

Eureka Dashboard:

The screenshot shows the Spring Eureka dashboard at localhost:8761. The top navigation bar includes links for flight-config-server, tracker, Google Account, Inbox, RabbitMQ Overview, and Eureka. A "Finish update" button is visible on the right.

System Status

Environment	test	Current time	2025-12-08T19:07:04 +0530
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	10
		Renews (last min)	2

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - 10.45.64.162:api-gateway:8087
AUTH-SERVICE	n/a (1)	(1)	UP (1) - 10.45.64.162:auth-service:8085
BOOKING-SERVICE	n/a (1)	(1)	UP (1) - 10.45.64.162:booking-service:8082
CONFIG-SERVER	n/a (1)	(1)	UP (1) - 10.45.64.162:config-server:8888
FLIGHT-SERVICE	n/a (1)	(1)	UP (1) - 10.45.64.162:flight-service:8081

General Info

Name	Value
total-avail-memory	130mb
num-of-cpus	8
current-memory-usage	105mb (80%)
server-uptime	00:01

Docker:

```
ananyaarisenapalli@Ananya-MacBook-Pro-3 FlightBookingMicro % docker compose up --build
[WARN@0000] /Users/ananyaarisenapalli/workspace-spring-tools-for-eclipse-4.32.2.RELEASE/FlightBookingMicro/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 12/14
  ✓ mysql Pulled
    eebad029ad6b Pull complete                                         20.3s
  ✓ 842b9f544a00 Pull complete                                         1.1s
  ✓ f7a792cd988e Pull complete                                         8.3s
  ✓ 05b8b580cccd Pull complete                                         1.4s
  ✓ 16e98e12849f Pull complete                                         9.1s
  ✓ bc6ec3e9c020 Pull complete                                         1.5s
  ✓ 44803d6a5f06 Pull complete                                         13.2s
  ✓ 4b45fb5942a4 Pull complete                                         1.4s
  ✓ 42c45335a3b Pull complete                                         1.4s
  ✓ faefaae1aeel1 Pull complete                                         8.4s
  ✓ 7c8fcbb5fe6b8 Pull complete                                         1.4s
  ✓ f424b1298731 Download complete                                         0.0s
  ✓ ce3c7466b966 Download complete                                         0.0s
[+] Building 84.3s (93/93) FINISHED
```

```
[+] Running 18/18
  ✓ flightbookingmicro-flight-service      Built
  ✓ flightbookingmicro-auth-service        Built
  ✓ flightbookingmicro-notification-service Built
  ✓ flightbookingmicro-api-gateway        Built
  ✓ flightbookingmicro-service-registry   Built
  ✓ flightbookingmicro-config-server     Built
  ✓ flightbookingmicro-booking-service   Built
  ✓ Network flightbookingmicro_backend   Created
  ✓ Volume flightbookingmicro_mysql-data Created
  ✓ Container flight_rabbitmq            Created
  ✓ Container flight_mysql              Created
  ✓ Container service-registry          Created
  ✓ Container notification-service     Created
  ✓ Container config-server            Created
  ✓ Container auth-service             Created
  ✓ Container booking-service          Created
  ✓ Container flight-service           Created
  ✓ Container api-gateway             Created
Attaching to api-gateway, auth-service, booking-service, config-server, flight-service, flight_mysql, flight_rabbitmq, notification-service, service-registry
Error response from daemon: ports are not available: exposing port TCP 0.0.0.0:3306 -> 127.0.0.1:0: listen tcp 0.0.0.0:3306: bind: address already in use
```

The screenshot shows the Docker Desktop application window. The left sidebar contains icons for Ask Gordon (Beta), Containers, Images, Volumes, Kubernetes, and Builds (selected). The main area is titled "Builds" with a "Give feedback" link. It features a purple banner encouraging users to "Build large and multi-platform Docker images faster in Docker Build Cloud." Below the banner, tabs for "Build history" (selected) and "Active builds" are shown, along with a search bar and filters for "Show only my builds". A table lists ten build entries, each with a checkbox, name, ID, builder (desktop-linux), duration, creation time, author (N/A), and a trash icon. The table includes pagination at the bottom.

	Name	ID	Builder	Duration	Created	Author
<input type="checkbox"/>	flight-service	td0jp6	desktop-linux	1m 23s	3 minutes ago	N/A
<input type="checkbox"/>	api-gateway	7jvypa	desktop-linux	1m 15s	3 minutes ago	N/A
<input type="checkbox"/>	auth-service	81f9lx	desktop-linux	1m 20s	3 minutes ago	N/A
<input type="checkbox"/>	config-server	ja84hb	desktop-linux	1m 15s	3 minutes ago	N/A
<input type="checkbox"/>	service-registry	wmfeym	desktop-linux	1m 17s	3 minutes ago	N/A
<input type="checkbox"/>	notification-service	22qnIm	desktop-linux	56.4s	3 minutes ago	N/A
<input type="checkbox"/>	booking-service	qnqk7p	desktop-linux	1m 23s	3 minutes ago	N/A
<input type="checkbox"/>	FlightBookingSystem-WebFlux	1lf4cj	desktop-linux	27.8s	2 days ago	N/A
<input type="checkbox"/>	FlightBookingSystem-WebFlux	qe7896	desktop-linux	26.3s	2 days ago	N/A
<input type="checkbox"/>	FlightBookingSystem-WebFlux	tgc908	desktop-linux	51.1s	2 days ago	N/A

Rows per page: 10 ▾ 1–10 of 11 > >

Engine running | RAM 3.56 GB CPU 0.50% Disk: 10.54 GB used (limit 223.63 GB) > Terminal v4.54.0

The screenshot shows the Docker Desktop application window. On the left, a sidebar navigation bar includes links for Ask Gordon (Beta), Containers, Images (selected), Volumes, Kubernetes, and Builds. Below these are Models, MCP Toolkit (Beta), Docker Hub, Docker Scout, and Extensions. The main content area is titled "Images" with a "Local" tab selected. It displays a table of 14 images, each with columns for Name, Tag, Image ID, Created, Size, and Actions. The images listed include rabbitmq, tomcat, mongo, hello-world, mcr.microsoft.com/azure-sql-edge, mysql, flightbookingmicro-notification-service, flightbookingmicro-configuration-server, flightbookingmicro-api-gateway, flightbookingmicro-service-registry, flightbookingmicro-auth-service, flightbookingmicro-flight-service, and flightbookingmicro-booking-service. The "flightbooking" images are highlighted in red. At the bottom of the table, it says "Showing 14 items". The status bar at the bottom indicates "Engine running", system resources (RAM 3.53 GB, CPU 1.00%, Disk 10.54 GB used / 223.63 GB), and the version "v4.54.0".

	Name	Tag	Image ID	Created	Size	Actions
	rabbitmq	3-management	e582c0bc7766	7 days ago	407.99 MB	
	tomcat	latest	3d41d69cc052	25 days ago	615.75 MB	
	mongo	6.0	0a83b2f824b2	2 months ago	1 GB	
	hello-world	latest	f7931603f70e	4 months ago	22.51 KB	
	mcr.microsoft.com/azure-sql-edge	latest	902628a8be89	3 years ago	2.52 GB	
	mysql	8.0	0275a35e79c6	6 days ago	1.06 GB	
	flightbookingmicro-notification-service	latest	94fefef11c563	3 minutes ago	435.16 MB	
	flightbookingmicro-configuration-server	latest	9fae14d5a33b	2 minutes ago	502.94 MB	
	flightbookingmicro-api-gateway	latest	21cf08c6bdef	2 minutes ago	500.51 MB	
	flightbookingmicro-service-registry	latest	102563c34520	2 minutes ago	509.1 MB	
	flightbookingmicro-auth-service	latest	43ab705dd02b	2 minutes ago	547.38 MB	
	flightbookingmicro-flight-service	latest	835f657d3ea7	2 minutes ago	546 MB	
	flightbookingmicro-booking-service	latest	a673593fcac8	2 minutes ago	553.42 MB	

Docker logs:

```
[ananyaarisenapalli@ananya-MacBook-Pro-3 FlightBookingMicro % docker compose up --build >&1 | tee parent.log.log
time="2025-12-08T23:13:10+05:30" level=warning msg="/Users/ananyaarisenapalli/workspace-spring-tools-for-eclipse-4.32.2.RELEASE/FlightBookingMicro/docker-compose.yml: the attribute 'version' is obsolete
, it will be ignored, please remove it to avoid potential confusion"
#1 [internal] load local bake definitions
#1 reading from stdin 4.57kB done
#1 DONE 0.0s

#2 [flight-service internal] load build definition from Dockerfile
#2 transferring dockerfile: 447B done
#2 DONE 0.0s

#3 [api-gateway internal] load build definition from Dockerfile
#3 transferring dockerfile: 447B done
#3 DONE 0.0s

#4 [config-server internal] load build definition from Dockerfile
#4 transferring dockerfile: 447B 0.0s done
#4 DONE 0.0s

#5 [notification-service internal] load build definition from Dockerfile
#5 transferring dockerfile: 447B 0.0s done
#5 DONE 0.0s

#6 [booking-service internal] load build definition from Dockerfile
#6 transferring dockerfile: 447B done
#6 DONE 0.0s

#7 [auth-service internal] load build definition from Dockerfile
#7 transferring dockerfile: 447B done
#7 DONE 0.0s

#8 [service-registry internal] load build definition from Dockerfile
#8 transferring dockerfile: 447B done
#8 DONE 0.0s
```

MySQL Screenshots: Flight Database:

The screenshot shows the MySQL Workbench interface for a local instance (3306). The left sidebar contains navigation links for Administration, Schemas, Management, Performance, and Object Info. The main area displays a query editor with the following SQL code:

```
show tables;
select * from flights;
select * from flight_seats;
use bookingdb_micro;
show tables;
```

The Result Grid shows the data for the 'flights' table:

ID	Airline_Logo_URL	Airline_Name	Arrival_Time	Departure_Time	Destination	Flight_Number	Origin	Price	Total_Seats	Trip_Type
1	https://example.com/logo.png	AirExample	2025-12-15 05:30:00.000000	2025-12-15 04:00:00.000000	BOM	AI-101	HYD	4500	120	ONEW
2	http://example.com/logo.png	Air India	2025-12-10 07:00:00.000000	2025-12-10 04:30:00.000000	DEL	AI101	HYD	4500	120	ONEW
3	https://indigo.in/logo.png	Indigo Airlines	2025-12-15 05:25:00.000000	2025-12-15 03:15:00.000000	MUM	8E302	BLR	3800.5	150	ROUND
4	https://indigo.in/logo.png	Indigo Airlines	2025-12-15 05:25:00.000000	2025-12-15 03:15:00.000000	MUM	8E302	BLR	3800.5	150	ROUND
5	https://indigo.in/logo.png	Indigo Airlines	2025-12-15 05:25:00.000000	2025-12-15 03:15:00.000000	MUM	AE302	BLR	3800.5	150	ROUND
6	https://indigo.in/logo.png	Indigo Airlines	2025-12-15 05:25:00.000000	2025-12-15 03:15:00.000000	MUM	AE302	BLR	3800.5	150	ROUND
7	https://indigo.in/logo.png	Indigo Airlines	2025-12-15 05:25:00.000000	2025-12-15 03:15:00.000000	MUM	AE302	BLR	3800.5	150	ROUND
8	https://spicejet.aero/logo.png	SpiceJet	2025-12-12 02:20:00.000000	2025-12-12 00:50:00.000000	CCU	SG409	MAA	5200	50	ONEW
9	http://images.vistara.com/logo.png	Vistara	2025-12-15 05:15:00.000000	2025-12-15 03:00:00.000000	BOM	UK979	BLR	5200	180	ONEW
10	http://airasia/logo.png	AirAsia	2025-12-18 11:30:00.000000	2025-12-18 09:30:00.000000	IOM	IS115	BLR	2500	150	ONEW
11	http://airasia/logo.png	AirAsia	2025-12-18 11:30:00.000000	2025-12-18 09:30:00.000000	IHYD	IS115	BLR	2000	100	ONEW
12	http://airasia/logo.png	AirAsia	2025-12-18 11:30:00.000000	2025-12-18 09:30:00.000000	CHENNAI	IS115	MUM	2000	100	ONEW
13	http://airasia/logo.png	AirAsia	2025-12-18 11:30:00.000000	2025-12-18 09:30:00.000000	PJB	IS115	DEL	1000	200	ONEW

The Action Output section shows the results of the last two queries:

Action	Time	Response	Duration / Fetch Time
use flightdb_micro	23:07:27	0 row(s) affected	0.068 sec
select * from flights LIMIT 0, 1000	23:07:29	13 row(s) returned	0.030 sec / 0.000016...

Query Completed

Local instance 3306 - Warning - not supported

Administration **Schemas**

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info **Session**

No object selected

```

54 • use flightdb_micro;
55 • show tables;
56 • select * from flights;
57 • select * from flight_seats;
58 • use bookingdb_micro;
100% ◇ 28:57

```

Query 1 **SQL File 3***

Limit to 1000 rows

Result Grid **Filter Rows:** **Search:** **Edit:** **Export/Import:**

ID	Passenger Name	Seat Number	Status	Flight ID
1	HULL	1A	AVAILABLE	1
2	HULL	1B	AVAILABLE	1
3	HULL	1C	AVAILABLE	1
4	HULL	1D	AVAILABLE	1
5	HULL	1A	AVAILABLE	2
6	HULL	1B	AVAILABLE	2
7	HULL	1C	AVAILABLE	2
8	HULL	2A	AVAILABLE	3
9	HULL	2B	AVAILABLE	3
10	HULL	2C	AVAILABLE	3
11	HULL	3A	AVAILABLE	3
12	HULL	3B	AVAILABLE	3
13	HULL	2A	AVAILABLE	4
14	HULL	2B	AVAILABLE	4
15	HULL	2C	AVAILABLE	4
16	HULL	3A	AVAILABLE	4
17	HULL	3B	AVAILABLE	4
18	HULL	2A	AVAILABLE	5
19	HULL	2B	AVAILABLE	5
20	HULL	2C	AVAILABLE	5
21	HULL	3A	AVAILABLE	5
22	HULL	RR	AVAILABLE	5

flight_seats 2

Action Output

Time	Action	Response	Duration / Fetch Time
23:07:27	use flightdb_micro	0 row(s) affected	0.068 sec
23:07:29	select * from flights LIMIT 0, 1000	13 row(s) returned	0.030 sec / 0.000016...
23:07:57	select * from flight_seats LIMIT 0, 1000	812 row(s) returned	0.0067 sec / 0.00051...

Query Completed

Booking Database:

Local instance 3306 - Warning - not supported

Administration **Schemas**

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info **Session**

No object selected

```

57 • select * from flight_seats;
58 • use bookingdb_micro;
59 • show tables;
60 • select * from bookings;
61 • select * from passengers;
100% ◇ 24:60

```

Query 1 **SQL File 3***

Limit to 1000 rows

Result Grid **Filter Rows:** **Search:** **Edit:** **Export/Import:**

ID	Cancelled At	Created At	Flight ID	Num Seats	PNR	Status	Total Pri...	User Email	Version
27	HULL	2025-12-04 05:35:20.066070	2	1	8894CA5C	ACTIVE	4500	test@gmail.com	HULL
28	HULL	2025-12-04 05:35:37.765510	2	1	5F7B752D	ACTIVE	4500	test@gmail.com	HULL
29	HULL	2025-12-04 05:35:39.163333	2	1	D5BD778B	ACTIVE	4500	test@gmail.com	HULL
30	HULL	2025-12-08 06:41:01.318193	10	2	6BEE729F	ACTIVE	0	alice@example.com	0
31	2025-12-08 07:11:48.069455	2025-12-08 06:52:41.076648	10	2	B99C79D2	CANCELLED	0	charlie@example.com	0
32	HULL	2025-12-08 06:57:39.187867	10	2	CAF0C0EF	ACTIVE	5000	alice@example.com	0
33	HULL	2025-12-08 07:43.436378	10	2	D689C213	ACTIVE	5000	alice@example.com	0
34	HULL	2025-12-08 07:45.03339	10	2	40FCF6AA	ACTIVE	5000	alice@example.com	0
35	HULL	2025-12-08 07:12:52.752340	10	2	5314295D	ACTIVE	5000	alice@example.com	0
36	2025-12-08 14:49:42.541712	2025-12-08 09:22:09.595054	10	1	27690E8C	CANCELLED	2500	ananyaairsenapalli...	0
37	2025-12-08 10:30:56.197468	2025-12-08 09:47.300639	10	1	5E9A6C01	CANCELLED	2500	ananyaairsenapalli...	0
38	HULL	2025-12-08 10:50:01.100000	10	1	5F5451FA	ACTIVE	2500	ananyaairsenapalli...	0
39	HULL	2025-12-08 10:58:27.603391	10	1	68B87F75	ACTIVE	2500	ananyaairsenapalli...	0
40	HULL	2025-12-08 10:40:23.469531	10	1	E9A86722	ACTIVE	2500	chethan7330@gmail...	0
41	2025-12-08 14:46:45.452472	2025-12-08 10:47.444739	9	2	1AF828AF	CANCELLED	10400	chethan7330@gmail...	0
42	2025-12-08 11:06:04.634516	2025-12-08 10:48:27.987870	8	2	D0C49E45	CANCELLED	10400	chethan7330@gmail...	0
43	HULL	2025-12-08 10:47.51.028468	8	3	7A173AE3	ACTIVE	15600	shreyairsenapalli...	0
44	HULL	2025-12-08 11:01:03.304730	8	1	30EC942C	ACTIVE	5200	pakalukurati@gmail...	0
45	2025-12-08 14:43:45.771627	2025-12-08 10:42:54.629424	10	1	348FFAFA	CANCELLED	2500	pakalukurati@gmail...	0
46	HULL	2025-12-08 14:46:50.334451	10	1	E98F03E1	ACTIVE	2500	pakalukurati@gmail...	0
47	2025-12-08 15:34:36.604753	2025-12-08 14:48:56.772660	10	1	137A7F16	ACTIVE	2500	ananyaairsenapalli...	0
48	HULL	2025-12-08 15:34:36.301343	10	1	137A7F16	ACTIVE	2500	ananyaairsenapalli...	0

bookings 3

Action Output

Time	Action	Response	Duration / Fetch Time
23:07:27	use flightdb_micro	0 row(s) affected	0.068 sec
23:07:29	select * from flights LIMIT 0, 1000	13 row(s) returned	0.030 sec / 0.000016...
23:07:57	select * from flight_seats LIMIT 0, 1000	812 row(s) returned	0.0067 sec / 0.00051...
23:08:05	use bookingdb_micro	0 row(s) affected	0.00068 sec
23:08:09	select * from bookings LIMIT 0, 1000	47 row(s) returned	0.0024 sec / 0.00002...

Query Completed

Local instance 3306 - Warning - not supported

Administration Schemas **Query 1** SQL File 3* Limit to 1000 rows Context Help Snippets

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info Session

No object selected

```

58 • use bookingdb_micro;
59 • show tables;
60 • select * from bookings;
61 • select * from passengers;

```

100% 26:61

Result Grid Filter Rows: Search Edit: Export/Import:

id	age	gender	meal_preferenc...	passenger_name	seat_number	booking_id
3	28	F	VEG	Alice	1A	2
4	30	M	VEG	Bob	1B	2
5	20	M	VEG	Bob	3A	3
6	20	M	VEG	Bob	3A	4
7	23	F	VEG	Anu	1A	5
8	20	M	VEG	Bob	3A	6
9	20	M	VEG	Bob	3A	12
10	20	M	VEG	Bob	3A	11
11	20	M	VEG	Bob	3A	1
12	20	M	VEG	Bob	3A	14
13	20	M	VEG	Bob	3A	13
14	20	M	VEG	Bob	3A	9
15	20	M	VEG	Bob	3A	10
16	20	M	VEG	Bob	3A	15
17	20	M	VEG	Bob	3A	8
18	20	M	VEG	Bob	3A	16
19	20	M	VEG	Bob	3A	17
20	20	M	VEG	Bob	3A	18
21	20	M	VEG	Bob	3A	19
22	20	M	VEG	Bob	3A	20
23	20	M	VEG	Bob	3A	21
24	20	M	VEG	Roh	4A	22

passengers 4

Action Output

Time	Action	Response	Duration / Fetch Time
23:07:27	use flightdb_micro	0 row(s) affected	0.068 sec
23:07:29	select * from flights LIMIT 0, 1000	13 row(s) returned	0.030 sec / 0.000016...
23:07:57	select * from flight_seats LIMIT 0, 1000	812 row(s) returned	0.0067 sec / 0.00051...
23:08:05	use bookingdb_micro	0 row(s) affected	0.00068 sec
23:08:09	select * from bookings LIMIT 0, 1000	47 row(s) returned	0.0024 sec / 0.00002...
23:08:20	select * from passengers LIMIT 0, 1000	58 row(s) returned	0.0031 sec / 0.00002...

Query Completed

Auth Database:

Local instance 3306 - Warning - not supported

Administration Schemas **Query 1** SQL File 3* Limit to 1000 rows Context Help Snippets

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info Session

No object selected

```

62
63 • use authdb;
64 • show tables;
65 • select * from users;

```

100% 21:65

Result Grid Filter Rows: Search Edit: Export/Import:

id	email	password	role
2	alice@example.com	\$2a\$10\$A15Kce8MTnFO/PaTuVXQluiqY9XV...	ROLE_USER
3	privy@example.com	\$2a\$10\$DDPmnb84qL50vOTWUqezXD1a3e7...	ROLE_USER
4	Rahul@example.com	\$2a\$10\$JmDqtp1tRISX70HeRE2elOOj5BF...	ROLE_USER
5	Riya@example.com	\$2a\$10\$P32Zt1S3dyDv9g83ta4ULTk7QdUmW...	ROLE_ADMIN

users 5

Action Output

Time	Action	Response	Duration / Fetch Time
23:07:27	use flightdb_micro	0 row(s) affected	0.068 sec
23:07:29	select * from flights LIMIT 0, 1000	13 row(s) returned	0.030 sec / 0.000016...
23:07:57	select * from flight_seats LIMIT 0, 1000	812 row(s) returned	0.0067 sec / 0.00051...
23:08:05	use bookingdb_micro	0 row(s) affected	0.00068 sec
23:08:09	select * from bookings LIMIT 0, 1000	47 row(s) returned	0.0024 sec / 0.00002...
23:08:20	select * from passengers LIMIT 0, 1000	58 row(s) returned	0.0031 sec / 0.00002...
23:08:29	use authdb	0 row(s) affected	0.00077 sec
23:08:31	select * from users LIMIT 0, 1000	4 row(s) returned	0.0028 sec / 0.00001...

Query Completed

Postman Screenshots:

Flight-service:

Add Inventory-

The screenshot shows the Postman application interface. On the left, the sidebar displays collections, environments, history, and flows. The main workspace shows a collection named "flight-service(micro)". A specific POST request titled "POST add inventory" is selected. The request URL is "http://localhost:8081/api/flight/airline/inventory/add". The request body is set to "raw" JSON and contains the following payload:

```
1 {
2   "airline": "AirAsia",
3   "airlineLogoUrl": "http://airasia/logo.png",
4   "flightNumber": "I515",
5   "origin": "BLR",
6   "destination": "GOI",
7   "departure": "2025-12-18T15:00:00",
8   "arrival": "2025-12-18T17:00:00",
9   "price": 2500,
10  "totalSeats": 150
11 }
```

The response section shows a 201 Created status with a response body containing:

```
1 {
2   "id": 10
3 }
```

At the bottom, there are various navigation and utility buttons.

Search Flight-

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Ananyarao Arisenapalli's Workspace' containing collections like 'API-Gateway', 'Booking-service(micro)', and 'flight-service(micro)'. The main area shows a POST request to 'http://localhost:8081/api/flight/search'. The request body is a JSON object:

```
1 {
2   "origin": "BLR",
3   "destination": "BOM",
4   "travelDate": "2025-12-15"
5 }
```

The response status is 200 OK, with a duration of 147 ms and a size of 389 B. The response body is a JSON array of flight details:

```
1 [
2   {
3     "flightId": 9,
4     "departureTime": "2025-12-15T08:30:00",
5     "arrivalTime": "2025-12-15T10:45:00",
6     "airlineName": "Vistara",
7     "airlineLogoUrl": "https://images.vistara.com/logo.png",
8     "price": 5200.0,
9     "tripType": "ONEWAY",
10    "seatsAvailable": 180
11  }
12 ]
```

Postman from CLI(Newman):

```
[ananyaarisenapalli@Ananyas-MacBook-Pro-3 Documents % newman run "flight-servcie(micro).postman_collection.json"
newman

flight-servcie(micro)

→ add inventory
  POST http://localhost:8081/api/flight/airline/inventory/add [201 Created, 204B, 763ms]

→ search flight
  POST http://localhost:8081/api/flight/search [201 Created, 383B, 39ms]

→ get flight with id
  GET http://localhost:8081/api/flight/11 [200 OK, 4.53kB, 18ms]

      iterations          | executed | failed
      :-----:             | :-----: | :----:
      requests           |       1   |     0
      test-scripts        |       0   |     0
      prerequest-scripts |       0   |     0
      assertions          |       0   |     0

total run duration: 850ms
total data received: 4.58kB (approx)
average response time: 273ms [min: 18ms, max: 763ms, s.d.: 346ms]
```

Booking-Service: Book ticket-

The screenshot shows the Postman interface with the following details:

- Collection:** Ananyarao Arisenapalli's Workspace
- Request:** POST /api/flight/booking/10
- Body (JSON):**

```
1 {
2   "numSeats": 1,
3   "userEmail": "ananyaarisenapalli@gmail.com",
4   "passengers": [
5     {
6       "name": "Priya",
7       "age": 21,
8       "gender": "F",
9       "seatNumber": "23",
10      "mealPreference": "VEG"
11    }
12  ]
13 }
```

- Response Status:** 201 Created
- Response Body (JSON):**

```
1 {
2   "pn": "5F64D1FA",
3   "flightId": 10,
4   "userEmail": "ananyaarisenapalli@gmail.com",
5   "numSeats": 1,
6   "totalPrice": 2500.0,
7   "status": "ACTIVE",
8   "createdAt": "2025-12-08T10:33:07.789896Z",
9   "passengers": [
10    {
11      "name": "Priya",
12      "age": 21,
13      "gender": "F",
14    }
15  ]
16 }
```

Search with PNR-

The screenshot shows the Postman interface with the following details:

- Collection:** Ananyarao Arisenapalli's Workspace
- Request:** GET /api/flight/ticket/91B862A2
- Body (JSON):**

```
1 {
2   "pn": "91B862A2",
3   "flightId": 2,
4   "userEmail": "alice@example.com",
5   "numSeats": 1,
6   "totalPrice": 4500.0,
7   "status": "ACTIVE",
8   "createdAt": "2025-12-04T05:32:17.631606Z",
9   "passengers": [
10    {
11      "name": "Piya",
12      "age": 21,
13      "gender": "F",
14      "seatNumber": "2A",
15      "mealPreference": "VEG"
16    }
17  ]
18 }
```

- Response Status:** 200 OK

Get History with Email-

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, Environments, History, Flows, Files (BETA).
- Request URL:** `http://localhost:8082/api/flight/booking/history/charlie@example.com`
- Method:** GET
- Headers:** None
- Body:** JSON response showing flight history for Charlie King and Dana Lee.
- Response:** 200 OK, 147 ms, 554 B

```
1 [
2   {
3     "pnr": "BB9C79D2",
4     "flightId": 10,
5     "userEmail": "charlie@example.com",
6     "numSeats": 2,
7     "totalPrice": 0.0,
8     "status": "ACTIVE",
9     "createdAt": "2025-12-08T06:50:41.076648Z",
10    "passengers": [
11      {
12        "name": "Charlie King",
13        "age": 35,
14        "gender": "M",
15        "seatNumber": null,
16        "mealPreference": "VEG"
17      },
18      {
19        "name": "Dana Lee",
20        "age": 32,
21        "gender": "F",
22        "seatNumber": null,
23        "mealPreference": "NONVEG"
24    }
25  ]
26 }
```

Delete Booking-

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, Environments, History, Flows, Files (BETA).
- Request URL:** `http://localhost:8082/api/flight/booking/cancel/91B862A2`
- Method:** DELETE
- Headers:** Host, User-Agent, Accept, Accept-Encoding, Connection, X-User-Email (set to alice@example.com)
- Body:** JSON response indicating booking cancellation.
- Response:** 200 OK, 170 ms, 246 B

```
1 {
2   "pnr": "91B862A2",
3   "message": "Booking cancelled successfully",
4   "status": "CANCELLED"
5 }
```

Newman:

```
[ananyaarisenapalli@Ananyas-MacBook-Pro-3 Documents % newman run "Booking-service(micro).postman_collection.json"
newman

Booking-service(micro)

→ book ticket
  POST http://localhost:8082/api/flight/booking/10 [201 Created, 491B, 586ms]

→ search by pnr
  GET http://localhost:8082/api/flight/ticket/91B862A2 [200 OK, 417B, 39ms]

→ search by email
  GET http://localhost:8082/api/flight/booking/history/ananyaarisenapalli@gmail.com [200 OK, 1.54kB, 59ms]

→ New Request
  DELETE http://localhost:8082/api/flight/booking/cancel/0B6AD529 [200 OK, 246B, 55ms]


```

	executed	failed
iterations	1	0
requests	4	0
test-scripts	0	0
prerequest-scripts	0	0
assertions	0	0
total run duration:	782ms	
total data received:	1.93kB (approx)	
average response time:	184ms [min: 39ms, max: 586ms, s.d.: 231ms]	

API-Gateway:

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections, environments, history, flows, and files. The main workspace shows a collection named 'Ananyaar Arisenapalli's Workspace' with several environments listed. One environment, 'API-Gateway(Flight-micro)', is selected. Under this environment, a specific request 'GET search flights' is highlighted with a blue selection bar. The request details show a POST method to 'http://localhost:8081/FLIGHT-SERVICE/api/flights/search'. The body of the request contains a JSON object: { "origin": "HYD", "destination": "DEL", "travelDate": "2025-12-10" }. The response tab shows a successful '201 Created' status with a response time of 474 ms and a response size of 338 B. The response body is displayed as JSON, showing a single flight record with details like flight ID, departure time, arrival time, airline name, price, trip type, and seats available.

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Ananyarao Arisenapalli's Workspace' with various collections, environments, flows, and files. A collection named 'API-Gateway(Flight-micro)' is expanded, showing several endpoints under 'Booking-service(micro)'. One endpoint, 'GET /get by pnr', is currently selected. The main workspace shows a 'GET' request to 'http://localhost:8081/BOOKING-SERVICE/api/flight/ticket/106A56B0'. The response status is '200 OK' with a duration of 1.67s and a size of 448B. The response body is a JSON object representing a flight ticket:

```

1  {
2    "pnr": "106A56B0",
3    "flightId": 2,
4    "userEmail": "alice@example.com",
5    "numSeats": 2,
6    "totalPrice": 9000.0,
7    "status": "CANCELLED",
8    "createdAt": "2025-11-30T06:54:04.275492Z",
9    "passengers": [
10      {
11        "name": "Alice",
12        "gender": "F",
13        "age": 28,
14        "seatNumber": "1A",
15        "mealPreference": "VEG"

```

Newman:

```
[ananyaarisenapalli@Ananyas-MacBook-Pro-3 Documents % newman run "API-Gateway(Flight-micro).postman_collection.json"

newman

API-Gateway(Flight-micro)

→ search flights
  POST http://localhost:8081/FLIGHT-SERVICE/api/flights/search [201 Created, 338B, 402ms]

→ get by pnr
  GET http://localhost:8081/BOOKING-SERVICE/api/flight/ticket/106A56B0 [200 OK, 448B, 57ms]

→ Book ticket
  POST http://localhost:8081/BOOKING-SERVICE/api/flight/booking/2 [201 Created, 429B, 49ms]


```

	executed	failed
iterations	1	0
requests	3	0
test-scripts	0	0
prerequest-scripts	0	0
assertions	0	0

```

total run duration: 538ms
total data received: 795B (approx)
average response time: 169ms [min: 49ms, max: 402ms, s.d.: 164ms]

```

Auth-Service: Role-User:

Sign-up

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Ananyaraos Arisenapalli's Workspace, Collections, Environments, History, Flows, Files (BETA), Cloud View, Find and replace, Console, Terminal.
- Request URL:** POST http://localhost:8085/auth/signup
- Body (JSON):**

```
1 {
2   "email": "alice@example.com",
3   "password": "P@ssword123",
4   "role": "ROLE_USER"
5 }
```
- Response:** 201 Created, 636 ms, 388 B. Body: {"message": "created", "email": "alice@example.com"}
- Bottom:** Runner, Start Proxy, Cookies, Vault, Trash.

Sign-In

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Ananyaraos Arisenapalli's Workspace, Collections, Environments, History, Flows, Files (BETA), Cloud View, Find and replace, Console, Terminal.
- Request URL:** POST http://localhost:8085/auth/signin
- Body (JSON):**

```
1 {
2   "email": "alice@example.com",
3   "password": "P@ssword123"
4 }
```
- Response:** 200 OK, 206 ms, 622 B. Body: {"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhbGljZUBleGFtcGxIImNvbSisImp0aSi6ImUwMDc2YzQ1LWM1MmUtNDc4MC05MmFmLT1zM2E4NGVjNTMyMSIsInJvbGVzIjoiuk9MRV9vUOVSiawiawFOIjoxNzY1mjAyNTgwLCJleHaiOjE3NjUyMDYxODB9. HEERzRGyElkX-91lRS9vIseIijLCqFpvNhtULVkm", "expiresInSeconds": 3600, "role": "ROLE_USER"}
- Bottom:** Runner, Start Proxy, Cookies, Vault, Trash.

Validate

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Shows collections, environments, history, flows, and files.
- Request URL:** `http://localhost:8085/auth/validate`
- Method:** GET
- Headers:** Host, User-Agent, Accept, Accept-Encoding, Connection, Authorization (set to `Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJhbGjZ...`)
- Body:** JSON response (200 OK):

```
1 {  
2   "valid": true,  
3   "email": "alice@example.com",  
4   "role": "ROLE_USER",  
5   "jti": "e0076c45-c52e-4780-92af-233a84ec5321"  
6 }
```
- Test Results:** 200 OK, 96 ms, 440 B

Sign-out

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Shows collections, environments, history, flows, and files.
- Request URL:** `http://localhost:8085/auth/signout`
- Method:** POST
- Headers:** Host, User-Agent, Accept, Accept-Encoding, Connection, Authorization (set to `Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJhbGjZ...`)
- Body:** JSON response (200 OK):

```
1 {  
2   "message": "signed out"  
3 }
```
- Test Results:** 200 OK, 49 ms, 358 B

Role-Admin:

Sign-up

The screenshot shows the Postman interface for a workspace named "Ananyarao Arisenapali's Workspace". The left sidebar lists various collections and environments. The main area shows a POST request to "http://localhost:8085/auth/signup" under the "auth-service" collection. The request body is set to raw JSON:

```
1 {
2   "email": "Riya@example.com",
3   "password": "P@ssword123",
4   "role": "ROLE_ADMIN"
5 }
```

The response status is 201 Created, with a message and email returned:

```
1 {
2   "message": "created",
3   "email": "Riya@example.com"
4 }
```

Sign-In

The screenshot shows the Postman interface for the same workspace. The left sidebar lists collections and environments. The main area shows a POST request to "http://localhost:8085/auth/signin" under the "auth-service" collection. The request body is set to raw JSON:

```
1 {
2   "email": "Riya@example.com",
3   "password": "P@ssword123"
4 }
```

The response status is 200 OK, returning a token and user details:

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJSaXlHQGV4YV1wbGUuY29tIiwianRpIjoiOWIwMzh1NjQtOTk5MS00OTZhLWIwNzItNmU2NTM5OGF1N2I3Iiwicm9sZXMiOiJST0xFX0FETUl0IiwiZWFOIjoxNzY1MjEwNDYwLCJleHAiOjE3NjUyMTQwNjB9.D8-EeuIZpyOH1xNR_6_2w9EWh_YWaob_38bE4bZIujc",
3   "expiresInSeconds": 3600,
4   "role": "ROLE_ADMIN"
5 }
```

Validate

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, Environments, History, Flows, Files (BETA).
- Request URL:** `http://localhost:8085/auth/validate`
- Method:** GET
- Headers:** (7)
 - Postman-Token
 - Host
 - User-Agent
 - Accept
 - Accept-Encoding
 - Connection
 - Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJSaXhQ...
- Body:** (JSON)

```
1 {  
2   "valid": true,  
3   "email": "Riya@example.com",  
4   "role": "ROLE_ADMIN",  
5   "jti": "9b038e64-9991-496a-b072-6e65398ae7b7"  
6 }
```
- Status:** 200 OK

Sign-out

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, Environments, History, Flows, Files (BETA).
- Request URL:** `http://localhost:8085/auth/signout`
- Method:** POST
- Headers:** (8)
 - Host
 - User-Agent
 - Accept
 - Accept-Encoding
 - Connection
 - Authorization: Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJSaXhQ...
- Body:** (JSON)

```
1 {  
2   "message": "signed out"  
3 }
```
- Status:** 200 OK

Newman:

```
[ananyaraoisenapalli@Ananya-MacBook-Pro-3 Documents % newman run "auth-service.postman_collection.json"
newman

auth-service

→ signup
  POST http://localhost:8085/auth/signup [201 Created, 388B, 409ms]

→ signin
  POST http://localhost:8085/auth/signin [200 OK, 622B, 103ms]

→ validate
  GET http://localhost:8085/auth/validate [200 OK, 440B, 12ms]

→ signout
  POST http://localhost:8085/auth/signout [200 OK, 358B, 4ms]

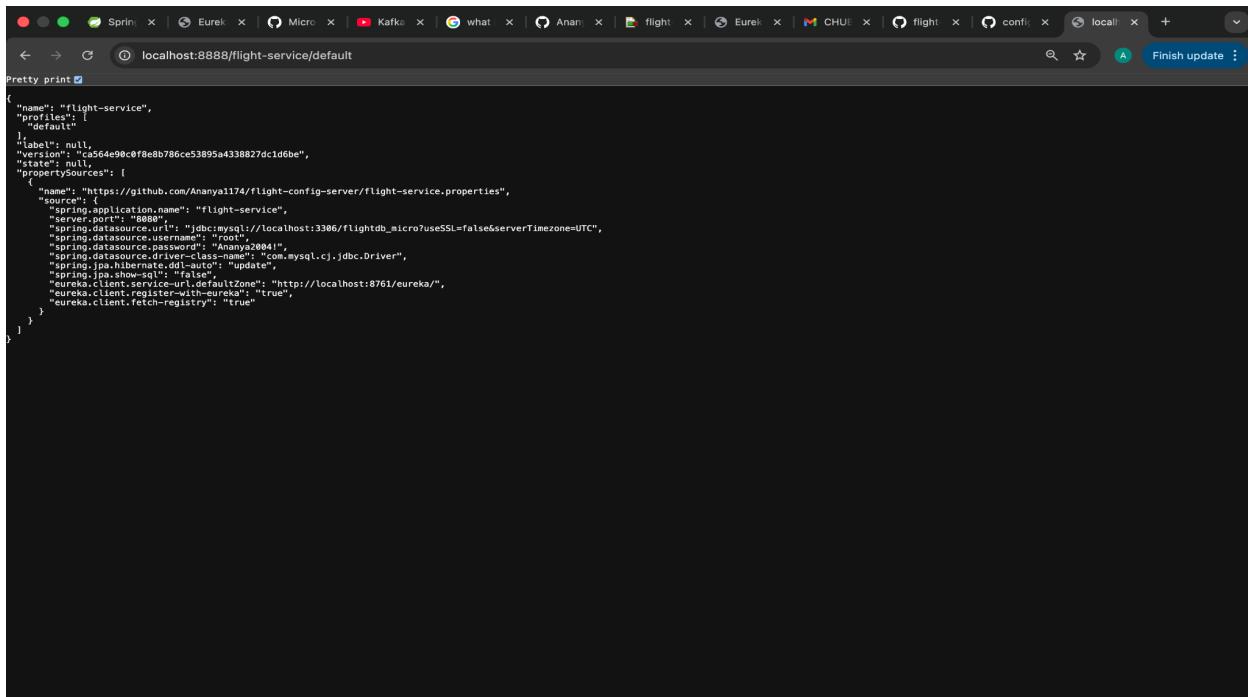
      iterations          |    executed |    failed |
      :-----:             |    :-----: |    :-----: |
        1                   |       1     |       0     |
      requests            |    executed |    failed |
      :-----:             |    :-----: |    :-----: |
        4                   |       4     |       0     |
      test-scripts         |    executed |    failed |
      :-----:             |    :-----: |    :-----: |
        0                   |       0     |       0     |
      prerequest-scripts  |    executed |    failed |
      :-----:             |    :-----: |    :-----: |
        0                   |       0     |       0     |
      assertions          |    executed |    failed |
      :-----:             |    :-----: |    :-----: |
        0                   |       0     |       0     |

total run duration: 567ms
total data received: 467B (approx)
average response time: 132ms [min: 4ms, max: 409ms, s.d.: 164ms]
```

Circuit Breaker:

The screenshot shows the Postman application interface. On the left, the sidebar displays various collections and environments. The main workspace is titled "Ananyarao Arisenapalli's Workspace". A specific collection named "Booking-service(micro)" is selected, and its "book ticket" endpoint is currently being viewed. The request method is set to "POST" and the URL is "http://localhost:8082/api/flight/booking/10". The "Body" tab is active, showing a raw JSON payload with two passengers: Alice and Bob. The response status is "503 Service Unavailable" with a timestamp of "2025-12-08T04:52:37.991544Z". The response body indicates a service unavailable error with message "Flight service unavailable. Try again later." and path "/api/flight/booking/10".

Config-Server:



```
{ "name": "flight-service", "profiles": [ "default" ], "label": null, "version": "1c564e98cef8d8b786ce53895a4338827dc1d6be", "tags": null, "propertySources": [ { "name": "https://github.com/ananya1174/flight-config-server/flight-service.properties", "source": { "spring.application.name": "flight-service", "server.port": "8888", "spring.datasource.url": "jdbc:mysql://localhost:3306/flightdb_micro?useSSL=false&serverTimezone=UTC", "spring.datasource.username": "Ananya20041", "spring.datasource.password": "Ananya20041", "spring.datasource.driver-class-name": "com.mysql.cj.jdbc.Driver", "spring.jpa.hibernate.ddl-auto": "update", "spring.jpa.show-sql": "false", "eureka.client.service-url.defaultZone": "http://localhost:8761/eureka/", "eureka.client.register-with-eureka": "true", "eureka.client.fetch-registry": "true" } } ] }
```

Message Broker-RabbitMQ:

```
ananyaarisenapalli@Ananya-MacBook-Pro-3 notification-service % java -jar target/notification-service-0.0.1-SNAPSHOT.jar
```

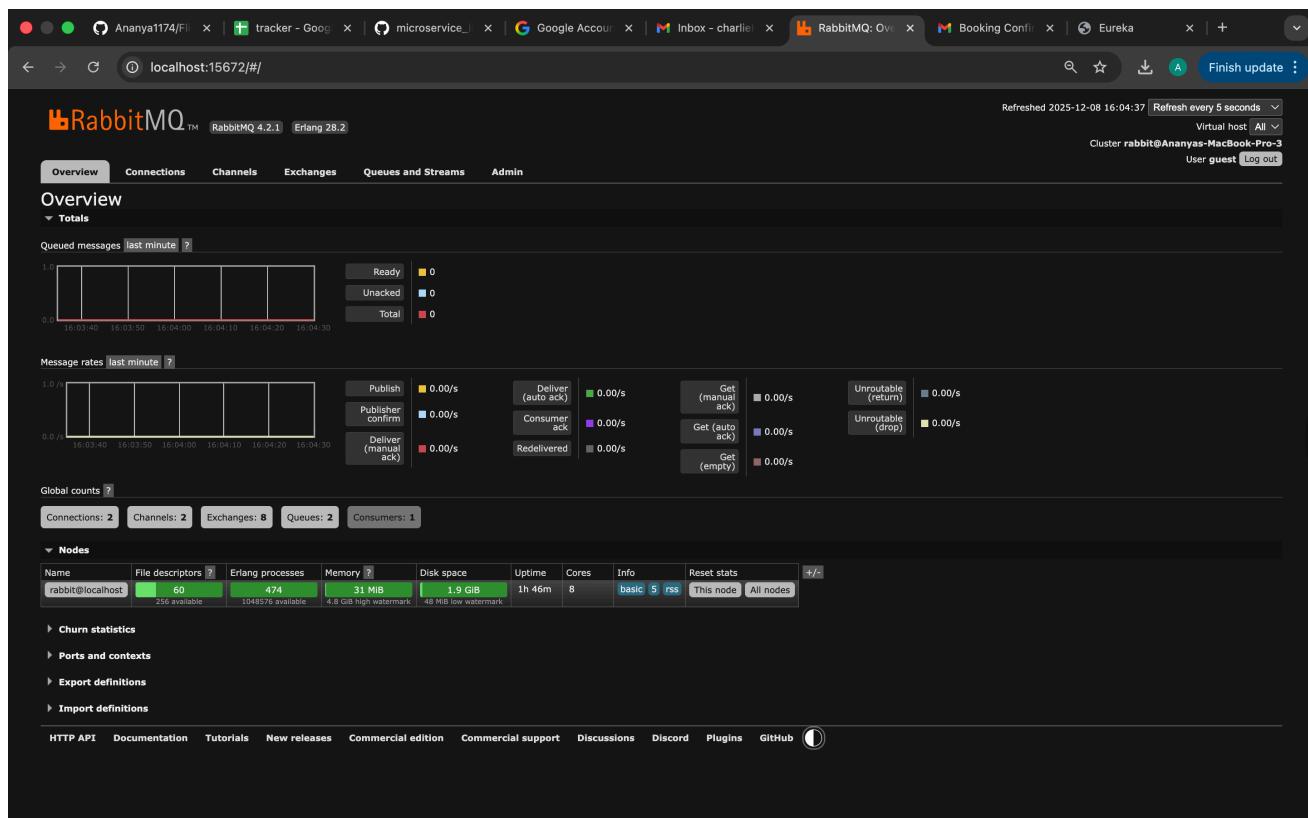


```
:: Spring Boot ::          (v4.0.0)
```

```
2025-12-08T15:59:06.459+05:30 INFO 8601 --- [notification-service] [main] c.n.NotificationServiceApplication      : Starting NotificationServiceApplication v0.0.1-SNAPSHOT using Java 21.0.1 with PID 8601 (/Users/ananyaarisenapalli/workspace-spring-tools-for-eclipse-4.32.2.RELEASE/FlightBookingMicro/notification-service/target/notification-service-0.0.1-SNAPSHOT.jar started by ananyaarisenapalli in /Users/ananyaarisenapalli/workspace-spring-tools-for-eclipse-4.32.2.RELEASE/FlightBookingMicro/notification-service)
2025-12-08T15:59:06.463+05:30 INFO 8601 --- [notification-service] [main] c.n.NotificationServiceApplication      : No active profile set, falling back to 1 default profile: "default"
2025-12-08T15:59:07.142+05:30 INFO 8601 --- [notification-service] [main] o.s.a.r.c.CachingConnectionFactory      : Attempting to connect to: [localhost:5672]
2025-12-08T15:59:07.190+05:30 INFO 8601 --- [notification-service] [main] o.s.a.r.c.CachingConnectionFactory      : Created new connection: rabbitConnectionFactory#a50d709:0/SimpleConnectio
n@1e9777898 [delegate=amqp://guest@127.0.0.1:5672/, localPort=62131]
2025-12-08T15:59:07.245+05:30 INFO 8601 --- [notification-service] [main] c.n.NotificationServiceApplication      : Started NotificationServiceApplication in 1.015 seconds (process running for 1.32)
Email sent to: ananyaarisenapalli@gmail.com
Email sent to: ananyaarisenapalli@gmail.com
```

```
2025-12-08T16:03:07.826+05:30 INFO 8833 --- [booking-service] [nio-8082-exec-5] c.bookingservice.service.BookingService : Booking saved: pnr=5F64D1FA, flightId=10, user=ananyaarisenapalli@gmail.com
2025-12-08T16:03:07.839+05:30 INFO 8833 --- [booking-service] [nio-8082-exec-5] c.bookingservice.service.BookingService : Published booking-created email for PNR 5F64D1FA
```

```
99caf0 [delegate=amqp://guest@127.0.0.1:5672/, localPort=62251]
2025-12-08T16:00:56.413+05:30 INFO 8833 --- [booking-service] [nio-8082-exec-2] c.bookingservice.service.BookingService : Published booking-cancelled email for PNR ED6FCEC1
```



Successful Booking Email-

The email inbox shows the following message:

Booking Confirmed: PNR ED6FCEC1

charliektest11@gmail.com
to me ▾

3:59 PM (0 minutes ago)

Hi,

Your booking is confirmed.

PNR: ED6FCEC1
Flight ID: 10
Seats: 1
Total: 2500.0

Passengers:
- Riya / Seat: 22

Thanks,
Booking Team

Reply Forward

Successful Booking Cancel Email-



SonarQube: Before:

The screenshot shows the SonarQube "Main Branch Summary" page for the project "flight-service". The summary indicates a "Passed" status with a green checkmark icon. Key metrics shown include:

- Security:** 0 Open issues
- Reliability:** 0 Open issues
- Maintainability:** 8 Open issues
- Accepted Issues:** 0
- Coverage:** 94.5% (No conditions set on 295 Lines to cover)
- Duplications:** 29.8% (No conditions set on 985 Lines)

The left sidebar shows navigation options like "Overview", "Main Branch", "Pull Requests", and "Branches".

After:

SonarCloud Main Branch Summary for Ananya1174 > flight-service > main. Last analysis had a warning. Quality Gate: Sonar way (Passed). Summary: 522 Lines of Code, Version 0.0.1-SNAPSHOT, Last analysis 41 seconds ago, d85c28d9.

Passed

New Code | Overall Code

Security	Reliability	Maintainability
0 Open issues (A)	0 Open issues (A)	0 Open issues (A)

Accepted Issues	Coverage	Duplications
0	82.2% (No conditions set on 170 Lines to cover)	0.0% (No conditions set on 672 Lines)

Jacoco:

Jacoco index.html report for flight-service. Coverage details:

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.flightservice.model	86%	n/a	5	36	7	53	5	36	0	2		
com.flightservice.dto	96%	n/a	4	103	4	140	4	103	0	6		
com.flightservice	37%	n/a	1	2	2	3	1	2	0	1		
com.flightservice.service	99%	71%	4	16	0	88	0	9	0	1		
com.flightservice.controller	100%	100%	0	5	0	11	0	4	0	1		
Total	38 of 906	95%	4 of 16	75%	14	162	13	295	10	154	0	11

Created with JaCoCo 0.8.10.202304240956

Jmeter:

20 requests:

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	Browse...				Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s...	Sent KB/sec	Avg. Bytes
HTTP Request	20	26	16	67	10.86	0.00%	1.1/sec	0.59	0.23	577.0
TOTAL	20	26	16	67	10.86	0.00%	1.1/sec	0.59	0.23	577.0

50 requests:

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	Browse...				Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s...	Sent KB/sec	Avg. Bytes
HTTP Request	50	19	8	137	17.22	0.00%	1.7/sec	0.96	0.38	577.0
TOTAL	50	19	8	137	17.22	0.00%	1.7/sec	0.96	0.38	577.0

100 requests:

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	Browse...				Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s...	Sent KB/sec	Avg. Bytes
HTTP Request	100	15	4	144	13.32	0.00%	1.7/sec	0.95	0.37	577.0
TOTAL	100	15	4	144	13.32	0.00%	1.7/sec	0.95	0.37	577.0