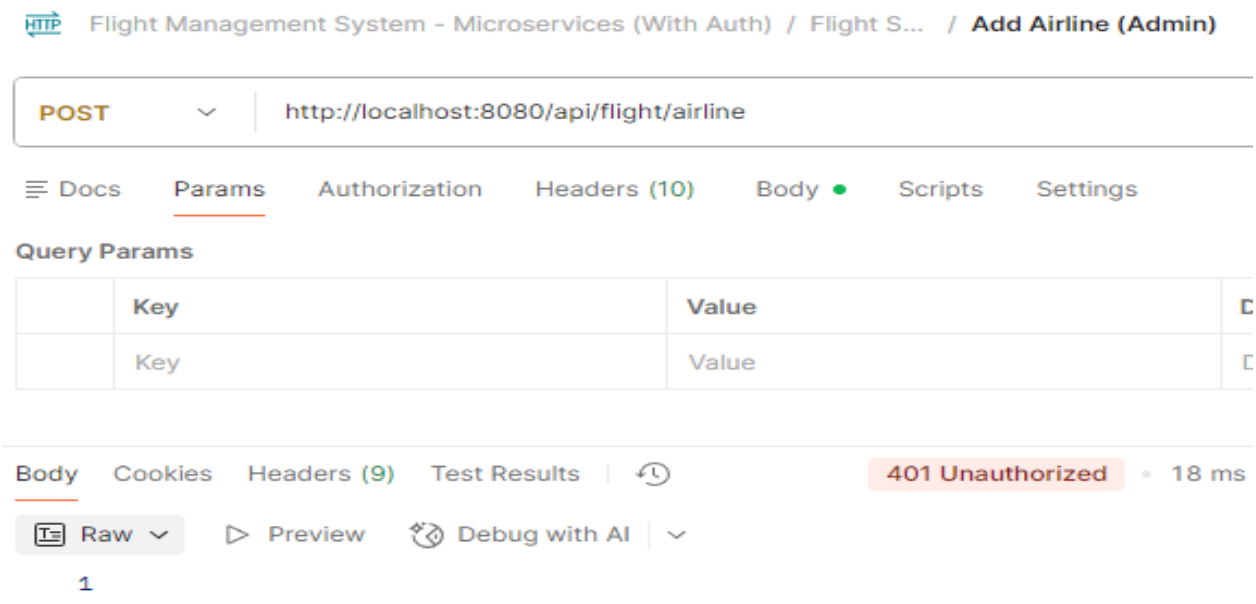
Login



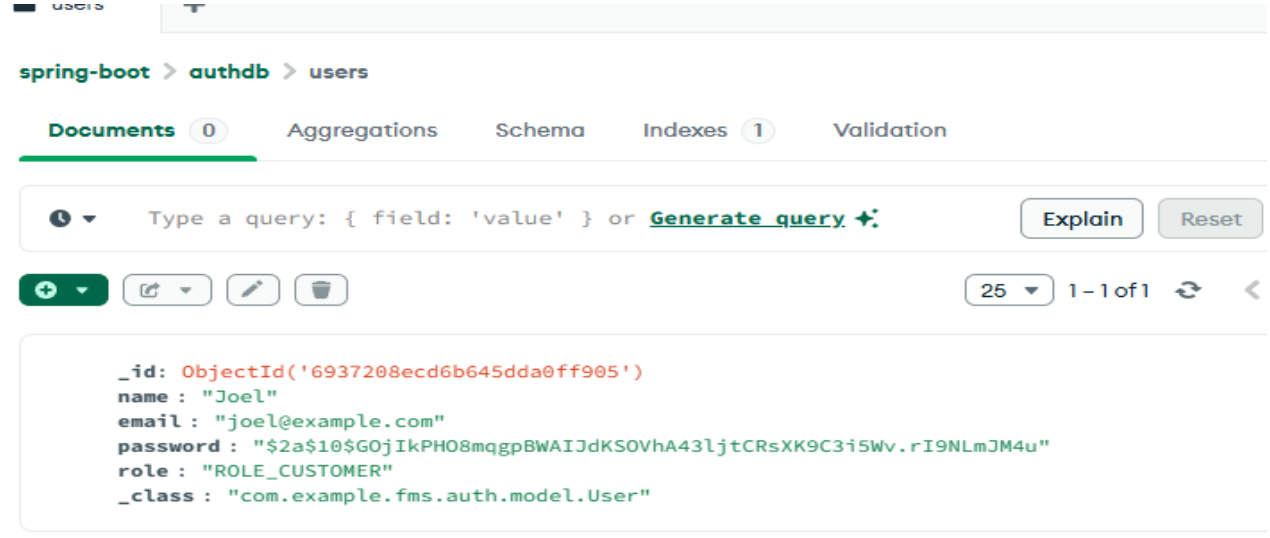
Logout



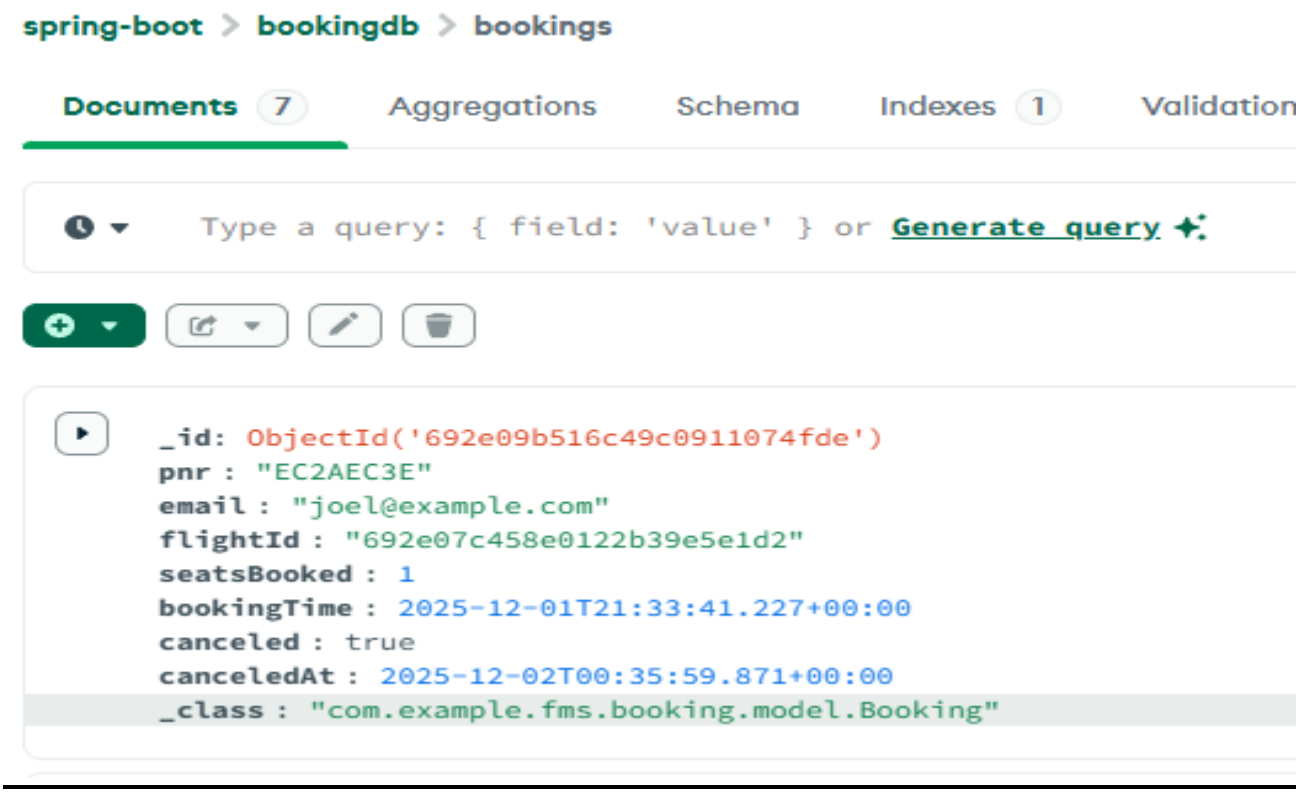
I have implemented Role Based Authentication so if a customer tries to access Admin Endpoints it will return 401 Error

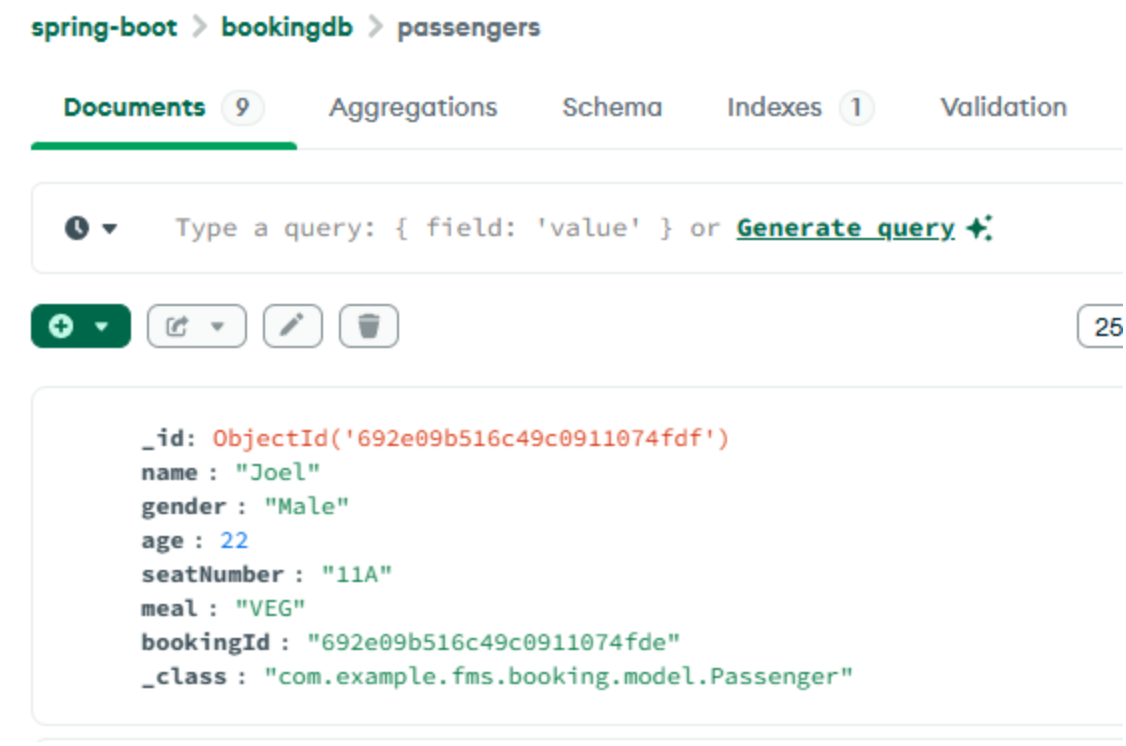


MongoDB – AuthDB

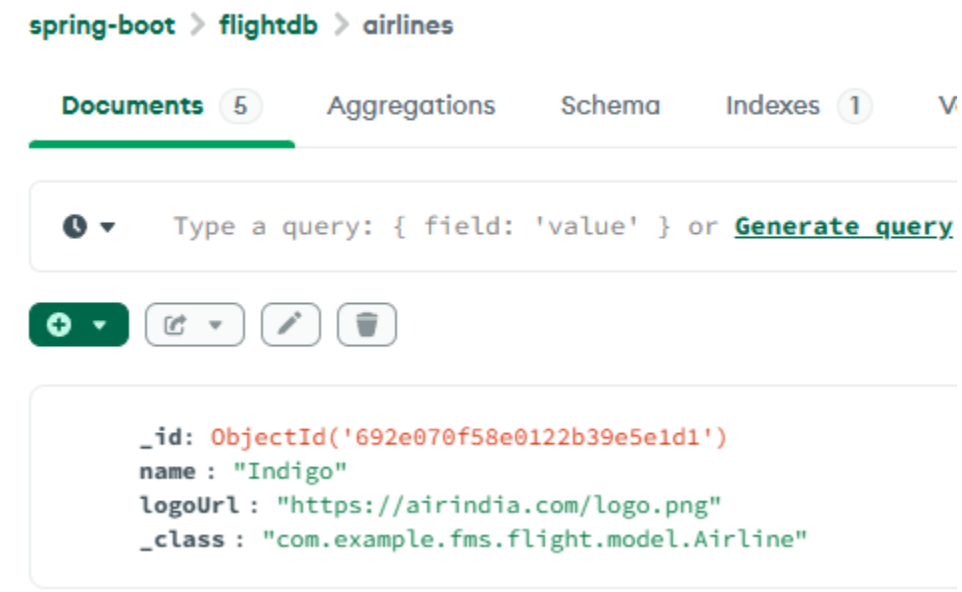


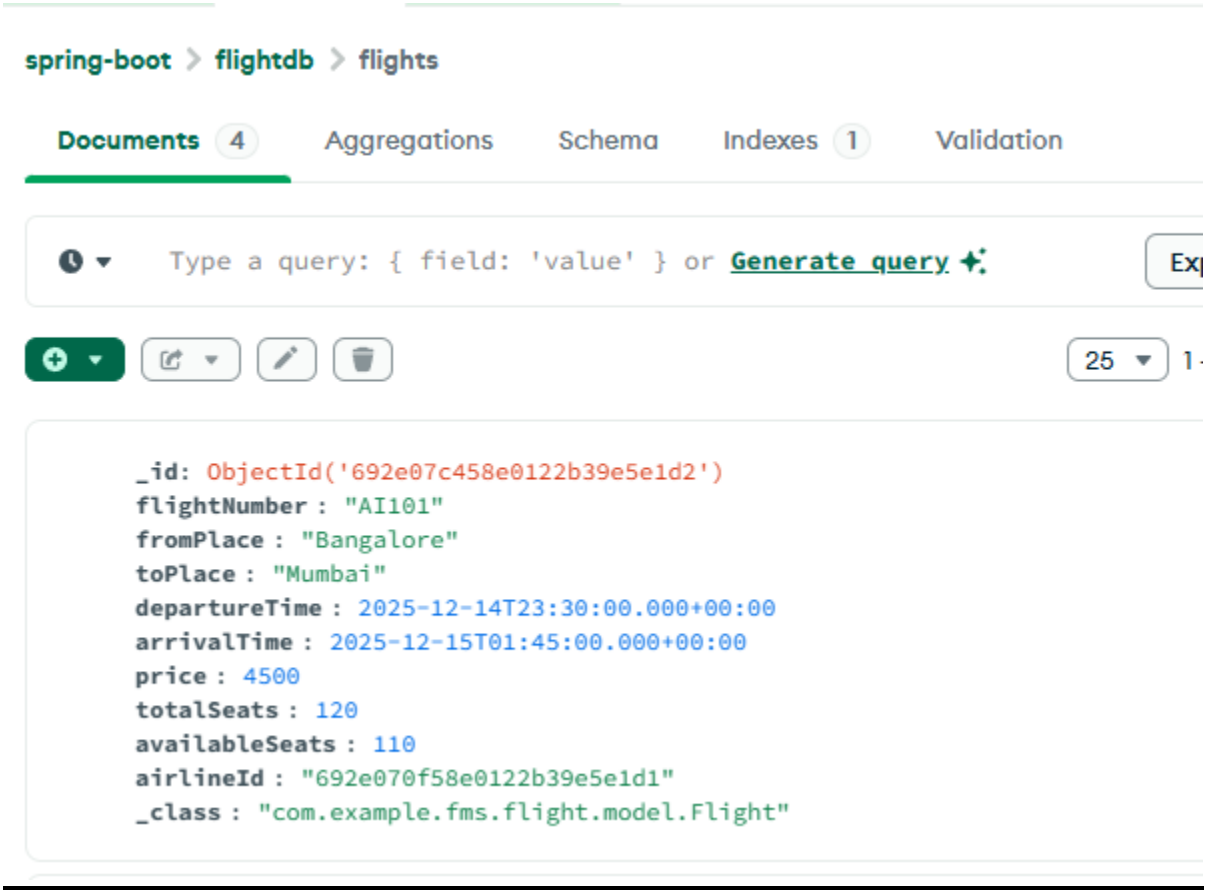
BookingDB



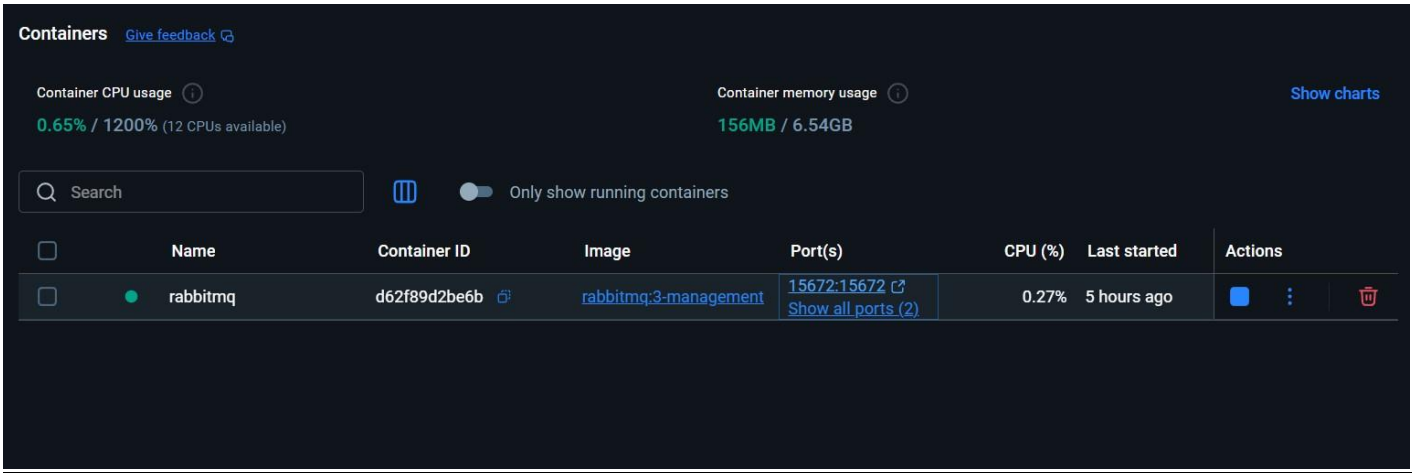

spring-boot > bookingdb > passengers
Documents 9
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
25
_id: ObjectId('692e09b516c49c0911074fdf')
name : "Joel"
gender : "Male"
age : 22
seatNumber : "11A"
meal : "VEG"
bookingId : "692e09b516c49c0911074fde"
_class : "com.example.fms.booking.model.Passenger"

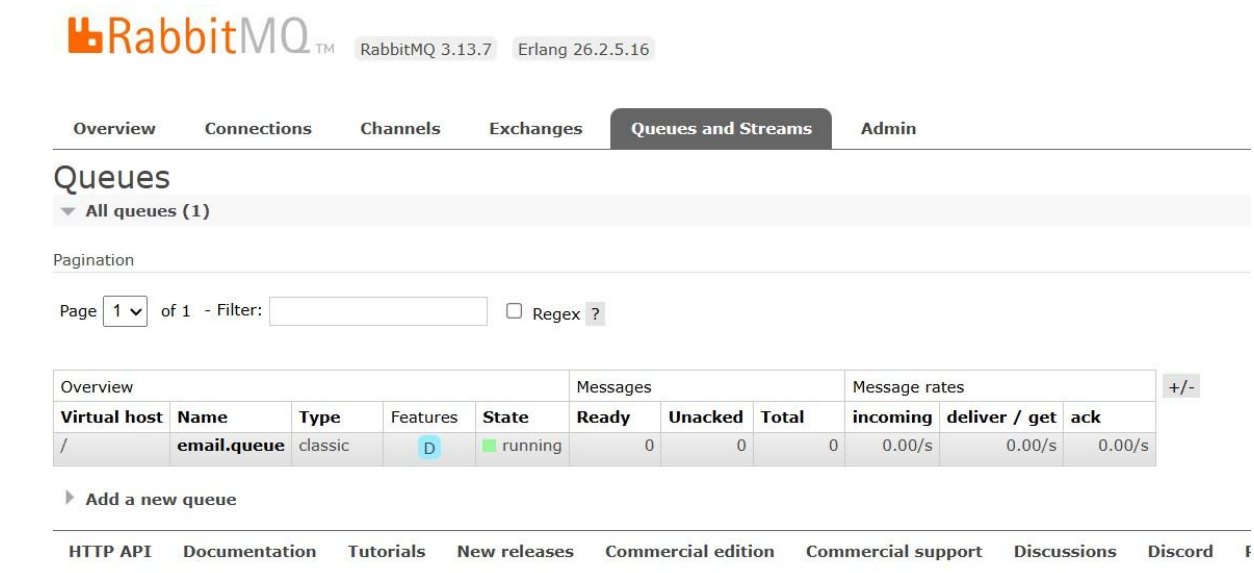
FlightDB

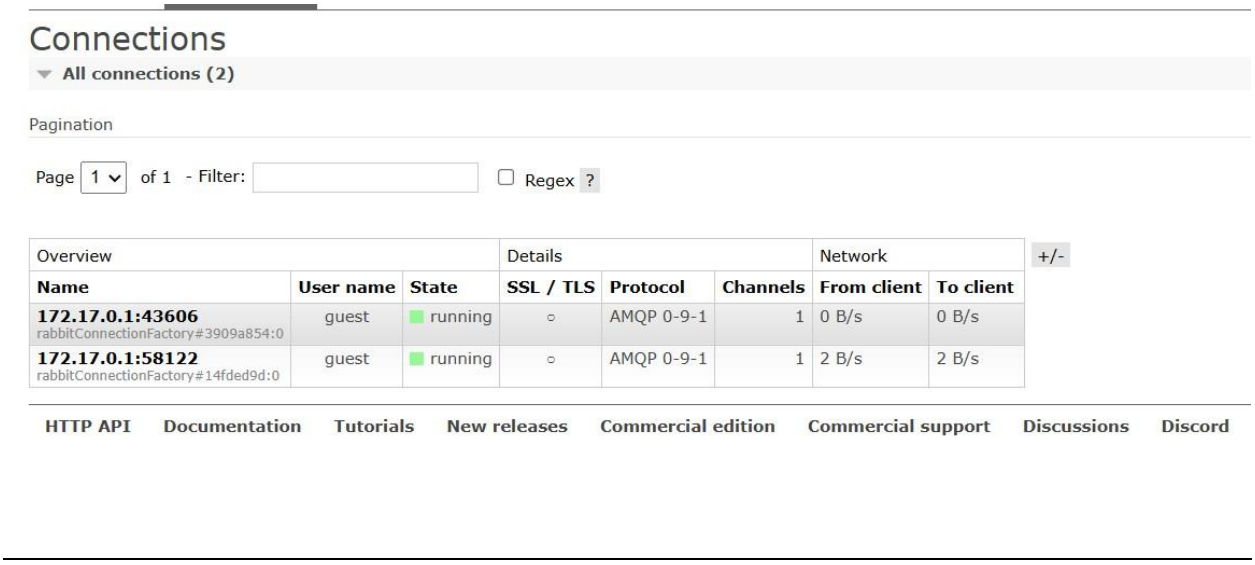

spring-boot > flightdb > airlines
Documents 5
Aggregations
Schema
Indexes 1
Type a query: { field: 'value' } or Generate query
_id: ObjectId('692e070f58e0122b39e5e1d1')
name : "Indigo"
logoUrl : "https://airindia.com/logo.png"
_class : "com.example.fms.flight.model.Airline"



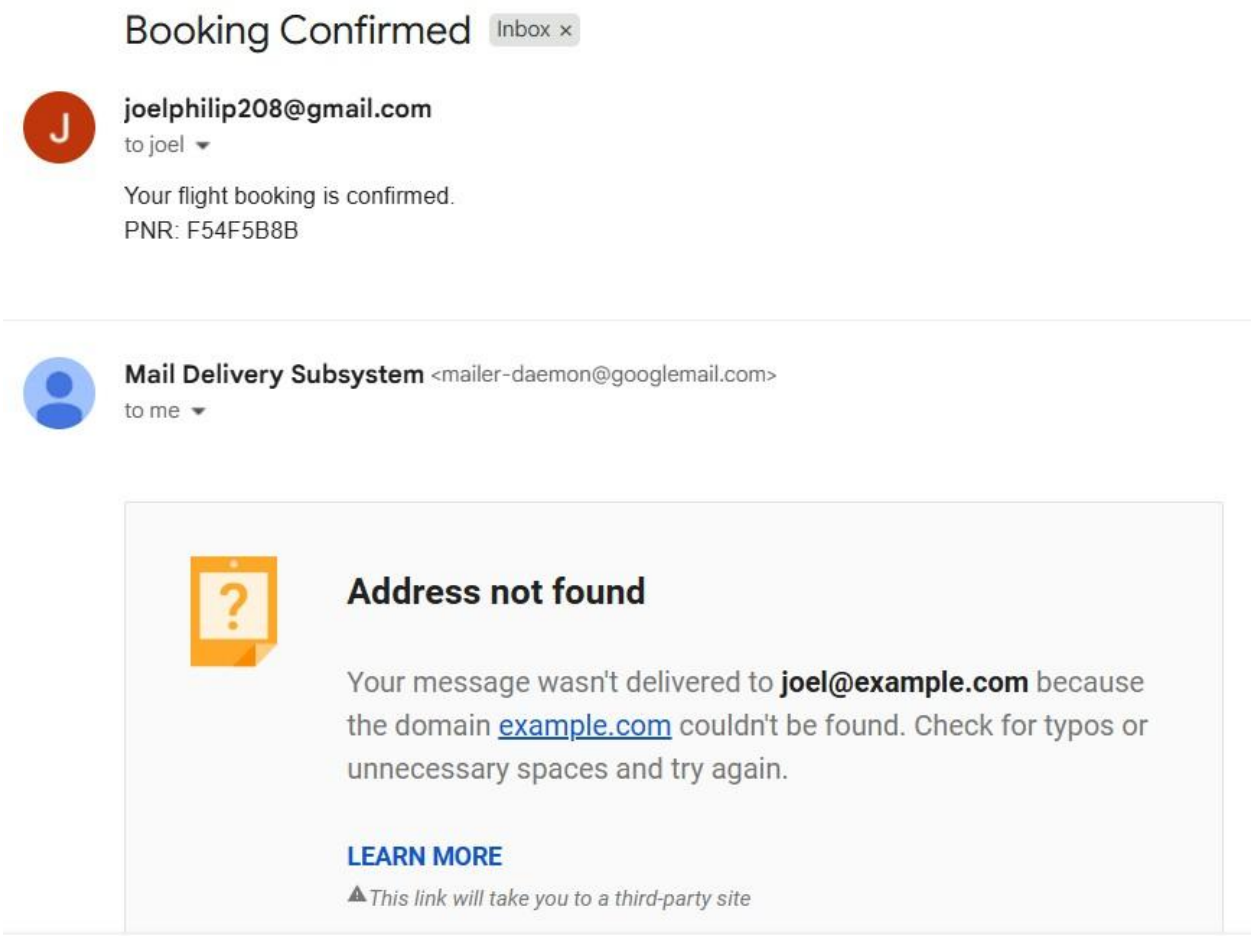
I have used RabbitMQ as my Message Broker to send emails




RabbitMQ™ RabbitMQ 3.13.7 Erlang 26.2.5.16
Overview Connections Channels Exchanges Queues and Streams Admin
Queues
All queues (1)
Pagination
Page 1 of 1 - Filter: Regex ?
Overview Messages Message rates +/-
Virtual host Name Type Features State Ready Unacked Total incoming deliver / get ack
/ email.queue classic D running 0 0 0 0.00/s 0.00/s 0.00/s
Add a new queue
HTTP API Documentation Tutorials New releases Commercial edition Commercial support Discussions Discord

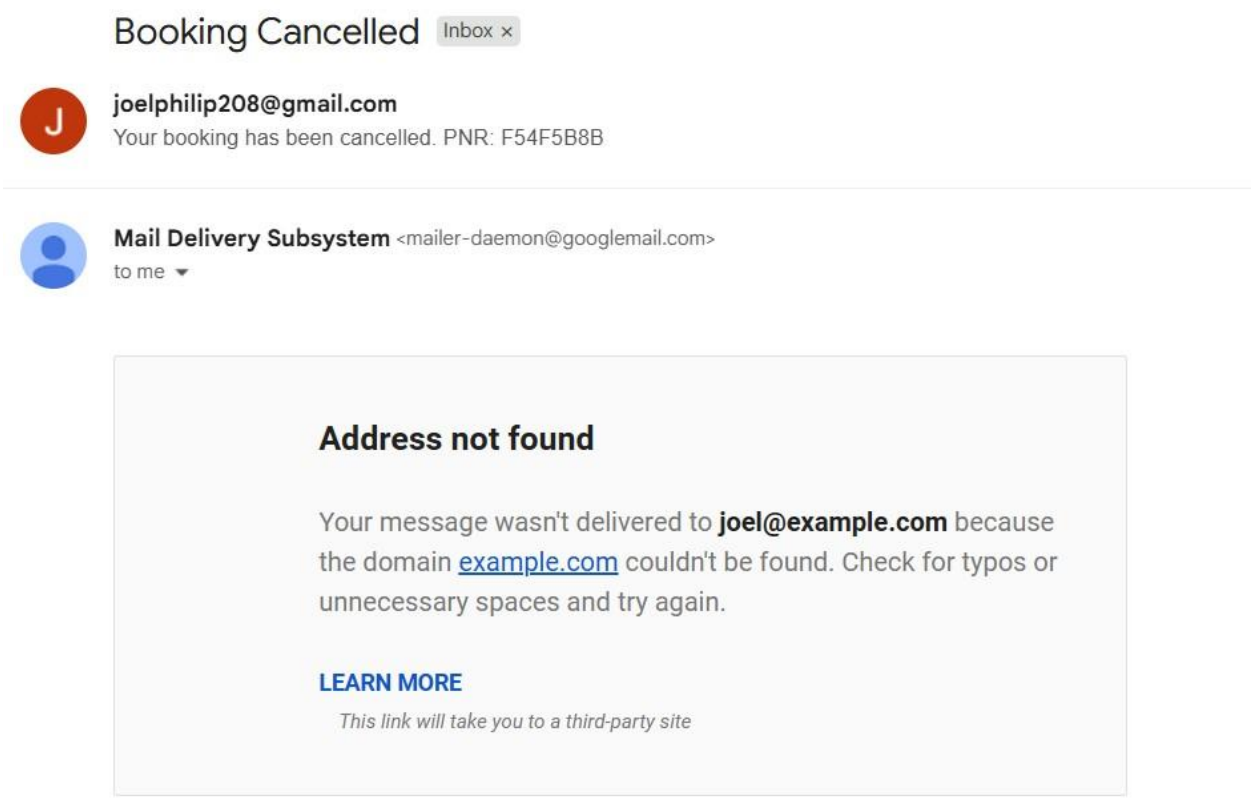

Connections
All connections (2)
Pagination
Page 1 of 1 - Filter: Regex ?
Overview Details Network +/-
Name User name State SSL / TLS Protocol Channels From client To client
172.17.0.1:43606 rabbitConnectionFactory#3909a854:0 guest running ○ AMQP 0-9-1 1 0 B/s 0 B/s
172.17.0.1:58122 rabbitConnectionFactory#14fded9d:0 guest running ○ AMQP 0-9-1 1 2 B/s 2 B/s
HTTP API Documentation Tutorials New releases Commercial edition Commercial support Discussions Discord

This is the email that it will send when you book or cancel a ticket




```
: EMAIL SERVICE → Using SMTP user: joelphilip208@gmail.com
: Attempting to send email to: joel@example.com
: Email sent successfully to joel@example.com
: Received EmailEvent from RabbitMQ: EmailEvent[to=joel@example.com, subject=Booking Confirmed, body=Your flight booking is confirmed.

: EMAIL SERVICE → Using SMTP user: joelphilip208@gmail.com
: Attempting to send email to: joel@example.com
: Email sent successfully to joel@example.com
: Received EmailEvent from RabbitMQ: EmailEvent[to=joel@example.com, subject=Booking Cancelled, body=Your booking has been cancelled.
```

I have used SMTP to do it and I put my credentials in my run configurations



I have used newman tool to test my API end points and every request gave a valid response

```
C:\Users\Hp\OneDrive\Documents\Postman - Collections>newman --version
6.2.1

C:\Users\Hp\OneDrive\Documents\Postman - Collections>newman run FlightManagementSystem-Microservices.postman_collection.json
newman
```

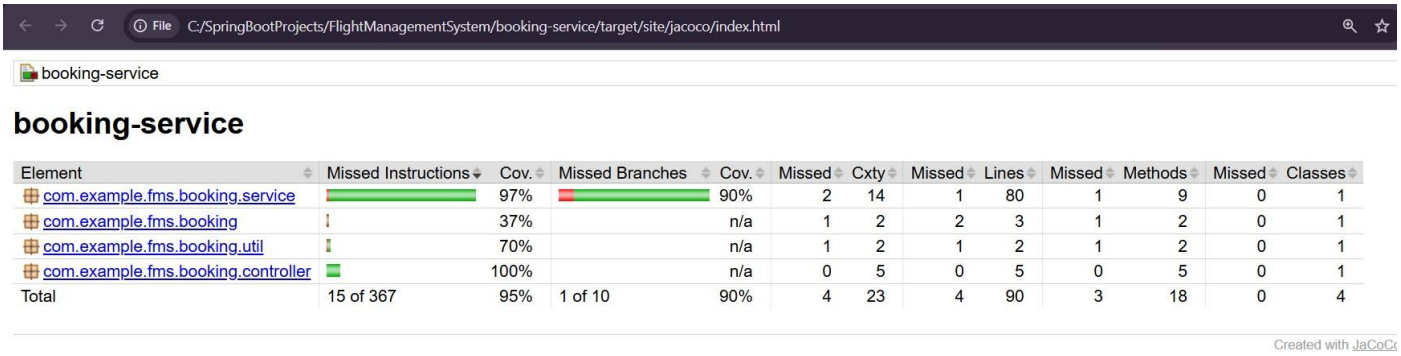
```
Flight Management System - Microservices

□ Health Check
└ API Gateway Health
  GET http://localhost:8080/actuator/health [200 OK, 114B, 235ms]

└ Eureka Server UI
  GET http://localhost:8761 [200 OK, 6.08kB, 153ms]

□ Flight Service (via API Gateway)
└ Add Airline (Admin)
  POST http://localhost:8080/api/flight/airline [201 Created, 149B, 529ms]

└ Add Flight Inventory (Admin)
  POST http://localhost:8080/api/flight/airline/692e070f58e0122b39e5e1d1/inventory [201 Created, 156B, 53ms]

└ Search Flights (User)
  POST http://localhost:8080/api/flight/search [200 OK, 583B, 34ms]

└ Get Flight Details
  GET http://localhost:8080/api/flight/details/692e07c458e0122b39e5e1d2 [200 OK, 342B, 18ms]

□ Booking Service (via API Gateway)
└ Book Flight
  POST http://localhost:8080/api/book/book [201 Created, 222B, 634ms]

└ Cancel Booking
  DELETE http://localhost:8080/api/book/cancel/EC2AEC3E [200 OK, 155B, 83ms]

└ Booking History
  GET http://localhost:8080/api/book/history/joel@example.com [200 OK, 595B, 28ms]

└ Get Ticket
  GET http://localhost:8080/api/book/ticket/EC2AEC3E [200 OK, 648B, 36ms]
```

	executed	failed
iterations	1	0
requests	10	0
test-scripts	0	0
prerequest-scripts	0	0
assertions	0	0
total run duration: 1007ms		
total data received: 8.21kB (approx)		
average response time: 16ms [min: 10ms, max: 29ms, s.d.: 6ms]		

JaCoCo Report

For Booking Service :-

Code coverage is 95 %



File C:/SpringBootProjects/FlightManagementSystem/booking-service/target/site/jacoco/index.html

booking-service

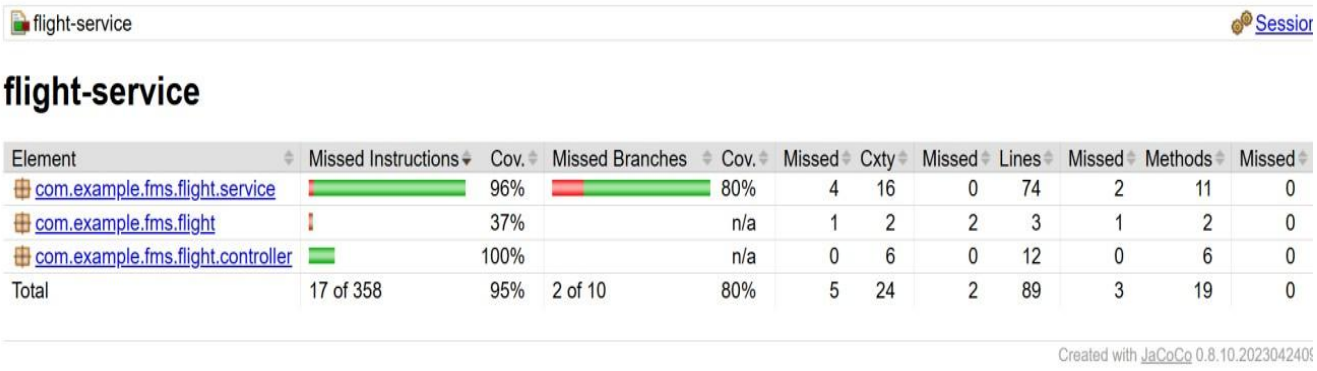
booking-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.example.fms.booking.service		97%		90%	2	14	1	80	1	9	0	1
com.example.fms.booking		37%		n/a	1	2	2	3	1	2	0	1
com.example.fms.booking.util		70%		n/a	1	2	1	2	1	2	0	1
com.example.fms.booking.controller		100%		n/a	0	5	0	5	0	5	0	1
Total	15 of 367	95%	1 of 10	90%	4	23	4	90	3	18	0	4

Created with JaCoCo

For Flight Service:-

Code Coverage is 95%



flight-service

Session

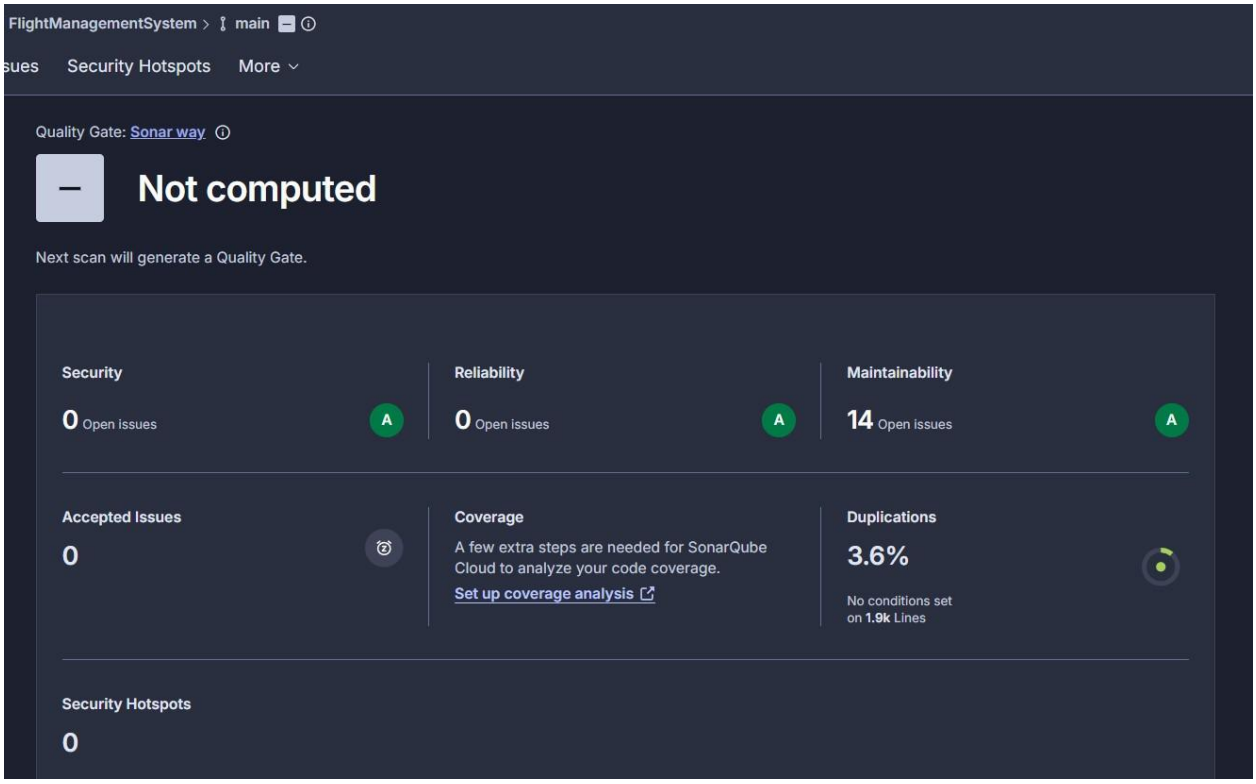
flight-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed
com.example.fms.flight.service		96%		80%	4	16	0	74	2	11	0
com.example.fms.flight		37%		n/a	1	2	2	3	1	2	0
com.example.fms.flight.controller		100%		n/a	0	6	0	12	0	6	0
Total	17 of 358	95%	2 of 10	80%	5	24	2	89	3	19	0

Created with JaCoCo 0.8.10.202304240

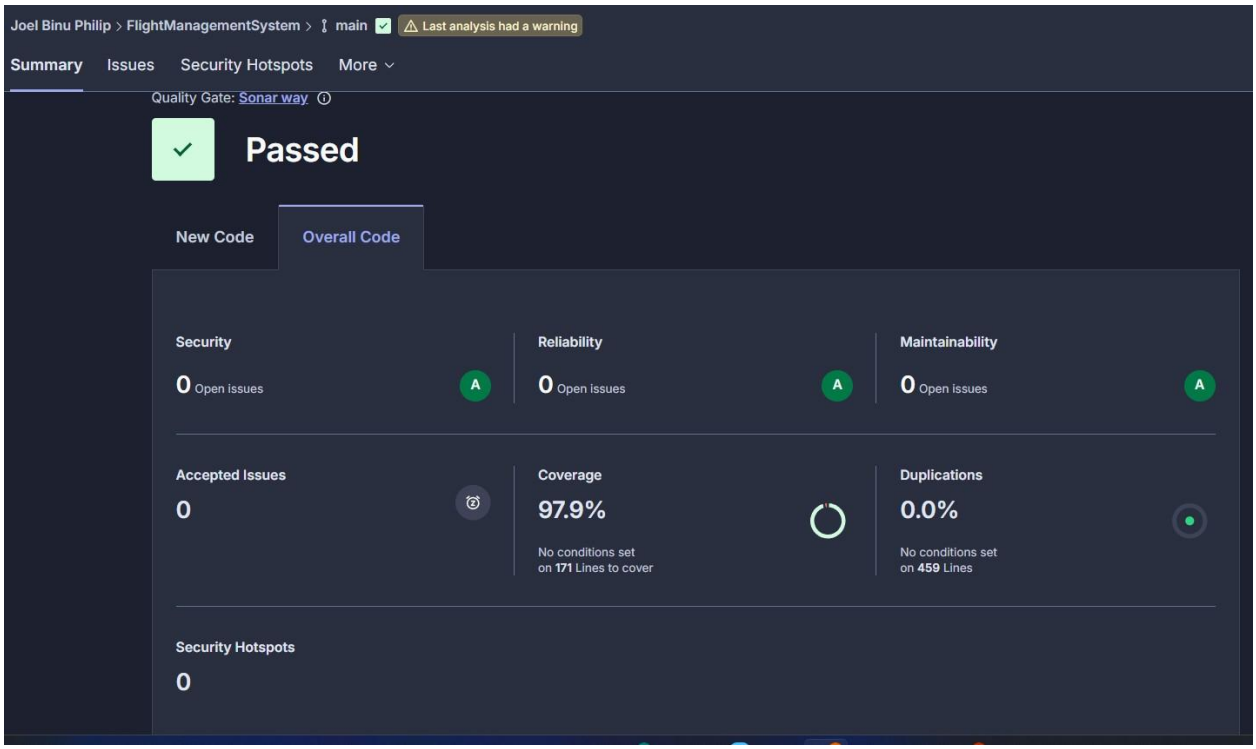
SonarQube Report

Before :-



Issue fixing history graph

After :-



JMeter Load Testing Using CLI

20 threads

```
C:\Users\Hp\Downloads> "C:\Tools\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter.bat" -n -t working_plan.jmx -l results20.jtl -e -o
 report20
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using working_plan.jmx
Starting standalone test @ 2025 Dec 2 07:23:55 IST (1764640435750)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary +    217 in 00:00:04 =   51.3/s Avg:     5 Min:     3 Max:    35 Err:     0 (0.00%) Active: 1 Started: 73 Finished: 72
summary +    293 in 00:00:06 =   51.2/s Avg:     4 Min:     3 Max:    10 Err:     0 (0.00%) Active: 0 Started: 170 Finished: 170
summary =    510 in 00:00:10 =   51.2/s Avg:     5 Min:     3 Max:    35 Err:     0 (0.00%)
Tidying up ...    @ 2025 Dec 2 07:24:05 IST (1764640445783)
... end of run
```

50 threads

```
C:\Users\Hp\Downloads> "C:\Tools\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter.bat" -n -t working_plan.jmx -l results50.jtl -e -o
 report50
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using working_plan.jmx
Starting standalone test @ 2025 Dec 2 07:24:42 IST (1764640482536)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    510 in 00:00:10 =   51.2/s Avg:     5 Min:     3 Max:    36 Err:     0 (0.00%)
Tidying up ...    @ 2025 Dec 2 07:24:52 IST (1764640492569)
... end of run
```

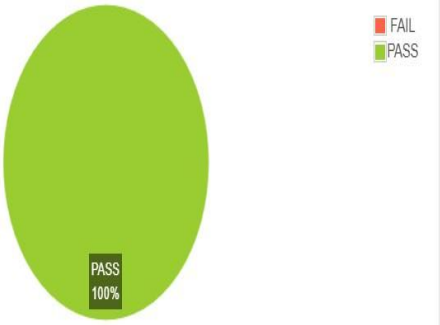
100 Threads

```
C:\Users\Hp\Downloads> "C:\Tools\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\jmeter.bat" -n -t working_plan.jmx -l results100.jtl -e -
o report100
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using working_plan.jmx
Starting standalone test @ 2025 Dec 2 07:25:12 IST (1764640512339)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary =    510 in 00:00:10 =   51.2/s Avg:     4 Min:     2 Max:    32 Err:     0 (0.00%)
Tidying up ...    @ 2025 Dec 2 07:25:22 IST (1764640522375)
... end of run
```

20 threads report

C:/Users/Hp/Downloads/report20/index.html

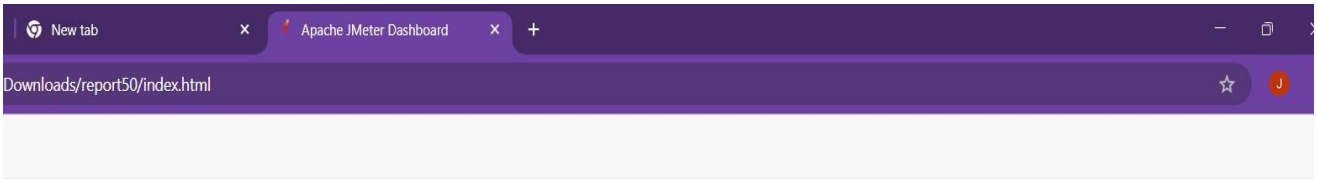
Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
1.000	500 ms	1 sec 500 ms	Total
1.000	500 ms	1 sec 500 ms	GET Ticket
1.000	500 ms	1 sec 500 ms	GET Flight Details
1.000	500 ms	1 sec 500 ms	GET Booking History



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	510	0	0.00%	5.28	3	35	4.00	7.90	8.00	10.00	51.87	28.37	7.36
GET Booking History	170	0	0.00%	3.86	3	6	4.00	4.90	5.00	6.00	17.36	10.64	2.46
GET Flight Details	170	0	0.00%	4.92	3	35	4.00	5.00	5.00	35.00	17.30	6.30	2.62
GET Ticket	170	0	0.00%	7.07	5	10	7.00	8.00	9.00	10.00	17.37	11.54	2.31

50 Threads Report



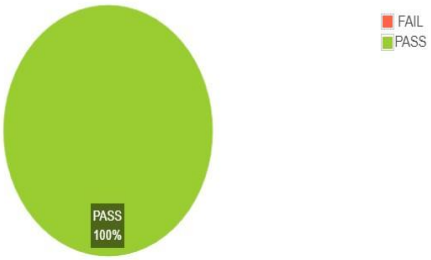
Test and Report information

Source file	"results50.jtl"
Start Time	"12/2/25, 7:24 AM"
End Time	"12/2/25, 7:24 AM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
1.000	**500 ms**	**1 sec 500 ms**	**Total**
1.000	500 ms	1 sec 500 ms	GET Ticket
1.000	500 ms	1 sec 500 ms	GET Flight Details
1.000	500 ms	1 sec 500 ms	GET Booking History

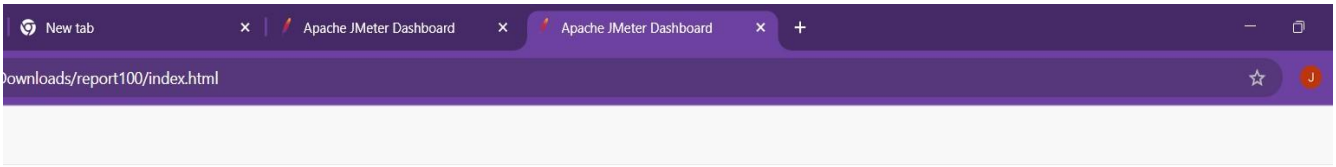
Requests Summary



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**510**	**0**	**0.00%**	**5.17**	**3**	**36**	**4.00**	**7.00**	**8.00**	**9.00**	**51.88**	**28.38**	**7.36**
GET Booking History	170	0	0.00%	3.85	3	6	4.00	5.00	5.00	6.00	17.37	10.64	2.46
GET Flight Details	170	0	0.00%	4.94	3	36	4.00	5.00	6.00	36.00	17.31	6.31	2.62
GET Ticket	170	0	0.00%	6.72	5	9	7.00	8.00	8.00	9.00	17.37	11.54	2.31

100 Threads



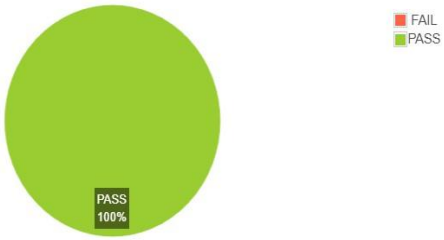
Test and Report information

Source file	"results100.jtl"
Start Time	"12/2/25, 7:25 AM"
End Time	"12/2/25, 7:25 AM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
1.000	**500 ms**	**1 sec 500 ms**	**Total**
1.000	500 ms	1 sec 500 ms	GET Ticket
1.000	500 ms	1 sec 500 ms	GET Flight Details
1.000	500 ms	1 sec 500 ms	GET Booking History

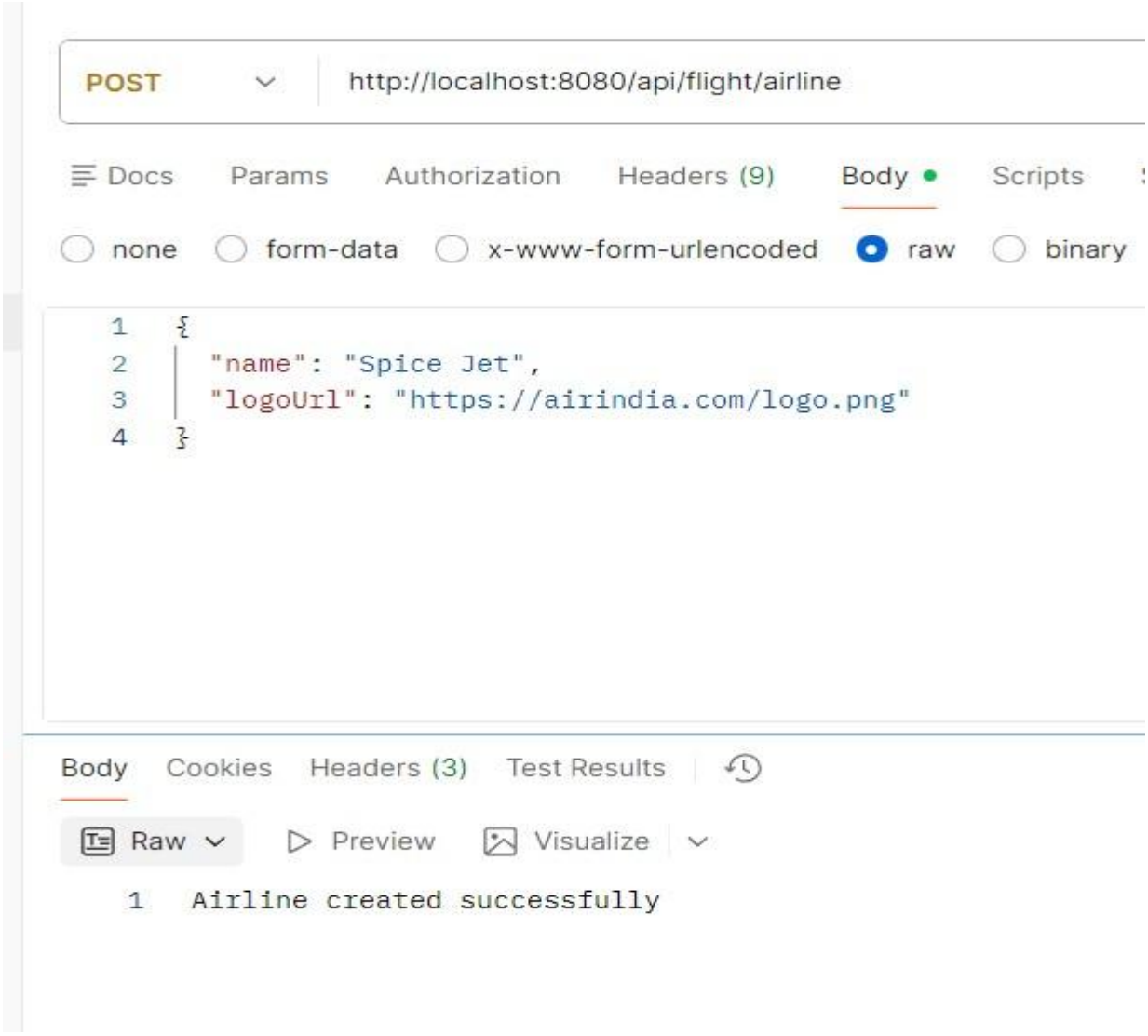
Requests Summary



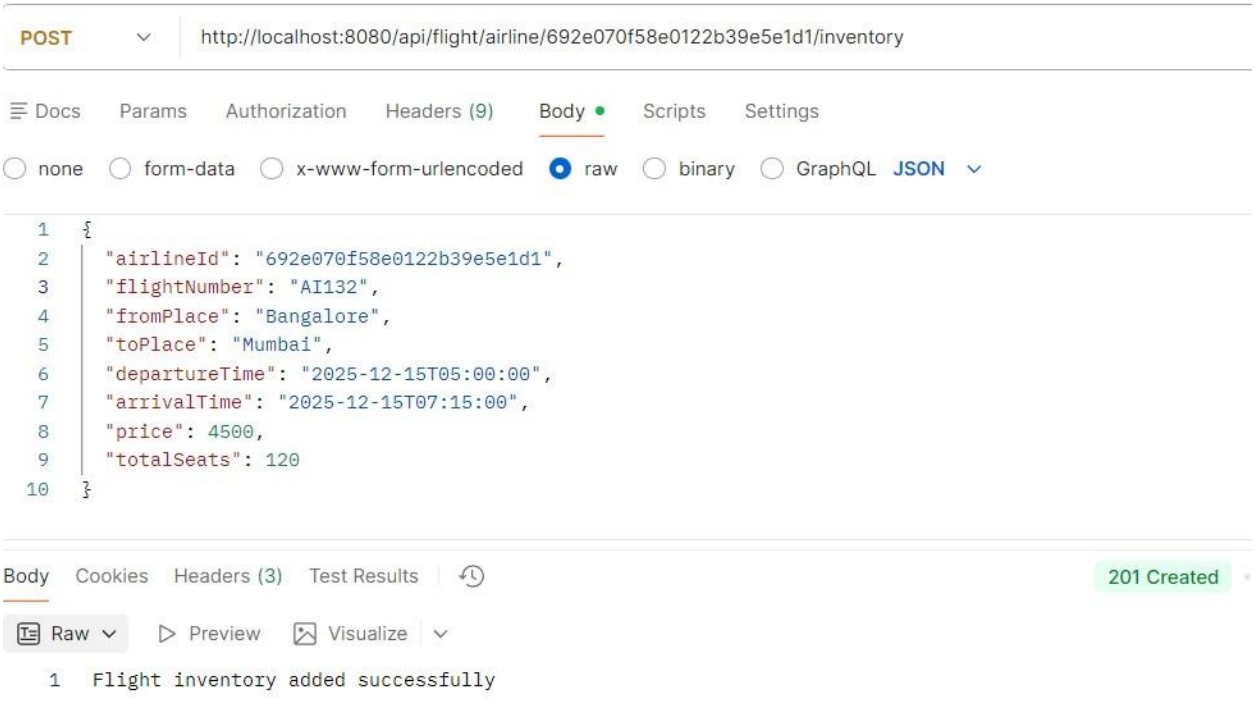
Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	510	0	0.00%	4.87	2	32	4.00	7.00	7.00	10.00	51.88	28.37	7.36
GET Booking History	170	0	0.00%	3.65	2	7	4.00	4.00	5.00	7.00	17.36	10.63	2.46
GET Flight Details	170	0	0.00%	4.64	3	32	4.00	5.00	5.00	32.00	17.31	6.30	2.62
GET Ticket	170	0	0.00%	6.34	4	10	6.00	7.90	8.00	10.00	17.36	11.53	2.31

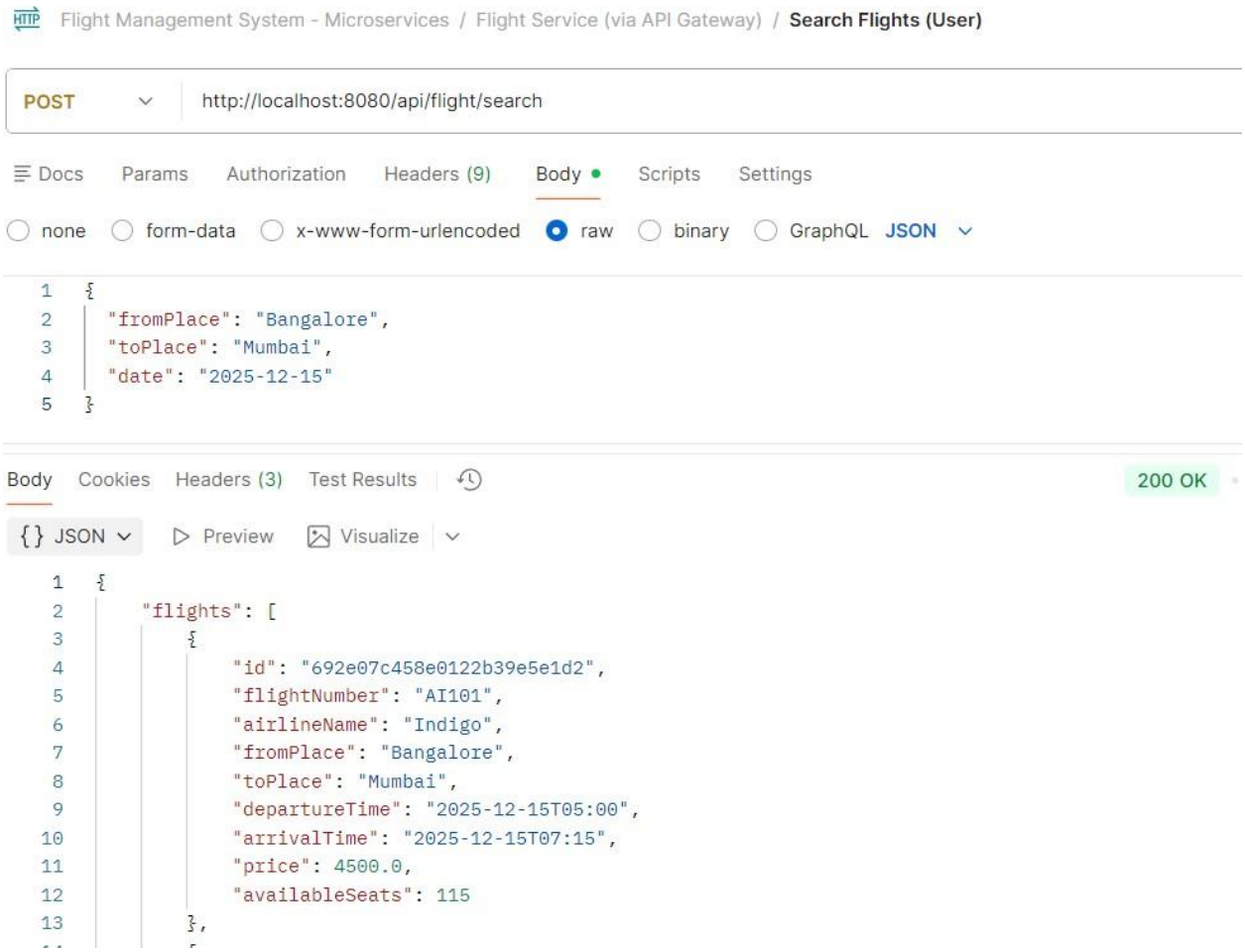
API Endpoints POST Add Airline :-



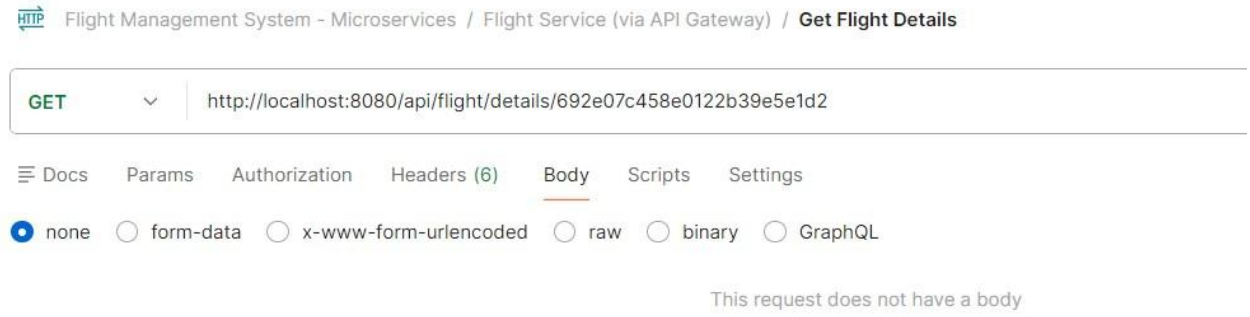
POST Add Flight Inventory

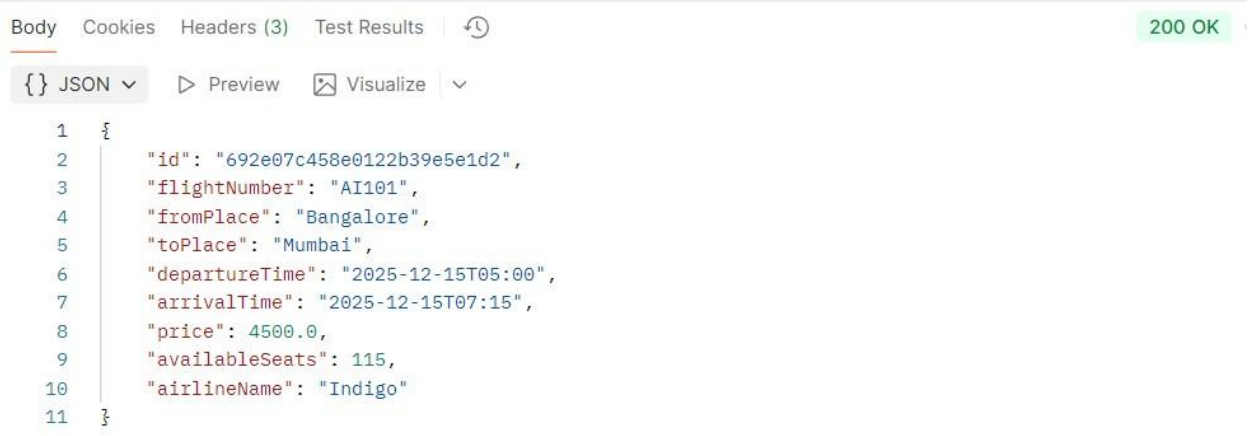


POST Search flights



GET Get Flight Details





POST Book Seat in Flight

Flight Management System - Microservices / Booking Service (via API Gateway) / **Book Flight**

POST http://localhost:8080/api/book/book

Docs | Params | Authorization | Headers (9) | Body | Scripts | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL **JSON**

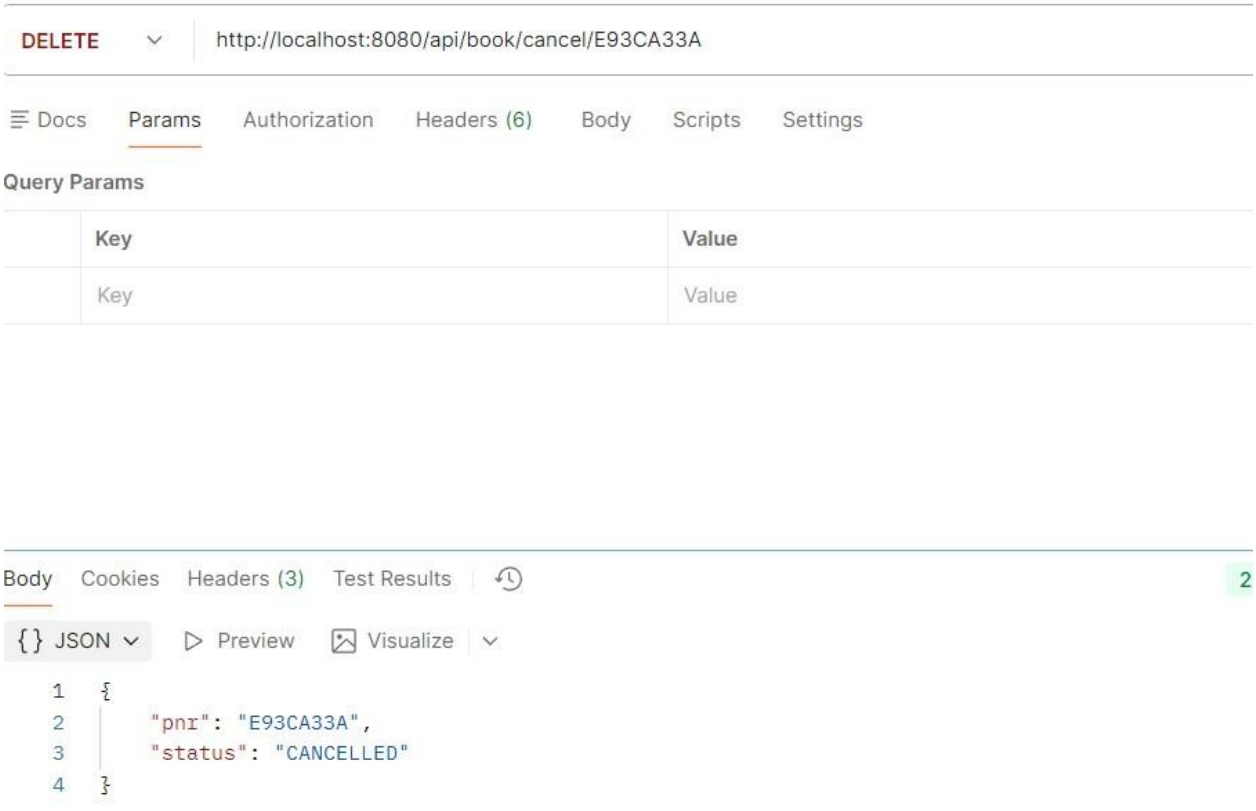
```
{
  "flightId": "692e07c458e0122b39e5e1d2",
  "email": "joel@example.com",
  "passengers": [
    {
      "name": "Joel",
      "gender": "Male",
      "age": 22,
      "seatNumber": "1A",
      "meal": "VEG"
    }
  ]
}
```

Body | Cookies | Headers (3) | Test Results | **201 Created**

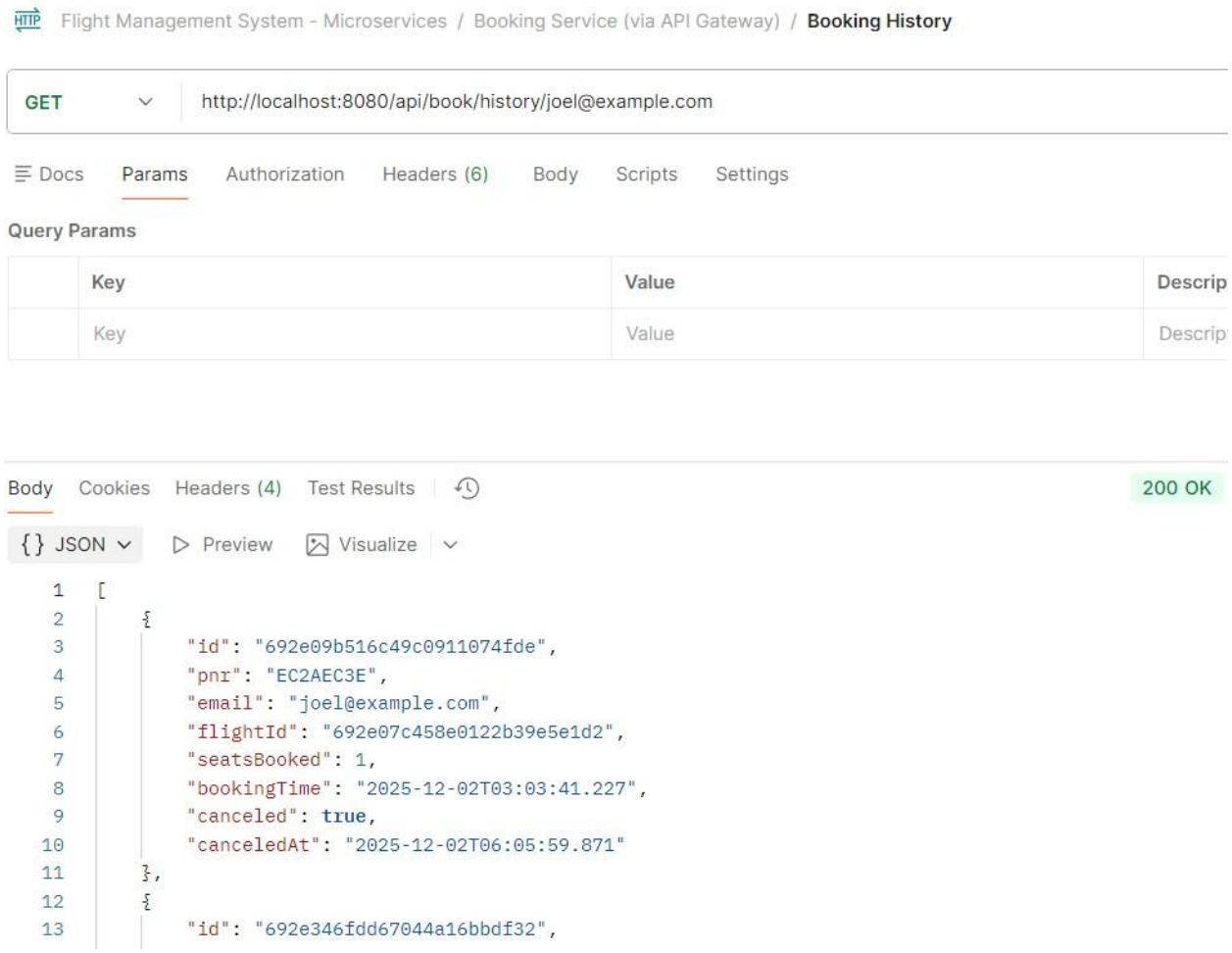
{ } JSON | Preview | Visualize

```
{
    "pnr": "E93CA33A",
    "email": "joel@example.com",
    "flightId": "692e07c458e0122b39e5e1d2",
    "status": "BOOKED"
}
```

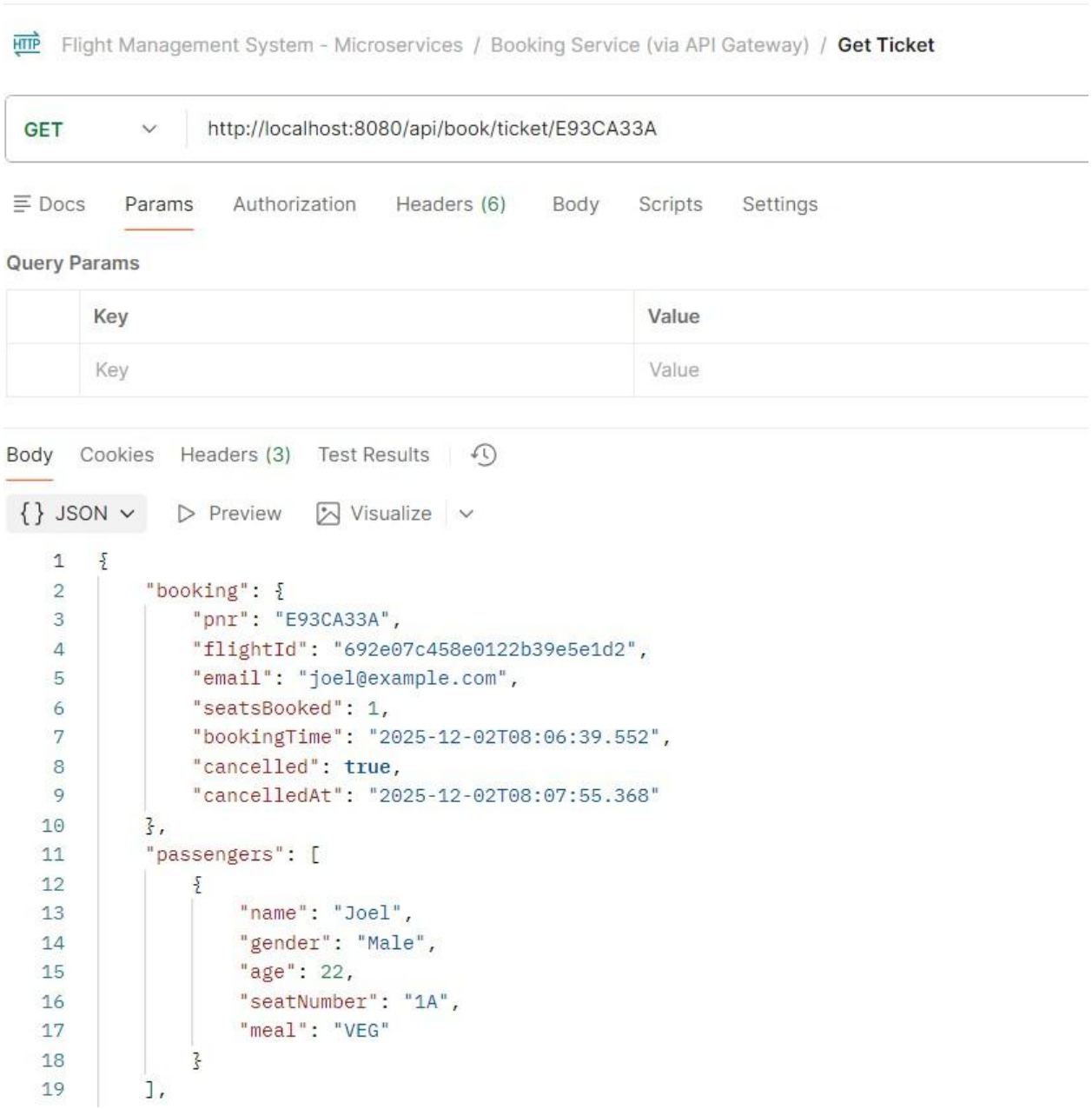
DELETE Cancel Booking



GET Get Booking History



GET Ticket Details



```
    }
],
"flight": {
    "id": "692e07c458e0122b39e5e1d2",
    "flightNumber": "AI101",
    "fromPlace": "Bangalore",
    "toPlace": "Mumbai",
    "departureTime": "2025-12-15T05:00",
    "arrivalTime": "2025-12-15T07:15",
    "price": 4500.0,
    "availableSeats": 114,
    "airlineName": "Indigo"
}
```