

IMPORTANT SCREENSHOTS:

Working eureka:

The screenshot shows the Spring Eureka interface at localhost:8761. It displays the following sections:

- System Status:** Shows environment (test), data center (default), current time (2025-11-30T04:32:26 +0530), uptime (00:02), lease expiration enabled (false), renew threshold (6), and renew count (2).
- DS Replicas:** Shows instances registered with Eureka across three applications: API-GATEWAY, BOOKING-SERVICE, and FLIGHT-SERVICE. Each instance is UP and assigned a specific IP and port.
- General Info:** Displays various system metrics such as total available memory (88mb), number of CPUs (8), current memory usage (58mb or 65%), and server up time.

MongoDB:(2 databases)

booking_db	20.48 kB	1	1
config	12.29 kB	0	2
flightdb	20.48 kB	1	1

FlightDB:

The screenshot shows the Compass MongoDB UI at localhost:27017. It displays the following interface:

- Connections:** Shows a connection to localhost:27017 with databases: admin, config, flight-db, flights, flight_booking, and flightdb.
- Documents:** Shows a single document in the flights collection with the following fields:

```
id: ObjectId("692b8227d1a16a2c6ca0c238")
originCity: "Mumbai"
destinationCity: "Delhi"
date: "2025-12-19"
totalSeats: 100
availableSeats: 100
_class: "com.flightservice.model.Flight"
```

BookingDB:

```

{
  "_id": "ObjectID('692b8227d1a16a2c6ca9c238')",
  "tripType": "ONE_WAY",
  "flightIdToWard": "692b8227d1a16a2c6ca9c238",
  "seatNumbersToWard": [
    {
      "0": "12A",
      "1": "12B"
    }
  ],
  "passengers": [
    {
      "0": {
        "name": "Ravi",
        "age": 28,
        "gender": "Male"
      },
      "1": {
        "name": "Sita",
        "age": 25,
        "gender": "Female"
      }
    }
  ],
  "meal": false,
  "bookingDateTime": "2025-11-29T23:33:34.617+00:00",
  "status": "BOOKED",
  "_class": "com.bookingservice.model.Booking"
}

```

Postman:

Add Inventory:

```

{
  "fromCity": "HYD",
  "toCity": "DEL",
  "date": "2025-12-10",
  "totalSeats": 100,
  "availableSeats": 100,
  "tripType": "ONE_WAY"
}

```

```

{
  "id": "692b8227d1a16a2c6ca9c238",
  "fromCity": "HYD",
  "toCity": "DEL",
  "date": "2025-12-10",
  "totalSeats": 100,
  "availableSeats": 100
}

```

Book Flight:

POST add [CONFLIK] POST Nev POST Nev POST Nev POST Nev POST add POST boo GET New > + v No environment

POST http://localhost:8080/booking/book

Body: raw

```

1 {
2   "tripType": "ONE WAY",
3   "flightIdOnward": "692b8227d1a16a2c6ca9c238",
4   "seatNumbersOnward": ["12A", "12B"],
5
6   "userName": "Ravi Kumar",
7   "email": "ravi@gmail.com",
8
9   "passengers": [
10     {"name": "Ravi", "age": 28, "gender": "Male"},
11     {"name": "Sita", "age": 25, "gender": "Female"}
12   ]
13 }

```

201 Created 61 ms 580 B Save Response

Get details by PNR:

POST add [CONFLIK] POST Nev POST Nev POST Nev POST Nev POST add POST boo GET get b > + v No environment

GET http://localhost:8082/booking/0ABAD8AE

Query Params

Key	Value	Description
Key	Value	Description

Body: raw

```

1 {
2   "id": "692b88a816c9d6448d85663d",
3   "pnr": "0ABAD8AE",
4   "tripType": "ONE WAY",
5   "flightIdOnward": "692b8227d1a16a2c6ca9c238",
6   "flightIdReturn": null,
7   "seatNumbersOnward": [
8     "12A",
9     "12B"
10   ],
11   "seatNumbersReturn": null,
12   "userName": "Ravi Kumar",
13   "email": "ravi@gmail.com",

```

200 OK 69 ms 620 B Save Response

Get details by EmailID:

The screenshot shows the Postman interface with the following details:

- Collection:** Joshika sree Polaku's Workspace
- Request Type:** GET
- URL:** http://localhost:8082/booking/history/ravi@gmail.com
- Params:** Key Value Description
- Body:** JSON (Preview shows a PNR document with ID, PNR, tripType, flightIdOnward, flightIdReturn, seatNumbersOnward, seatNumbersReturn, and userName).
- Response:** 200 OK | 39 ms | 666 B | Save Response

Cancel Booking:

The screenshot shows the Postman interface with the following details:

- Collection:** Joshika sree Polaku's Workspace
- Request Type:** DELETE
- URL:** http://localhost:8082/booking/cancel/0ABAD8AE
- Params:** Key Value Description
- Body:** Raw (Preview shows a single digit '1').
- Response:** 200 OK | 11 ms | 123 B | Save Response

Spring Implementation:

```

Console X
config-server - ConfigServerApplication [Spring Boot App] /Applications/SpringToolsForEclipse.app/Contents/Eclipse/plugins/org.eclipse.jst.java.core/openjdk.hotspot.jre.full.macosx.aarch64_21.0.9.v20251105-0741/jre/bin/java (30-Nov-2025, 6:51:11 am el)

:: Spring Boot :: (v3.2.2)

2025-11-30T05:51:12.332+05:30 INFO 28643 --- [config-server] [main] c.configserver.ConfigServerApplication : Starting ConfigServerApplication using Java 21.0.9 with P
2025-11-30T05:51:12.333+05:30 INFO 28643 --- [config-server] [main] c.configserver.ConfigServerApplication : No active profile set, falling back to 1 default profile:
2025-11-30T05:51:12.334+05:30 INFO 28643 --- [config-server] [main] o.s.cloud.context.ScenarioName : BeanFactory [172.17.0.12-6937-3282-b2d4-c8cce22af875]
2025-11-30T05:51:12.644+05:30 INFO 28643 --- [config-server] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8888 (http)
2025-11-30T05:51:12.649+05:30 INFO 28643 --- [config-server] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-11-30T05:51:12.649+05:30 INFO 28643 --- [config-server] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.18]
2025-11-30T05:51:12.668+05:30 INFO 28643 --- [config-server] [main] o.a.c.c.C.[Tomcat].localhost:1/ : Initializing Spring embedded WebApplicationContext
2025-11-30T05:51:12.668+05:30 INFO 28643 --- [config-server] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3
2025-11-30T05:51:14.154+05:30 INFO 28643 --- [config-server] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8888 (http) with context path ''
2025-11-30T05:51:14.162+05:30 INFO 28643 --- [config-server] [main] c.configserver.ConfigServerApplication : Started ConfigServerApplication in 1.941 seconds (process

```

Localhosts:

```

localhost:8888/flight-service/default
Pretty print
{
  "name": "flight-service",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "c849de042cc5e91cb0286187b89166d9ed840765",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/PolakuJoshiKaSree/FlightBooking-Config-Repo.git/flight-service.properties",
      "source": {
        "server.port": "8081",
        "spring.data.mongodb.uri": "mongodb://localhost:27017/flightdb"
      }
    }
  ]
}

```

```

localhost:8888/booking-service/default
Pretty print
{
  "name": "booking-service",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "c849de042cc5e91cb0286187b89166d9ed840765",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/PolakuJoshiKaSree/FlightBooking-Config-Repo.git/booking-service.properties",
      "source": {
        "server.port": "8082",
        "spring.data.mongodb.uri": "mongodb://localhost:27017/bookingdb"
      }
    }
  ]
}

```

localhost:8888/api-gateway/default

Pretty print

```
{
  "name": "api-gateway",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "c849de042cc5e91cb0286187b89166d9ed840765",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/PolakuJoshiKaSree/FlightBooking-Config-Repo.git/api-gateway.properties",
      "source": {
        "server.port": "8080",
        "spring.cloud.gateway.discovery.locator.enabled": "true"
      }
    }
  ]
}
```

Kafka :

```
polakujoshikasree@POLAKUs-Laptop FlightBookingApplication % kafka-topics --bootstrap-server localhost:9092 --list
booking-cancelled
booking-created
bookings-topic
cancellations-topic
polakujoshikasree@POLAKUs-Laptop FlightBookingApplication %
```

Running kafka :

The screenshot shows the Postman interface with a collection named 'Joshika sree Polaku's Workspace'. A new collection named 'kafka request sent' is being created. A POST request is being prepared to the URL `http://localhost:8082/booking/book`. The request body is set to raw JSON and contains the following data:

```

1  {
2    "tripType": "ONE WAY",
3    "flightIdOnward": "692b8227d1a16a2c6ca9c238",
4    "seatNumbersOnward": ["12A", "12B"],
5    "userName": "Ravi",
6    "email": "ravi@gmail.com",
7    "passengers": [
8      {"name": "Ravi", "age": 28, "gender": "Male"},
9      {"name": "Sita", "age": 25, "gender": "Female"}
10   ]
11 }
12

```

The response status is 201 Created, and the response body is displayed below:

```

1  {
2    "id": "692b0973f3342401641d1eccf",
3    "pnr": "14EE5F64",
4    "tripType": "ONE WAY",
5    "flightIdOnward": "692b8227d1a16a2c6ca9c238",
6    "flightIdReturn": null,
7    "seatNumbersOnward": [
8      "12A",
9      "12B"
10    ],
11    "seatNumbersReturn": null,
12    "userName": "Ravi",
13    "email": "ravi@gmail.com",

```

Kafka Notification:

```
polakujoshikasree@POLAKUs-Laptop FlightBookingApplication % brew services list
Name      Status  User          File
kafka     started polakujoshikasree ~/Library/LaunchAgents/homebrew.mxcl.kafka.plist
mongodb-community started polakujoshikasree ~/Library/LaunchAgents/homebrew.mxcl.mongodb-community.plist
zookeeper started polakujoshikasree ~/Library/LaunchAgents/homebrew.mxcl.zookeeper.plist
polakujoshikasree@POLAKUs-Laptop FlightBookingApplication % kafka-console-consumer --bootstrap-server localhost:9092 --topic booking-events --from-beginning
Booking Successful - PNR: 38A6510D
```

JMeter:

Post request (add inventory):

Test Plan

- Thread Group
 - HTTP Request
 - HTTP Header Manager
- Summary Report
- View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Request Body: Request Headers

```
1 POST http://localhost:8080/flights/add
2
3 POST data:
4 {
5   "fromCity": "Mumbai",
6   "toCity": "Delhi",
7   "date": "2023-12-10",
8   "totalSeats": 100,
9   "availableSeats": 100,
10  "tripType": "one-way"
11 }
12
13
14 [no cookies]
15
```

00:00:00 0/0 2/0

Test Plan

- Thread Group
 - HTTP Request
 - HTTP Header Manager
- Summary Report
- View Results Tree

Name: Summary Report

Comments:

Write results to file / Read from file

Request Body: Request Headers

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	13	11	18	1.97	0.00%	20.7/sec	5.16	6.61	255.0
TOTAL	20	13	11	18	1.97	0.00%	20.7/sec	5.16	6.61	255.0

00:00:00 0/0 2/0

Add Booking :

Test Plan

- Thread Group
 - HTTP Request
 - HTTP Header Manager
- Summary Report
- View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Request Body: Request Headers

```
1 POST http://localhost:8080/booking/book
2
3 POST data:
4 {
5   "tripType": "ONE WAY",
6   "flightIdOrWavetext": "09398227d1a162dc49c230",
7   "seatHoldersOrWavetext": ["12A", "12B"],
8   "userName": "Ravi Kumar",
9   "email": "ravipawan@gmail.com",
10  "passenger": [
11    {"name": "Ravi", "age": 28, "gender": "Male"},
12    {"name": "Sita", "age": 25, "gender": "Female"}
13  ]
14
15 [no cookies]
16
```

00:00:01 0/0 2/0

Test Plan

- Thread Group
 - HTTP Request
 - HTTP Header Manager
- Summary Report
- View Results Tree

Name: Summary Report

Comments:

Write results to file / Read from file

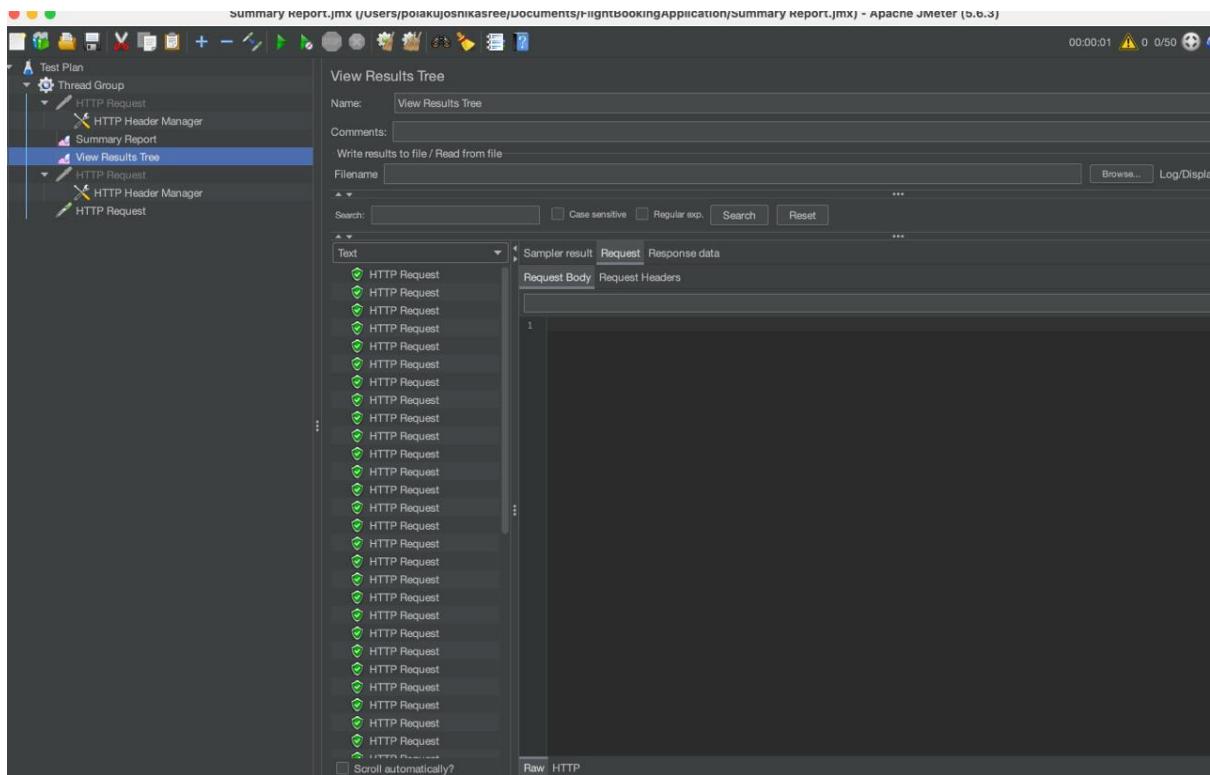
Request Body: Request Headers

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	13	11	18	1.97	0.00%	20.7/sec	5.16	6.61	255.0
TOTAL	20	13	11	18	1.97	0.00%	20.7/sec	5.16	6.61	255.0

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	10	5	16	3.23	100.00%	20.7/sec	6.21	10.03	307.0
TOTAL	20	10	5	16	3.23	100.00%	20.7/sec	6.21	10.03	307.0

Get PNR:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	50	6	4	11	1.54	0.00%	50.5/sec	31.04	8.50	630.0
TOTAL	50	6	4	11	1.54	0.00%	50.5/sec	31.04	8.50	630.0



SonarAnalysis:

A screenshot of the SonarQube web interface. The top navigation bar includes 'My Projects', 'My Issues', 'Explore', and an 'Upgrade' button. The main content area shows the 'Main Branch Summary' for the project 'BookingServiceApplication' in the 'master' branch. Key statistics displayed include '1x Lines of Code', 'Version 1.0.0', and 'Last analysis 8 minutes ago'. A prominent 'Passed' status is shown with a green checkmark icon. Below this, four main metrics are presented in a grid: 'New Issues' (8, 0 conditions set), 'Accepted Issues' (0, 0 valid issues), 'Coverage' (93.42%, required 40.0%), and 'Duplications' (0.0%, required 3.0%). The 'Security Hotspots' section shows 0 issues with 0 conditions set.