

INTRODUCTION

This microservices-based Flight Management System uses :

- 1. **Eureka Service Discovery:** For microservice registry
- 2. **OpenFeign and API Gateway:** For inter-service communication
- 3. **Resilience4j Circuit Breaker:** For fault tolerance
- 4. **Spring Cloud Config Server:** For centralized configuration
- 5. **RabbitMQ (using Docker) :** as an event-driven messaging layer (message broker)
- 6. **Notification Service (using Spring Mail)** for asynchronous email delivery.

The system persists data using **Spring Data JPA** with **MySQL**, supports validation using **Jakarta Validation**, and includes comprehensive **JUnit + Mockito** test coverage with **JaCoCo**.

EUREKA SERVER

← → ↺ ⓘ localhost:8761 🔍 ☆ 📁 👤 ⋮

Authn | edX 🎧 Lecture Videos | Intr... Analytics Academy 🌐 Machine Learning | ... 📺 Build Your First Web... 🌐 Google Cybersecuri... 📺 e 🧠 AI Ethics | IBM » 📁 All Bookmarks

System Status

Environment	test	Current time	2025-11-30T21:42:50 +0530
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	1

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - 10.6.172.123:api-gateway:8765
BOOKING-MICROSERVICE	n/a (1)	(1)	UP (1) - 10.6.172.123:booking-microservice:8082
FLIGHT-MICROSERVICE	n/a (1)	(1)	UP (1) - 10.6.172.123:flight-microservice:8081

General Info

CIRCUIT BREAKER (Resilience4j)

+

🔍 Search collections

> FMS

+ ☆ ⋮

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

🌐 FMS Microservice / Book Ticket

📄 Save ⌵

Share 📄

POST ⌵

http://localhost:8082/api/flight/booking/79448598

Send

☰ Docs

Params

Authorization

Headers (9)

Body ●

Scripts

Settings

⊙ none

⊙ form-data

⊙ x-www-form-urlencoded

⦿ raw

⊙ binary

⊙ GraphQL

JSON ⌵

🔗 Schema

Beauti

1 {

2 "email": "debashritamandal852@gmail.com",

3 "name": "Debashrita Mandal",

4 "numberOfSeats": 2,

5 "passengers": [

6 { "name": "Debashrita Mandal", "gender": "F", "age": 21 },

7 { "name": "Aahana Banerjee", "gender": "F", "age": 22 }

Body

Cookies

Headers (5)

Test Results

🕒

201 Created

• 3.10 s

• 253 B

🌐

📄 Save Response ⌵

{ } JSON ⌵

▶ Preview

🖼 Visualize ⌵

1 {

2 "message": "Cannot book right now: Flight service is unavailable",

3 "status": "FAILED"

4 }

RABBITMQ SERVER

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Containers [Give feedback](#)

Container CPU usage ⓘ
0.48% / 800% (8 CPUs available)

Container memory usage ⓘ
133.3MB / 7.36GB

Show charts

Search

Only show running containers

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
<input type="checkbox"/>	> rabbitmq	-	-	-	0.71%	<div><div></div><div></div><div></div><div></div><div></div></div>

Walkthroughs

Multi-container applications

8 mins

Containerize your application

3 mins

[View more in the Learning center](#)

Engine running

RAM 1.40 GB CPU 0.13% Disk: 1.73 GB used (limit 1006.85 GB)

v4.53.0

Registered RabbitMQ on Docker Desktop.

localhost:15672/#/

Authn | edX | Lecture Videos | Intr... | Analytics Academy | Machine Learning | ... | Build Your First Web... | Google Cybersecuri... | AI Ethics | IBM | All Bookmarks

RabbitMQ™

RabbitMQ 3.11.28 Erlang 25.3.2.9

Refreshed 2025-12-01 11:58:42

Refresh every 5 seconds

Virtual host All

Cluster rabbit@5acd0416240f

User fms_user Log out

Overview

Connections

Channels

Exchanges

Queues

Admin

Overview

Totals

Queued messages last minute ?

Currently idle

Message rates last minute ?

Currently idle

Global counts ?

Connections: 0 Channels: 0 Exchanges: 7 Queues: 0 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@5acd0416240f	38 1048576 available	0 943629 available	404 1048576 available	145 MiB 3.0 GiB high watermark	954 GiB 48 MiB low watermark	3m 58s	basic disc 2 rss	This node All nodes	

Churn statistics

Ports and contexts

Export definitions

Logged into RabbitMQ.

localhost:15672/#/queues

RabbitMQ 3.11.28 Erlang 25.3.2.9

Refreshed 2025-12-02 00:00:14 Refresh every 5 seconds

Virtual host All

Cluster rabbit@5acd0416240f

User fms_user Log out

Overview Connections Channels Exchanges Queues Admin

Queues

All queues (1)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Displaying 1 item , page size up to: 100

Overview					Messages			Message rates			
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack	
/fms	booking.email.queue	classic	D	Idle	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 Slack Plugins GitHub

Created a queue for Booking-Microservice.

mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/FMfcgzQdzcqZCFcLhQQWHkXPdVbbjfdX

Gmail

Search mail

Compose

Inbox 6,904

Starred

Snoozed

Sent

Drafts 8

Purchases 197

More

Labels +

Upgrade

Booking Confirmation - 4EA5863B

debashritamandal852@gmail.com

to me

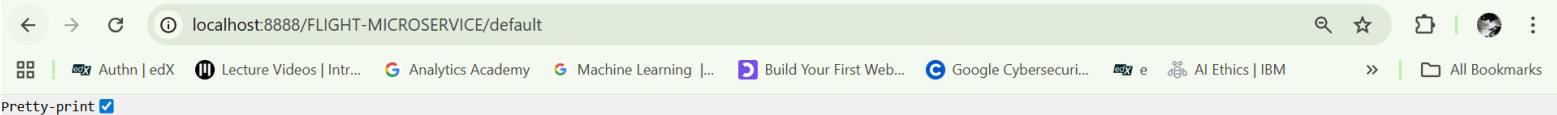
2:13 PM (0 minutes ago)

Hello Debashrita Mandal,
Your booking is confirmed.
PNR: 4EA5863B
Flight ID: 79448598

Reply Forward

Received an email after successful booking on the provided booking email id.

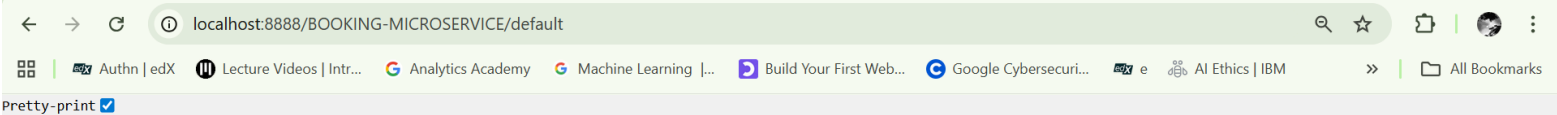
CONFIG SERVER



The screenshot shows a web browser at the URL `localhost:8888/FLIGHT-MICROSERVICE/default`. The browser's address bar and tabs are visible. Below the browser window, there is a 'Pretty-print' button and a JSON configuration for the 'FLIGHT-MICROSERVICE'.

```
{
  "name": "FLIGHT-MICROSERVICE",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "bfa1062be92f33b65632b32eadff05173493591c",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/Mad-iq/config_server_quizapp.git/FLIGHT-MICROSERVICE.properties",
      "source": {
        "spring.application.name": "FLIGHT-MICROSERVICE",
        "server.port": "8081",
        "spring.datasource.url": "jdbc:mysql://localhost:3306/flightdbnew?useSSL=false&serverTimezone=UTC",
        "spring.datasource.username": "root",
        "spring.datasource.password": "De@763071",
        "spring.jpa.hibernate.ddl-auto": "update",
        "spring.jpa.show-sql": "true",
        "eureka.client.register-with-eureka": "true",
        "eureka.client.fetch-registry": "true",
        "eureka.client.service-url.defaultZone": "http://10.6.172.123:8761/eureka/"
      }
    }
  ]
}
```

Uploaded the application.properties for Flight-microservice into the config server.



The screenshot shows a web browser at the URL `localhost:8888/BOOKING-MICROSERVICE/default`. The browser's address bar and tabs are visible. Below the browser window, there is a 'Pretty-print' button and a JSON configuration for the 'BOOKING-MICROSERVICE'.

```
{
  "name": "BOOKING-MICROSERVICE",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "bfa1062be92f33b65632b32eadff05173493591c",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/Mad-iq/config_server_quizapp.git/BOOKING-MICROSERVICE.properties",
      "source": {
        "spring.application.name": "BOOKING-MICROSERVICE",
        "server.port": "8082",
        "spring.datasource.url": "jdbc:mysql://localhost:3306/bookingdbnew?useSSL=false&serverTimezone=UTC",
        "spring.datasource.username": "root",
        "spring.datasource.password": "De@763071",
        "spring.jpa.hibernate.ddl-auto": "update",
        "spring.jpa.show-sql": "true",
        "resilience4j.circuitbreaker.instances.flightServiceCB.registerHealthIndicator": "true",
        "resilience4j.circuitbreaker.instances.flightServiceCB.slidingWindowSize": "5",
        "resilience4j.circuitbreaker.instances.flightServiceCB.minimumNumberOfCalls": "3",
        "resilience4j.circuitbreaker.instances.flightServiceCB.failureRateThreshold": "50",
        "resilience4j.circuitbreaker.instances.flightServiceCB.waitDurationInOpenState": "5s",
        "eureka.client.register-with-eureka": "true",
        "eureka.client.fetch-registry": "true",
        "eureka.client.service-url.defaultZone": "http://localhost:8761/eureka/"
      }
    }
  ]
}
```

Uploaded the application.properties for Booking-microservice into the config server.

API-GATEWAY

workspace

NewImport

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

POST add questions

GET get questions by category

GET generate random id

GET get all questions

POST get questions by id

POST score

POST create quiz

POST get quiz questions

POST Book Ticket

POST search flights

GET Search by pnr

POST Add inventory

+>

No environment

FMS Microservice / Add inventory

30 Nov 2025, 9:50 PM

Save

Share

POST

http://localhost:8765/FLIGHT-MICROSERVICE/api/flight

Send

Docs

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Beautifuly

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "airlineName": "IndiGo",

3 "source": "DELHI",

4 "destination": "MUMBAI",

5 "startDate": "2025-12-10T09:30:00",

6 "endDate": "2025-12-10T11:45:00",

7 "availableSeats": 10,

8 "ticketPrice": 4500,

9 "mealStatus": true

10 }

Body

Cookies

Headers (3)

Test Results

30 Nov 2025, 9:50 PM

201 Created

259 ms

182 B

...

{ } JSON

Preview

Visualize

...

1 {

2 "message": "Flight added successfully",

3 "flightId": "AFA9CCF7"

4 }

id View

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

POSTMAN TESTING

1. Add Inventory

workspace

NewImport

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

Quiz_t

POST scor

FMS A

POST Add

POST sea

POST Boo

GET Searc

+>

No environment

FMS Microservice / Add inventory

Save

Share

POST

http://localhost:8765/FLIGHT-MICROSERVICE/api/flight

Send

Docs

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Beautifuly

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "airlineName": "IndiGo",

3 "source": "DELHI",

4 "destination": "MUMBAI",

5 "startDate": "2025-12-10T09:30:00",

6 "endDate": "2025-12-10T11:45:00",

7 "availableSeats": 10,

8 "ticketPrice": 4500,

9 "mealStatus": true

10 }

Body

Cookies

Headers (3)

Test Results

201 Created

259 ms

182 B

Save Response

...

{ } JSON

Preview

Visualize

...

1 {

2 "message": "Flight added successfully",

3 "flightId": "AFA9CCF7"

4 }

id View

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

2. Search Flights

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

HTTP FMS Microservice / search flights

POST http://localhost:8765/FLIGHT-MICROSERVICE/api/flight/search

Send

Docs Params Authorization Headers (9) Body Scripts Settings Cookies

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

Body Cookies Headers (3) Test Results 200 OK 564 ms 307 B Save Response

JSON Preview Visualize

```
4      "availableSeats": 8,
5      "airline": "IndiGo",
6      "flightId": "79448598",
7      "availableSeatNumbers": [
8        "2B",
9        "2C",
10       "1C",
11       "2D",
12       "1D",
13       "1E",
14       "1F",
15       "2A"
16     ],
17     "price": 4500.0,
18     "dateTime": "2025-12-10T09:30"
```

Full View Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

3. Book Tickets

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

HTTP FMS Microservice / Book Ticket

POST http://localhost:8765/BOOKING-MICROSERVICE/api/flight/booking/79448598

Send

Docs Params Authorization Headers (9) Body Scripts Settings Cookies

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

Body Cookies Headers (3) Test Results 201 Created 421 ms 190 B Save Response

JSON Preview Visualize

```
5      "passengers": [
6        { "name": "Debashrita Mandal", "gender": "F", "age": 21 },
7        { "name": "Aahana Banerjee", "gender": "F", "age": 22 }
8      ],
9      "mealPreference": "VEG",
10     "seatNumbers": ["2A", "2B"]
11   }
```

Full View Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

4. Search by PNR

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

FMS Microservice / Search by pnr

Save Share

GET http://localhost:8765/BOOKING-MICROSERVICE/api/flight/ticket/D4F30643 Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

Body Cookies Headers (3) Test Results 200 OK 44 ms 409 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "pnr": "D4F30643",
3   "flightId": "79448598",
4   "status": "CONFIRMED",
5   "seatNumbers": [
6     "1A",
7     "1B"
8   ],
9   "email": "debashritamandal852@gmail.com",
10  "passengers": [
11    {
12      "id": 1,
13      "name": "Debashrita Mandal",
14      "gender": "F",
15    }
16  ]
17 }
```

id View Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

5. Search by email

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

> Spring_Mongo_Reactive

> Springboot_Mongodb

> WebfluxTutorial

FMS Microservice / Search by mail

Save Share

GET http://localhost:8765/BOOKING-MICROSERVICE/api/flight/booking/history/debashritamandal852@gr... Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

Body Cookies Headers (4) Test Results 200 OK 29 ms 377 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "email": "debashritamandal852@gmail.com",
3   "history": [
4     {
5       "status": "CONFIRMED",
6       "flightId": "79448598",
7       "pnr": "D4F30643",
8       "date": "2025-12-10"
9     },
10    {
11      "status": "CONFIRMED",
12      "flightId": "79448598",
13      "pnr": "D13A32F4",
14    }
15  ]
16 }
```

id View Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

6. Delete ticket

kspace

NewImport

+

Search collections

> FMS

> FMS Microservice

POST Add inventory

POST search flights

POST Book Ticket

GET Search by pnr

GET Search by mail

GET Delete flights

> FMS_Reactive

> FMS_Reactive_Mongo

> New Collection

> Quiz_microservices

POST add questions

GET get questions by category

GET generate random id

GET get all questions

POST get questions by id

POST score

POST create quiz

POST Book Ticke

POST search flight

GET Search by pnr

FMS Microserv

DEL Delete flight:

+ -

No environment

HTTP

FMS Microservice / Delete flights

Save

Share

DELETE

http://localhost:8082/api/flight/booking/cancel/D4F30643

Send

Docs

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

Body

Cookies

Headers (5)

Test Results

200 OK

832 ms

224 B

Save Response

{ } JSON

Preview

Visualize

1 {

2 "pnr": "D4F30643",

3 "message": "Ticket cancelled successfully"

4 }

d View

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

JACOCO AND SONARQUBE REPORTS:

FLIGHT-MICROSERVICE:

1. JACOCO REPORT
Code Coverage: 93%

File C:/Users/KIIT/Documents/workspace-spring-tools-for-eclipse-4.32.2.RELEASE/flight-microservice/target/site/jacoco/index.html

Authn | edX | Lecture Videos | Intr... | Analytics Academy | Machine Learning | ... | Build Your First Web... | Google Cybersecuri... | AI Ethics | IBM

flight-microservice

Sessions

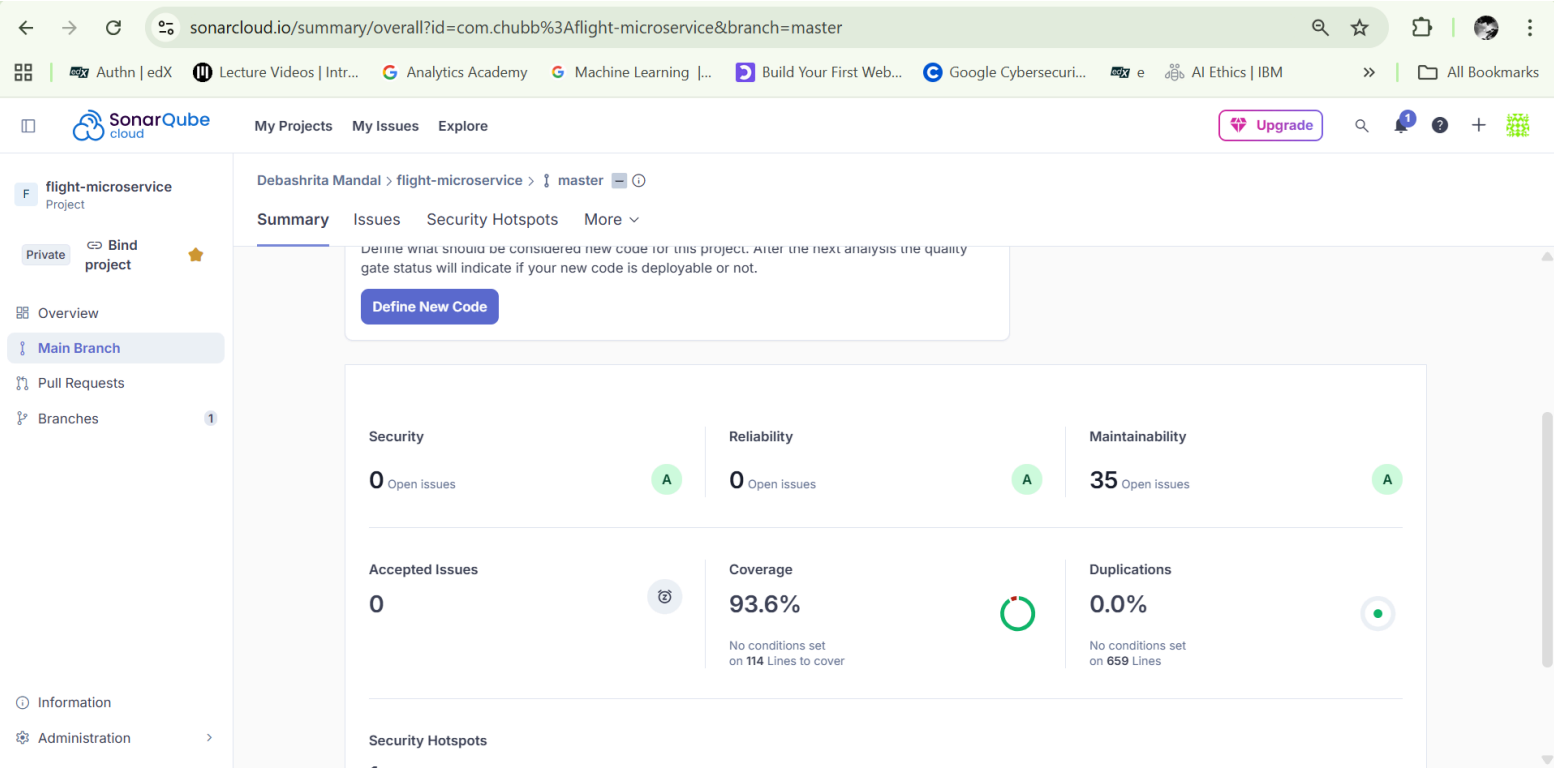
flight-microservice

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
com.flight.exception	<div></div>	82%	<div></div>	100%	1 6	2 14	1 5	0 1
com.flight	<div></div>	0%	<div></div>	n/a	2 2	3 3	2 2	1 1
com.flight.service	<div></div>	97%	<div></div>	85%	3 16	0 76	0 6	0 1
com.flight.controller	<div></div>	91%	<div></div>	100%	1 7	1 11	1 5	0 1
com.flight.model	<div></div>	100%	<div></div>	n/a	0 2	0 10	0 2	0 2
Total	32 of 532	93%	3 of 26	88%	7 33	6 114	4 20	1 6

Created with JaCoCo 0.8.10.202304240956

2. SONARQUBE REPORT

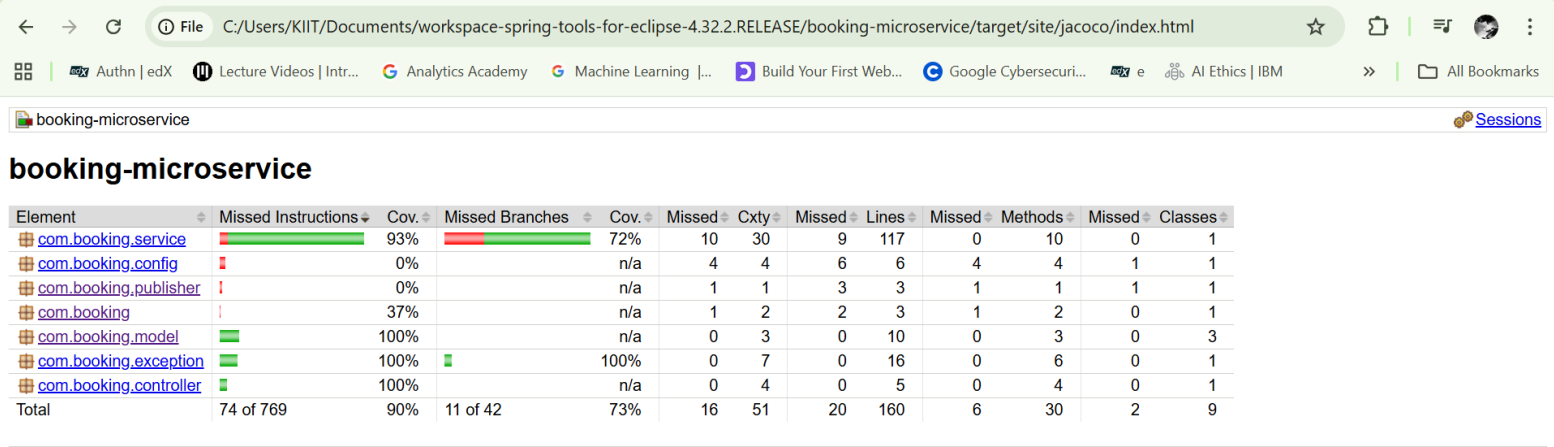
Code Coverage: 93.6%



BOOKING-MICROSERVICE

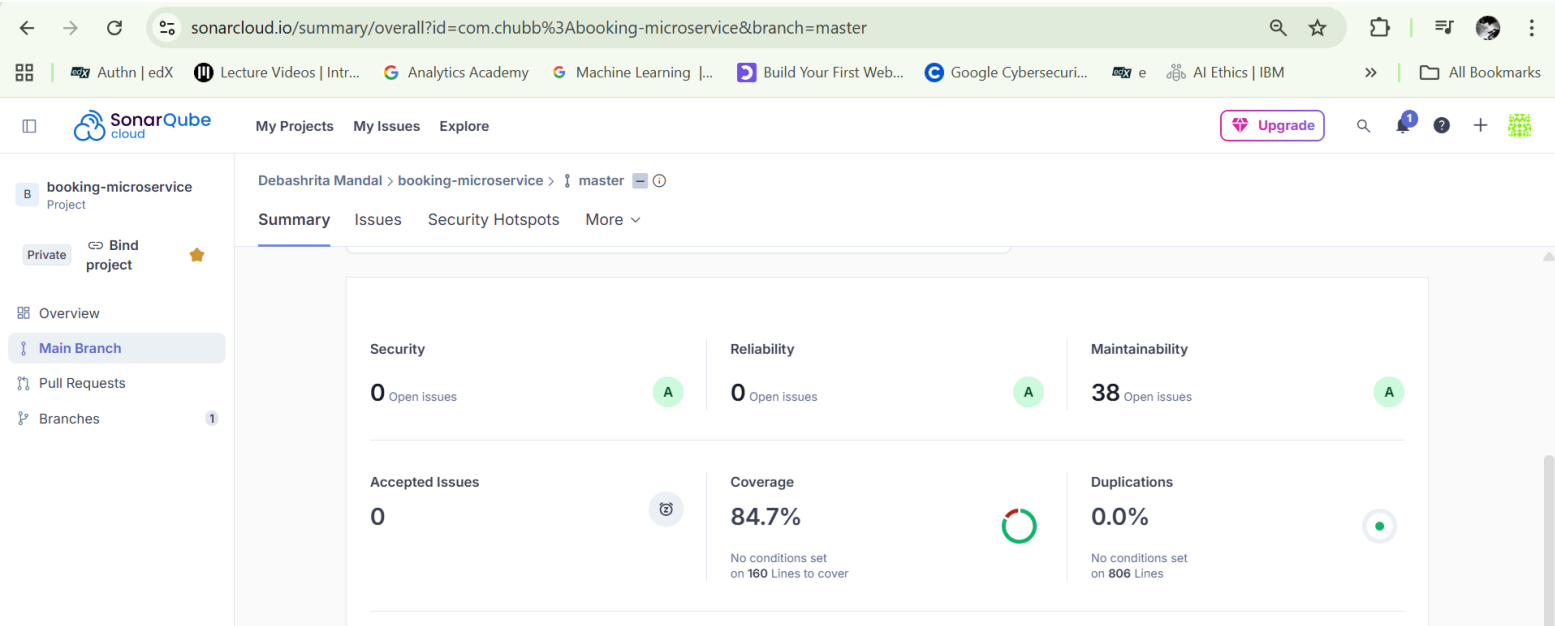
1. JACOCO REPORT:

Code Coverage: 90%



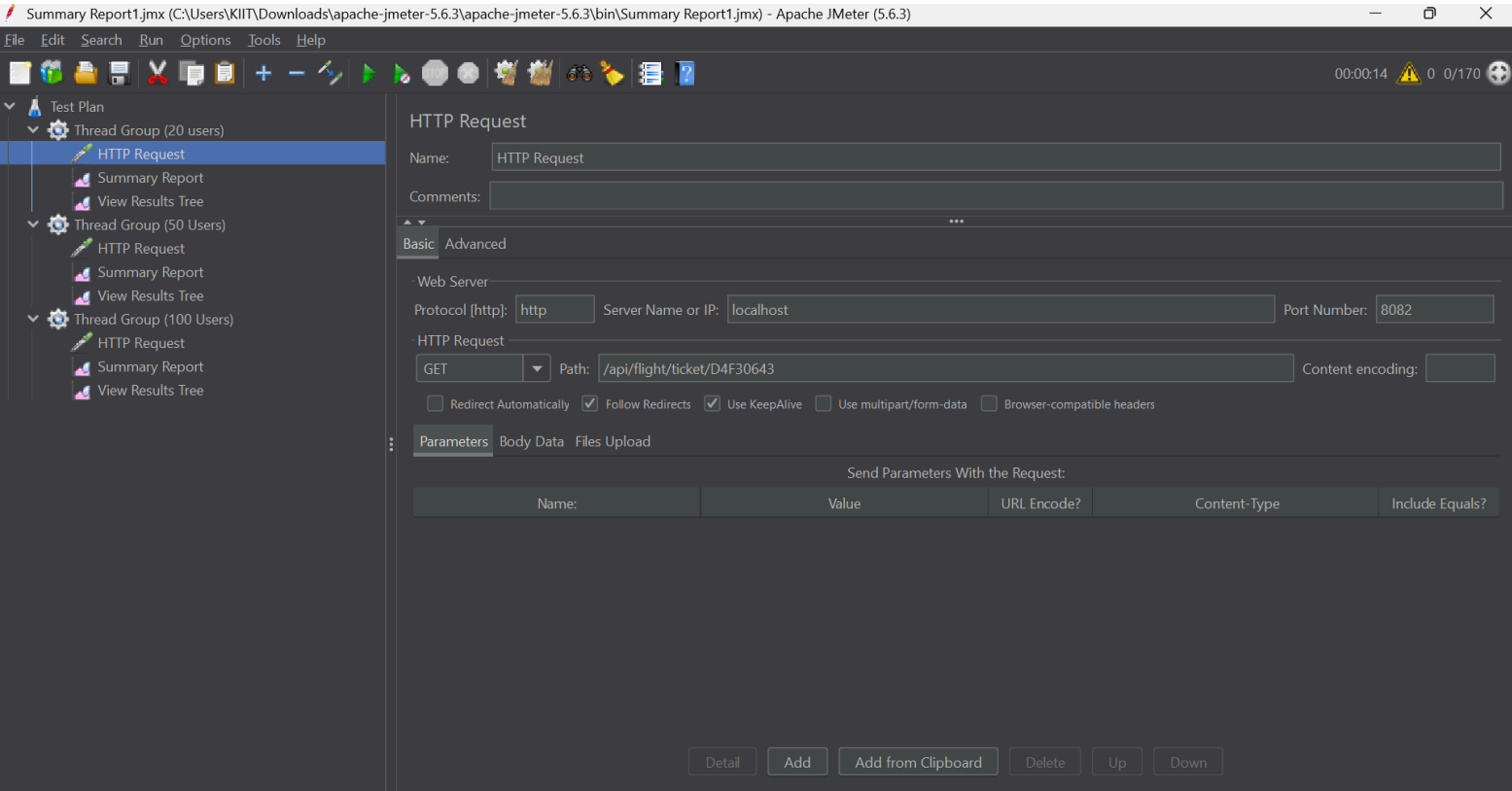
2. SONARQUBE REPORT

Code Coverage: 84.7%



JMETER TESTING

1. 20 USERS



[illegible]

- Test Plan
 - Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
 - Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
 - Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

- Test Plan
 - Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
 - Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
 - Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

The screenshot displays the Apache JMeter 5.6.3 interface. The title bar indicates the file path: "Summary Report1.jmx (C:\Users\KJIT\Downloads\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)". The top menu bar includes File, Edit, Search, Run, Options, Tools, and Help. Below the menu is a toolbar with various icons for file operations and testing. The left sidebar shows a tree view of the test plan. The main area is titled "HTTP Request" and contains configuration fields. The "Basic" tab is active, showing the "Name" field set to "HTTP Request", a "Comments" field, and a "Web Server" section with "Protocol [http:]" set to "http", "Server Name or IP:" set to "localhost", and "Port Number:" set to "8082". The "HTTP Request" section shows the "Method" as "GET" and the "Path" as "/api/flight/ticket/D4F30643". There are checkboxes for "Redirect Automatically", "Follow Redirects", "Use KeepAlive", "Use multipart/form-data", and "Browser-compatible headers". The "Parameters" tab is also visible, showing a table for sending parameters with the request. The table has columns for "Name:", "Value", "URL Encode?", "Content-Type", and "Include Equals?". At the bottom, there are buttons for "Detail", "Add", "Add from Clipboard", "Delete", "Up", and "Down".

Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:14 0 0/170

Test Plan

- Thread Group (20 users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (50 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree
- Thread Group (100 Users)
 - HTTP Request
 - Summary Report
 - View Results Tree

Summary Report

Name: Summary Report

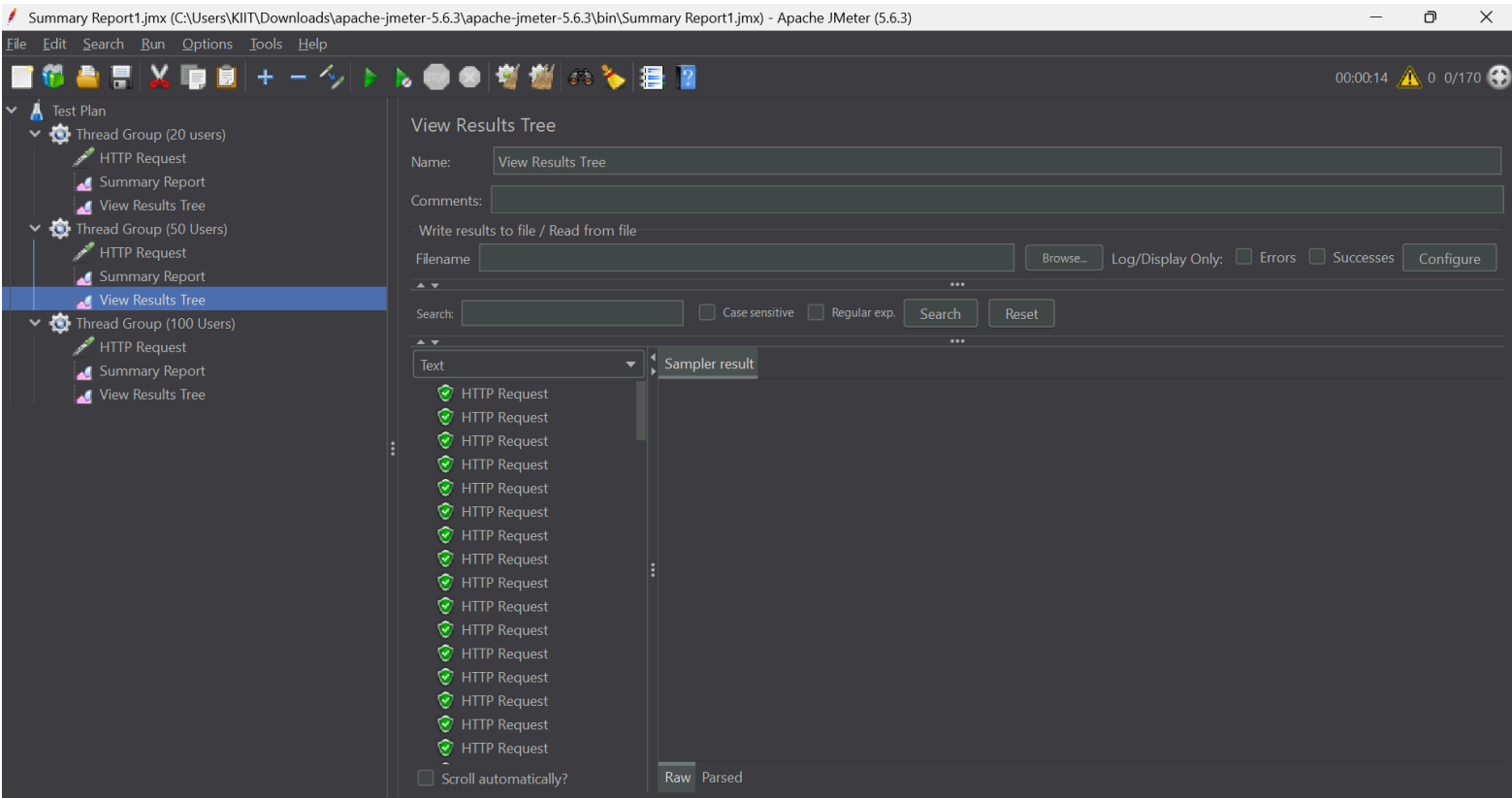
Comments:

Write results to file / Read from file

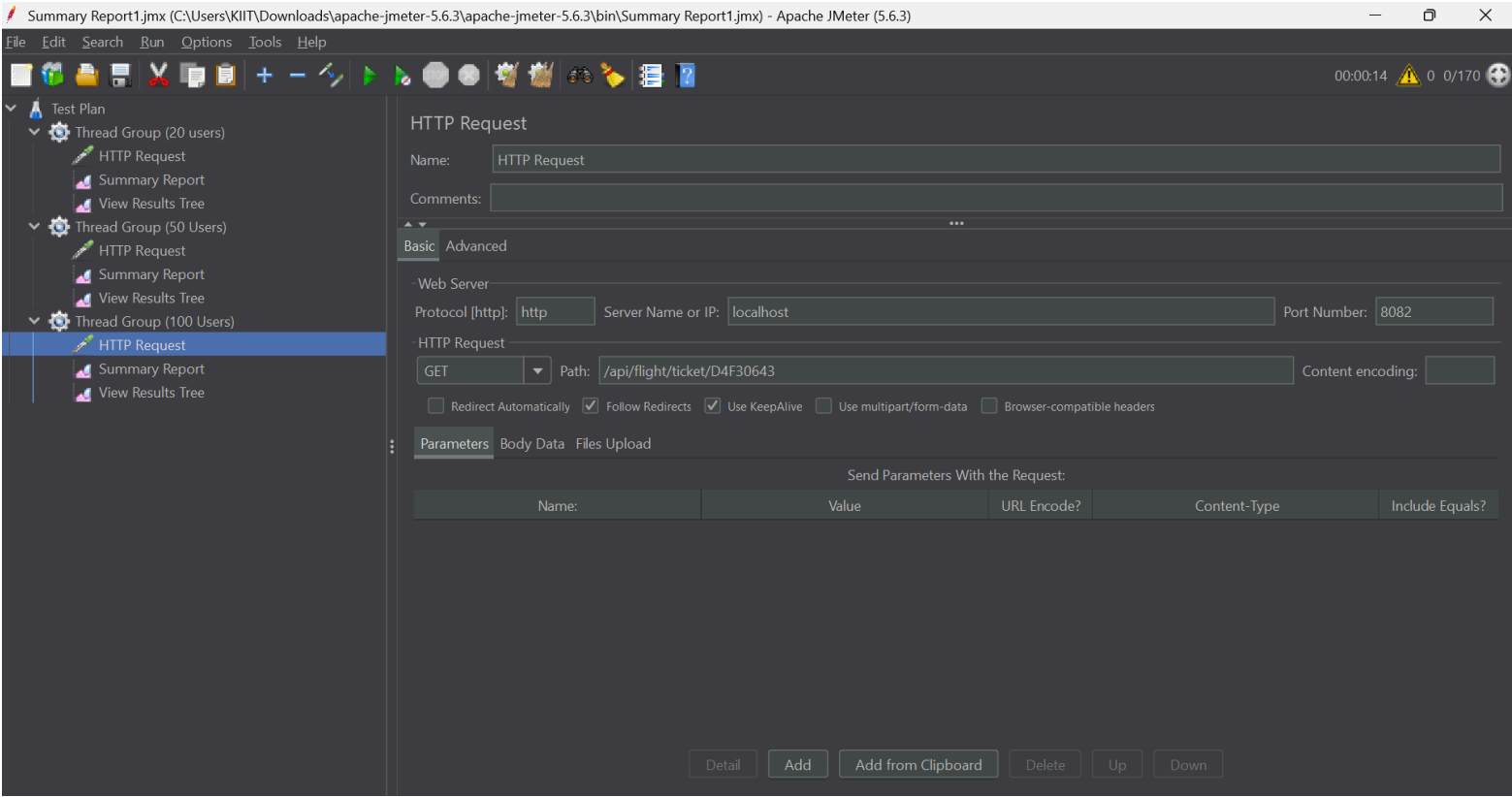
Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
HTTP Request	100	5	3	13	1.73	0.00%	4.6/min	0.03	0.01	389.0
TOTAL	100	5	3	13	1.73	0.00%	4.6/min	0.03	0.01	389.0

☐ Include group name in label? ☒ Save Table Header



3. 100 USERS



[illegible]

The screenshot displays the Apache JMeter 5.6.3 application window. The title bar reads "Summary Report1.jmx (C:\Users\KIIT\Downloads\apache-jmeter-5.6.3\bin\Summary Report1.jmx) - Apache JMeter (5.6.3)". The interface is divided into three main sections:

- Left Panel (Test Plan Tree):** Shows a hierarchical structure of the test plan. It includes a "Test Plan" root, followed by three "Thread Group" entries: "Thread Group (20 users)", "Thread Group (50 Users)", and "Thread Group (100 Users)". Each thread group contains an "HTTP Request" and a "Summary Report". The "View Results Tree" element is highlighted under the "Thread Group (100 Users)".
- Top Panel (View Results Tree Configuration):** Contains settings for the selected "View Results Tree" element. The "Name" field is set to "View Results Tree". The "Comments" field is empty. The "Write results to file / Read from file" section includes a "Filename" field, a "Browse..." button, and checkboxes for "Log/Display Only", "Errors", and "Successes". A "Configure" button is also present. The "Search" section includes a search input field, checkboxes for "Case sensitive" and "Regular exp.", and "Search" and "Reset" buttons.
- Bottom Panel (View Results Tree Data):** Displays the results of the test. The "Text" tab is selected, showing a list of "Sampler result" entries. Each entry is preceded by a green checkmark icon and the text "HTTP Request". The list is scrollable, and a "Scroll automatically?" checkbox is located at the bottom left. The "Raw" and "Parsed" tabs are visible at the bottom right.