FLIGHT BOOKING SYSTEM

(Spring Boot WebFlux + Reactive MongoDB)

1. Introduction

This document provides a complete overview of the Flight Booking System implemented using:

- Spring Boot WebFlux (Reactive, non-blocking)
- Reactive MongoDB
- Reactive Streams (Mono, Flux)
- JaCoCo for code coverage
- SonarQube for code quality
- JMeter for load testing

The system supports:

User-side features:

- Search flights
- Book tickets with passenger details & seat selection
- Retrieve bookings via PNR
- View booking history using email or user ID
- Cancel bookings (with 24-hour rule)

Admin-side features:

- Create and manage airlines
- Create and manage flights
- Automatic seat availability updates
- Retrieve flights by airline

This version is fully reactive, non-blocking, and optimized for high concurrency.

2. Features Overview

2.1 User Features

- Search flights using:
  - source airport
  - destination airport
  - journey date
- Book flight tickets with:
  - passenger list
  - seat selection
  - meal type
- Retrieve ticket via PNR (unique booking reference)
- Cancel booking
- View booking history:
  - by email
  - by user ID

2.2 Admin Features

- Add airline

- Add flight
- Modify airline
- Delete airline
- Get all flights for a specific airline

2.3 System Features
- Non-blocking, fully reactive
- Business validations
- Seat conflict detection
- Consistent seat availability management
- Global exception handling
- Auto-generated PNR codes

3. Architecture
The system follows a modular, clean, reactive architecture:

Client → Controller → Service → Repository → MongoDB

Layers Explained

- Controller Layer

  Handles HTTP requests using Spring WebFlux.

- Service Layer

  Contains all business logic:
  - booking validation
  - seat conflict detection
  - cancellation rules

  Uses Mono and Flux.

- Repository Layer

  Uses ReactiveMongoRepository for:
  - asynchronous queries
  - reactive CRUD operations

- Entity Layer

  Defines MongoDB document models using:
  @Document, @Id, and validation annotations.

- Global Exception Handler

  Handles:
  - Validation errors
  - Not found
  - Seat conflicts
  - Business rule violations

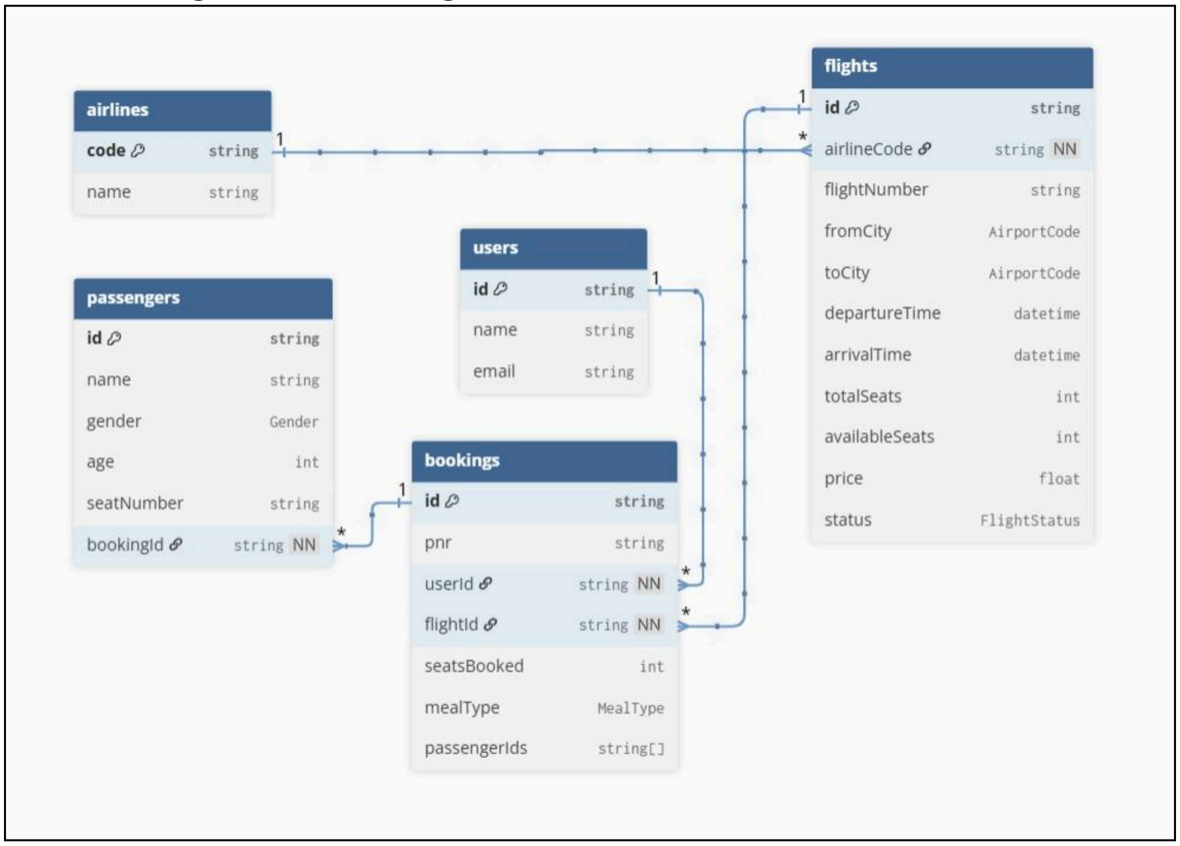
- Testing Layer
  - WebFluxTest (controller)

- Reactor Test + StepVerifier (service)
- Mockito (mocking repositories)

4. Technology Stack

Component	Technology
Backend	Spring Boot 3 + WebFlux
Database	MongoDB (Reactive)
ORM	Spring Data Reactive MongoDB
Testing	JUnit5, Mockito
Build Tool	Maven
Code Coverage	JaCoCo
Code Quality	SonarQube
Performance Testing	Apache JMeter
Java Version	Java 17

5. Database Design - Reactive MongoDB



Validation Rules Summary:

Airline Validations:
- Code required
- Name required

User Validations
- Name required
- Valid email format

Flight Validations
- airlineCode: required, must exist
- fromCity != toCity
- departure < arrival
- departure > now
- totalSeats >= 1
- price >= 0

Passenger Validations
- name required
- age = 1-120
- seatNumber = ^[A-Z][0-9]+$
- No duplicate seat numbers in request
- No seat conflict with existing bookings

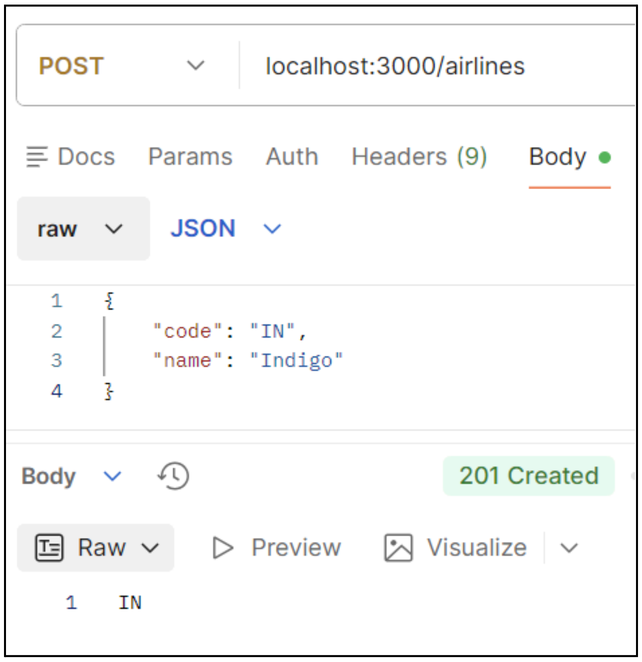
Booking Validations
- user exists
- flight exists
- seatsBooked = passengers.length
- enough available seats
- booking prevented if conflict
- cancellation not allowed < 24 hours
- unique PNR per booking
- Reduce flight seats on success

5. API Documentation (with sample input + positive outputs)
(this document only contains some of the api cases)

A dedicated document containing all APIs for every entity (Airline, User, Flight, Booking, Passenger) with complete positive and negative request/response examples is available here:
https://docs.google.com/document/d/1FgRpyqi2NfTODbka28p0oVAvJWYLtPqGo84eZlSM8KY/edit?usp=sharing

5.1 Airline APIs

➢ Create Airline : POST /airlines



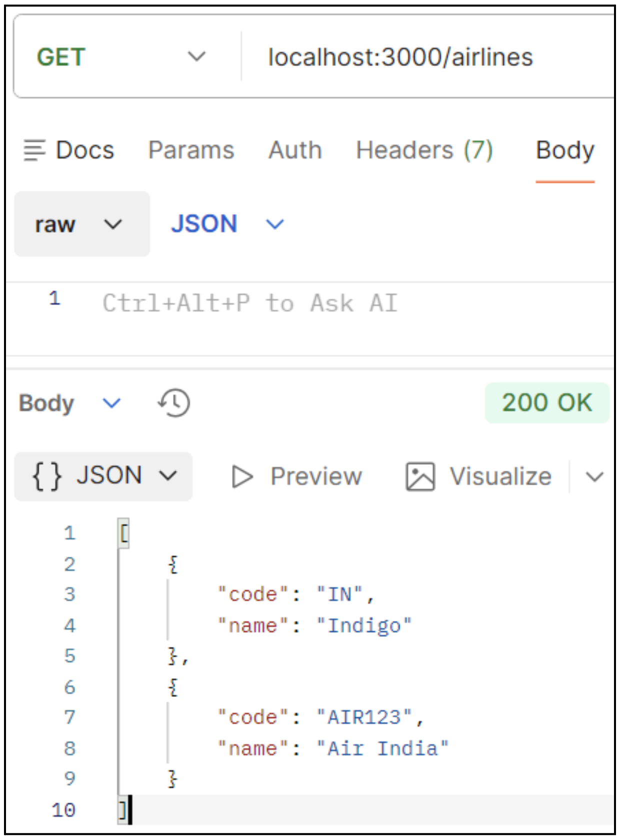
localhost:3000/airlines

```
{
    "code": "IN",
    "name": "Indigo"
}
```

201 Created

```
IN
```

➢ Get All Airlines : GET /airlines

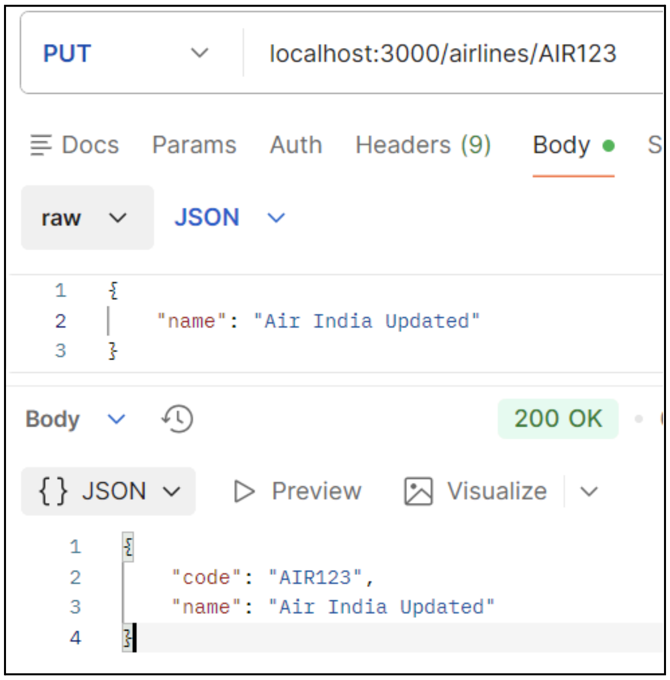


localhost:3000/airlines

200 OK

```
[
    {
        "code": "IN",
        "name": "Indigo"
    },
    {
        "code": "AIR123",
        "name": "Air India"
    }
]
```

➢ Update Airline : PUT /airlines/{code}



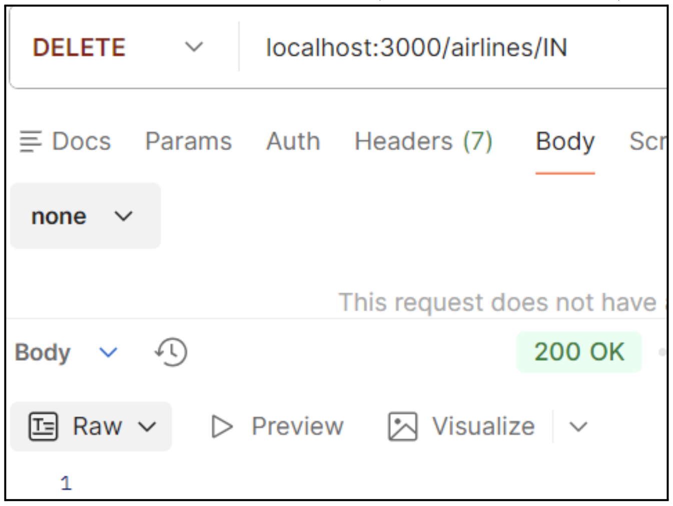
PUT localhost:3000/airlines/AIR123

```
{
    "name": "Air India Updated"
}
```

200 OK

```
{
    "code": "AIR123",
    "name": "Air India Updated"
}
```

➢ Delete Airline : DELETE /airlines/{code} → 200 OK
If not found → 404 Not Found (Airline Not Found)



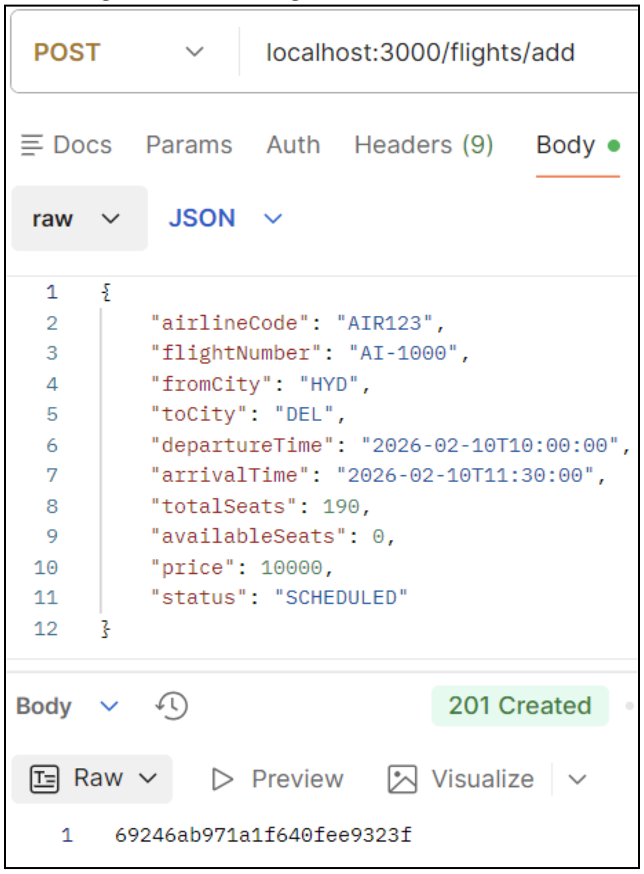
DELETE localhost:3000/airlines/IN

This request does not have

200 OK

5.2 Flight APIs

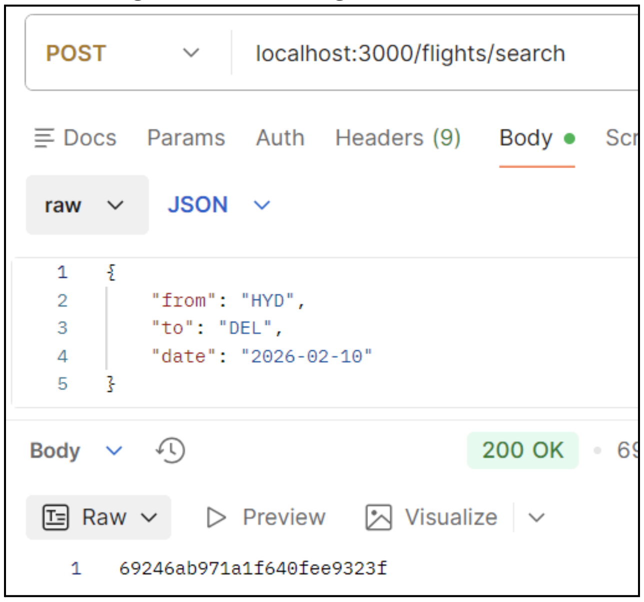
➢ Add Flight : POST /flights/add



```
{
    "airlineCode": "AIR123",
    "flightNumber": "AI-1000",
    "fromCity": "HYD",
    "toCity": "DEL",
    "departureTime": "2026-02-10T10:00:00",
    "arrivalTime": "2026-02-10T11:30:00",
    "totalSeats": 190,
    "availableSeats": 0,
    "price": 10000,
    "status": "SCHEDULED"
}
```

```
69246ab971a1f640fee9323f
```

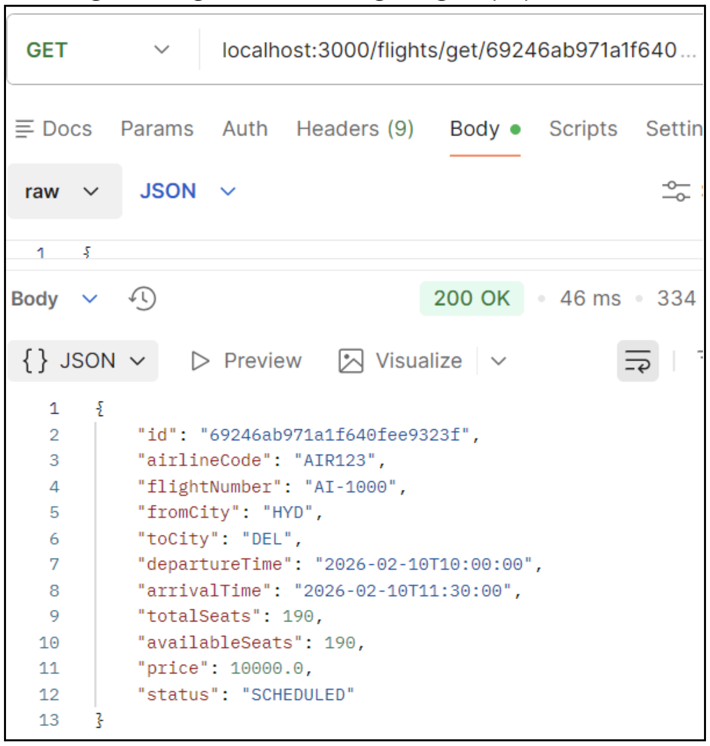
➢ Search Flights : POST /flights/search



```
{
    "from": "HYD",
    "to": "DEL",
    "date": "2026-02-10"
}
```

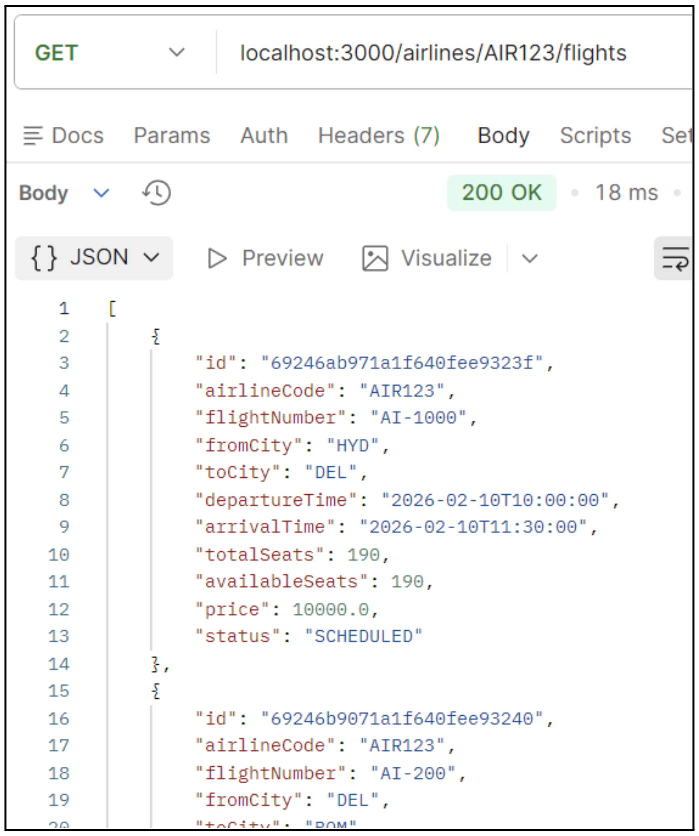
```
69246ab971a1f640fee9323f
```

➢ Get Flight using id : GET /flights/get/{id}


GET
localhost:3000/flights/get/69246ab971a1f640...
Docs Params Auth Headers (9) Body Scripts Settin
raw JSON
Body 200 OK 46 ms 334
{} JSON Preview Visualize

```
{
    "id": "69246ab971a1f640fee9323f",
    "airlineCode": "AIR123",
    "flightNumber": "AI-1000",
    "fromCity": "HYD",
    "toCity": "DEL",
    "departureTime": "2026-02-10T10:00:00",
    "arrivalTime": "2026-02-10T11:30:00",
    "totalSeats": 190,
    "availableSeats": 190,
    "price": 10000.0,
    "status": "SCHEDULED"
}
```

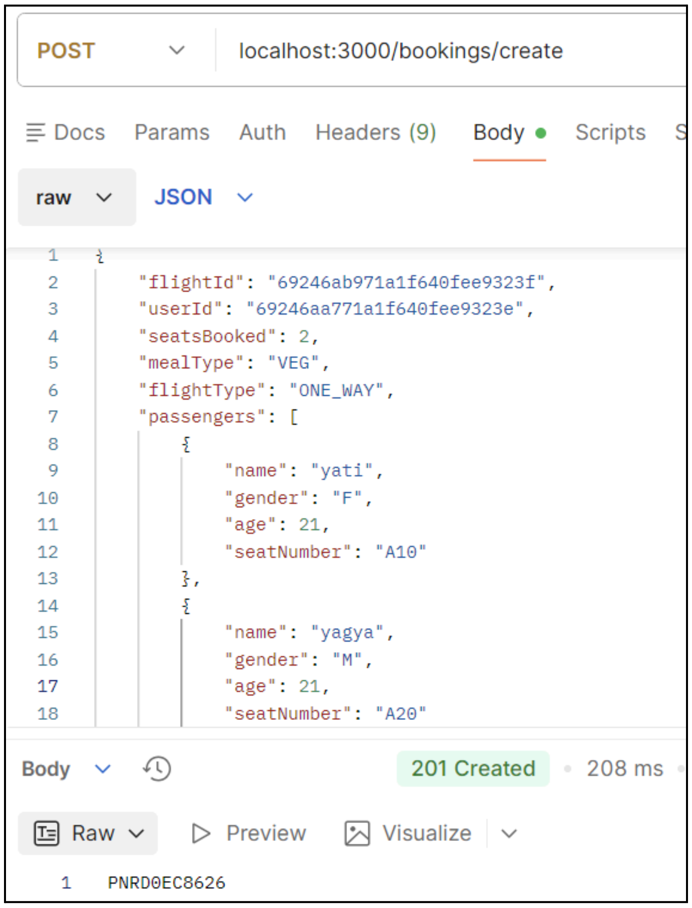
➢ Get Flights by Airline : GET /airlines/{code}/flights


GET
localhost:3000/airlines/AIR123/flights
Docs Params Auth Headers (7) Body Scripts Se
Body 200 OK 18 ms
{} JSON Preview Visualize

```
[
    {
        "id": "69246ab971a1f640fee9323f",
        "airlineCode": "AIR123",
        "flightNumber": "AI-1000",
        "fromCity": "HYD",
        "toCity": "DEL",
        "departureTime": "2026-02-10T10:00:00",
        "arrivalTime": "2026-02-10T11:30:00",
        "totalSeats": 190,
        "availableSeats": 190,
        "price": 10000.0,
        "status": "SCHEDULED"
    },
    {
        "id": "69246b9071a1f640fee93240",
        "airlineCode": "AIR123",
        "flightNumber": "AI-200",
        "fromCity": "DEL",
        "toCity": "BOM",
```

5.3 Booking APIs

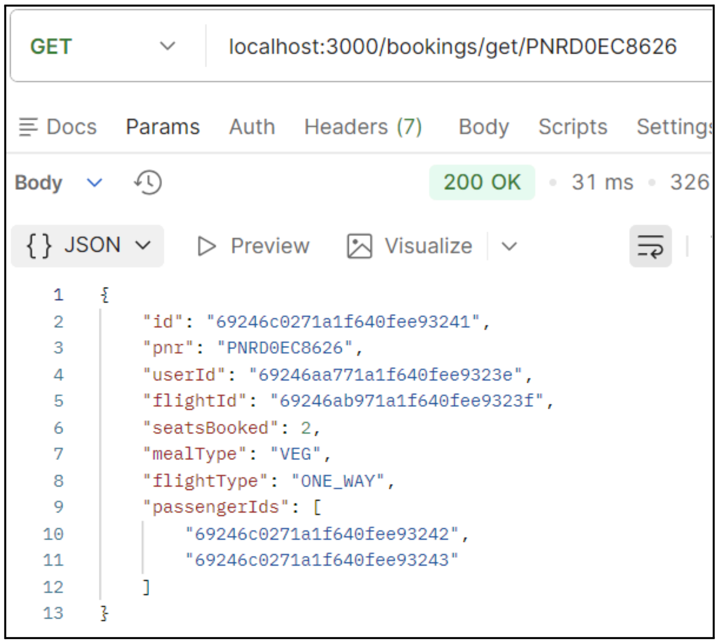
➢ Create Booking : POST /bookings/create



```
{
    "flightId": "69246ab971a1f640fee9323f",
    "userId": "69246aa771a1f640fee9323e",
    "seatsBooked": 2,
    "mealType": "VEG",
    "flightType": "ONE_WAY",
    "passengers": [
        {
            "name": "yati",
            "gender": "F",
            "age": 21,
            "seatNumber": "A10"
        },
        {
            "name": "yagya",
            "gender": "M",
            "age": 21,
            "seatNumber": "A20"
```

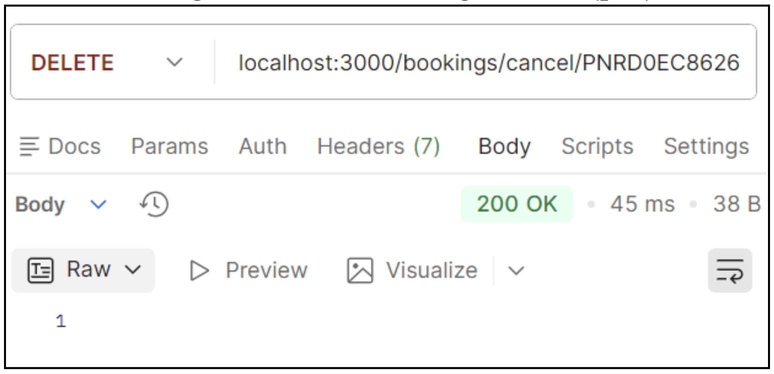
```
PNRD0EC8626
```

➢ Get Booking : GET /bookings/get/{pnr}



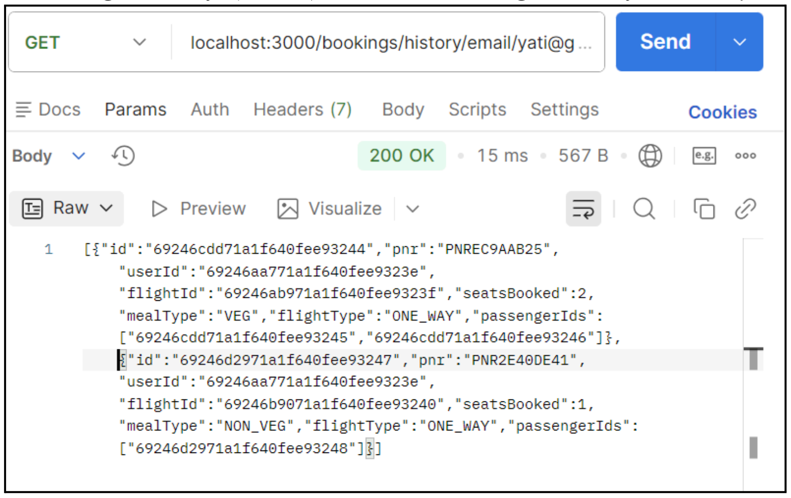
```
{
    "id": "69246c0271a1f640fee93241",
    "pnr": "PNRD0EC8626",
    "userId": "69246aa771a1f640fee9323e",
    "flightId": "69246ab971a1f640fee9323f",
    "seatsBooked": 2,
    "mealType": "VEG",
    "flightType": "ONE_WAY",
    "passengerIds": [
        "69246c0271a1f640fee93242",
        "69246c0271a1f640fee93243"
    ]
}
```

➢ Cancel Booking : DELETE /bookings/cancel/{pnr}



DELETE
localhost:3000/bookings/cancel/PNRD0EC8626
Docs Params Auth Headers (7) Body Scripts Settings
Body
200 OK · 45 ms · 38 B
Raw Preview Visualize

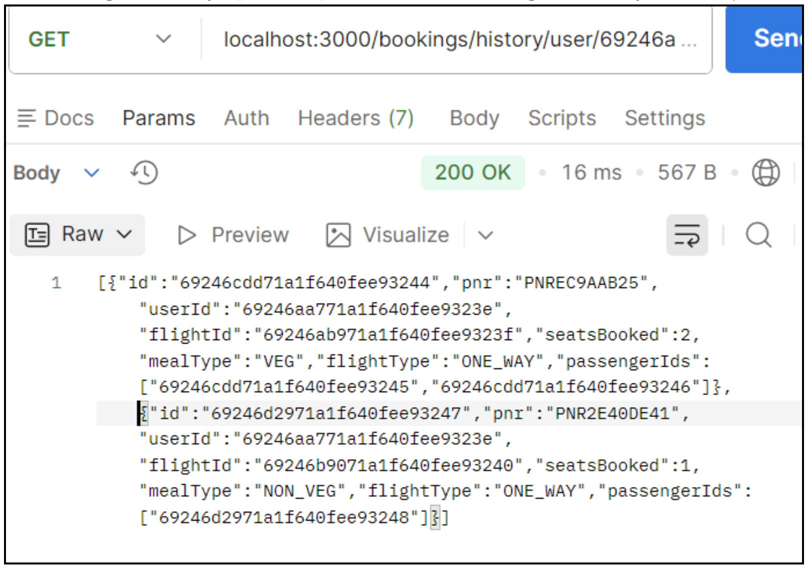
➢ Booking History (email) : GET /bookings/history/email/{email}



GET
localhost:3000/bookings/history/email/yati@g ...
Send
Docs Params Auth Headers (7) Body Scripts Settings
Cookies
Body
200 OK · 15 ms · 567 B
Raw Preview Visualize

```
[{"id":"69246cdd71a1f640fee93244","pnr":"PNREC9AAB25",
  "userId":"69246aa771a1f640fee9323e",
  "flightId":"69246ab971a1f640fee9323f","seatsBooked":2,
  "mealType":"VEG","flightType":"ONE_WAY","passengerIds":
  ["69246cdd71a1f640fee93245","69246cdd71a1f640fee93246"]},
  {"id":"69246d2971a1f640fee93247","pnr":"PNR2E40DE41",
  "userId":"69246aa771a1f640fee9323e",
  "flightId":"69246b9071a1f640fee93240","seatsBooked":1,
  "mealType":"NON_VEG","flightType":"ONE_WAY","passengerIds":
  ["69246d2971a1f640fee93248"]}]
```

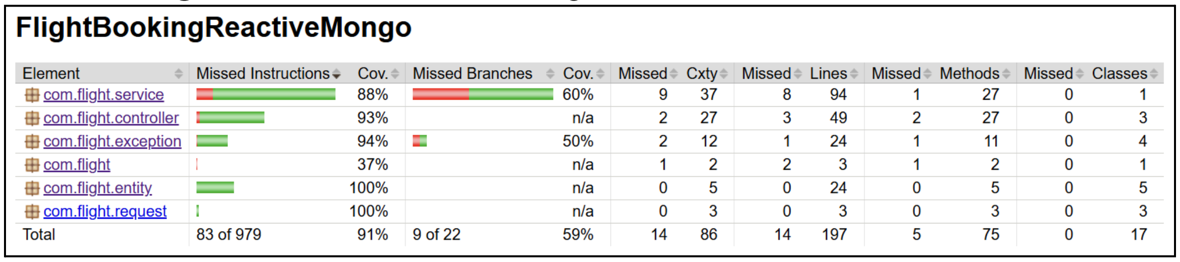
➢ Booking History (userId) : GET /bookings/history/user/{userId}



GET
localhost:3000/bookings/history/user/69246a ...
Sen
Docs Params Auth Headers (7) Body Scripts Settings
Body
200 OK · 16 ms · 567 B
Raw Preview Visualize

```
[{"id":"69246cdd71a1f640fee93244","pnr":"PNREC9AAB25",
  "userId":"69246aa771a1f640fee9323e",
  "flightId":"69246ab971a1f640fee9323f","seatsBooked":2,
  "mealType":"VEG","flightType":"ONE_WAY","passengerIds":
  ["69246cdd71a1f640fee93245","69246cdd71a1f640fee93246"]},
  {"id":"69246d2971a1f640fee93247","pnr":"PNR2E40DE41",
  "userId":"69246aa771a1f640fee9323e",
  "flightId":"69246b9071a1f640fee93240","seatsBooked":1,
  "mealType":"NON_VEG","flightType":"ONE_WAY","passengerIds":
  ["69246d2971a1f640fee93248"]}]
```

6. **Code Coverage - JaCoCo** - Overall Coverage: 91%

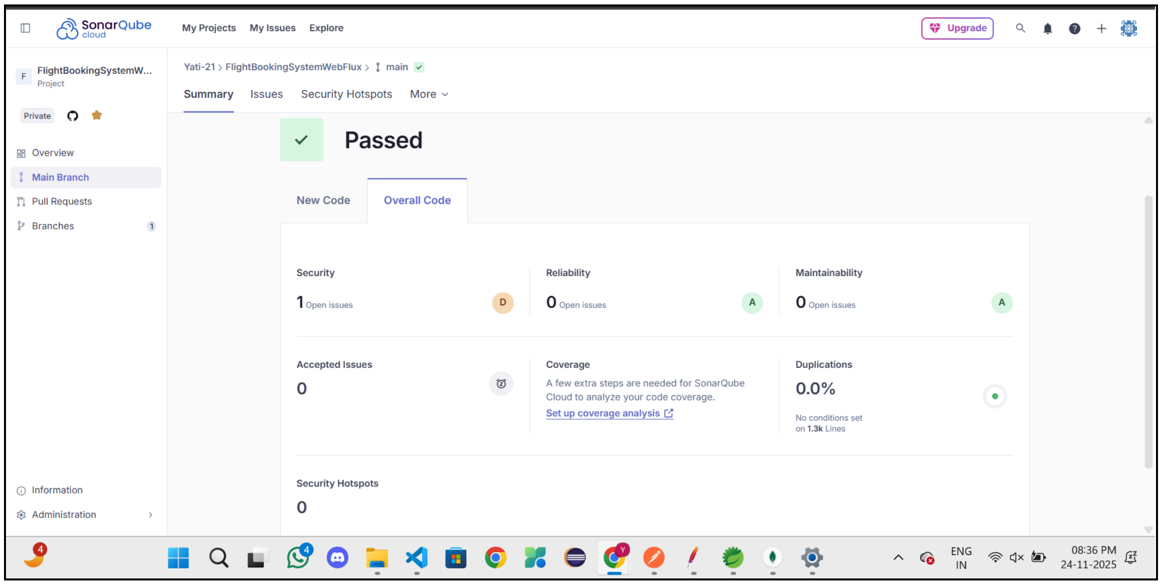


FlightBookingReactiveMongo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.flight.service		88%		60%	9	37	8	94	1	27	0	1
com.flight.controller		93%		n/a	2	27	3	49	2	27	0	3
com.flight.exception		94%		50%	2	12	1	24	1	11	0	4
com.flight		37%		n/a	1	2	2	3	1	2	0	1
com.flight.entity		100%		n/a	0	5	0	24	0	5	0	5
com.flight.request		100%		n/a	0	3	0	3	0	3	0	3
Total	83 of 979	91%	9 of 22	59%	14	86	14	197	5	75	0	17

7. **SonarQube Analysis**

After Fixes :



SonarQube cloud — My Projects · My Issues · Explore — Upgrade

FlightBookingSystemW... Project — Private

Overview · Main Branch · Pull Requests · Branches 1 · Information · Administration

Yati-21 › FlightBookingSystemWebFlux › main

Summary · Issues · Security Hotspots · More

Passed

New Code · Overall Code

Security: 1 Open issues — D

Reliability: 0 Open issues — A

Maintainability: 0 Open issues — A

Accepted Issues: 0

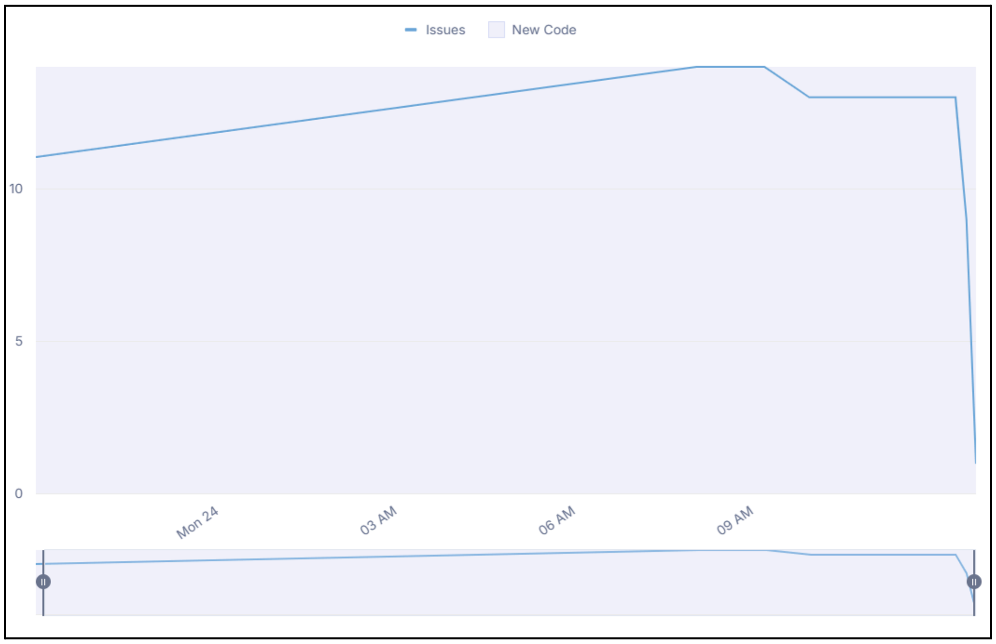
Coverage: A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis

Duplications: 0.0% — No conditions set on 1.3k Lines

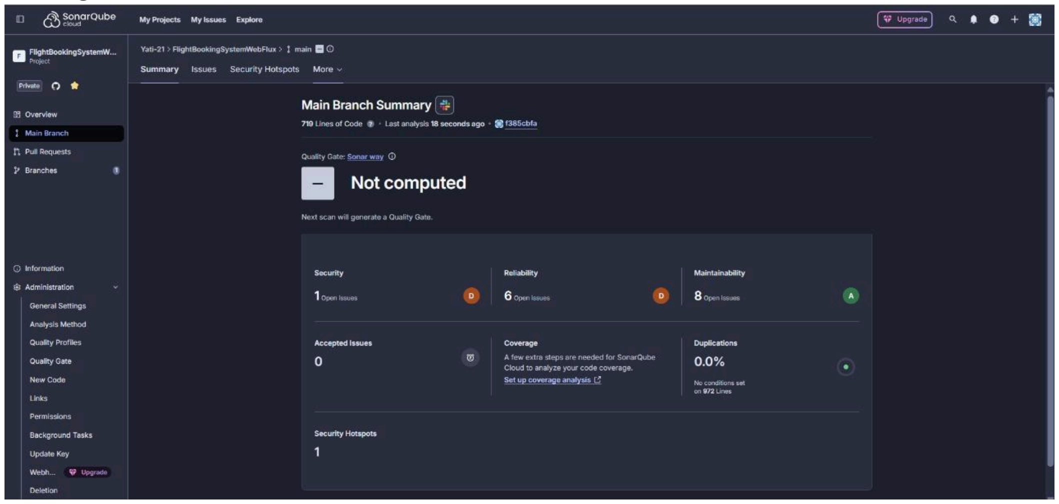
Security Hotspots: 0

ENG IN 08:36 PM 24-11-2025

Issues over time :



Before/During Fixes :



SonarQube cloud · My Projects · My Issues · Explore · Upgrade

FlightBookingSystemW... Project · Private

Overview · Main Branch · Pull Requests · Branches

Information · Administration: General Settings · Analysis Method · Quality Profiles · Quality Gate · New Code · Links · Permissions · Background Tasks · Update Key · Webh... Upgrade · Deletion

Yati-21 > FlightBookingSystemWebFlux > main

Summary · Issues · Security Hotspots · More

Main Branch Summary

719 Lines of Code · Last analysis 18 seconds ago · f385cbfa

Quality Gate: Sonar way

Not computed

Next scan will generate a Quality Gate.

Security: 1 Open issues (D)

Reliability: 6 Open issues (D)

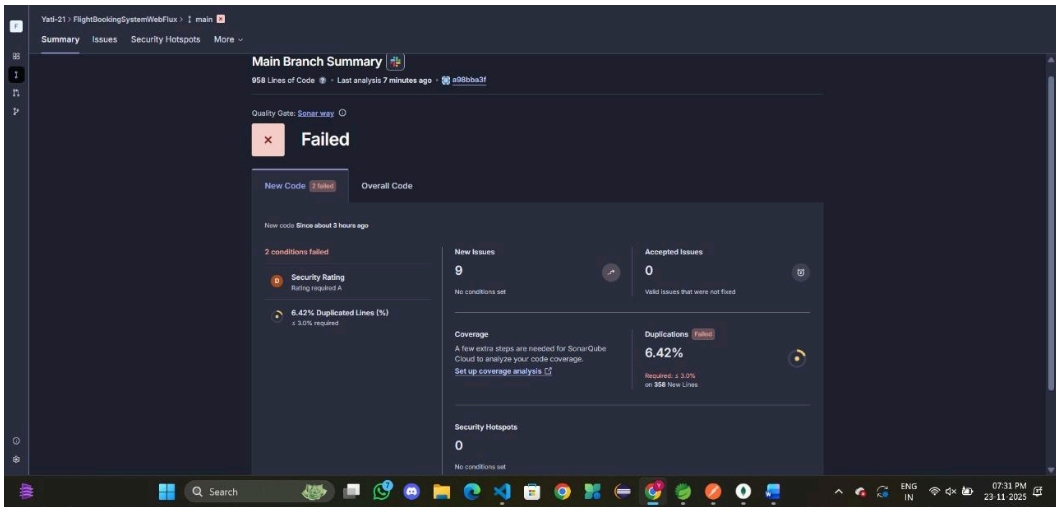
Maintainability: 8 Open issues (A)

Accepted Issues: 0

Coverage: A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis

Duplications: 0.0% — No conditions set on 972 Lines

Security Hotspots: 1



Yati-21 > FlightBookingSystemWebFlux > main

Summary · Issues · Security Hotspots · More

Main Branch Summary

958 Lines of Code · Last analysis 7 minutes ago · a98bba3f

Quality Gate: Sonar way

Failed

New Code (2 failed) · Overall Code

New code Since about 3 hours ago

2 conditions failed

- Security Rating — Rating required A (D)
- 6.42% Duplicated Lines (%) — ≤ 3.0% required

New Issues: 9 — No conditions set

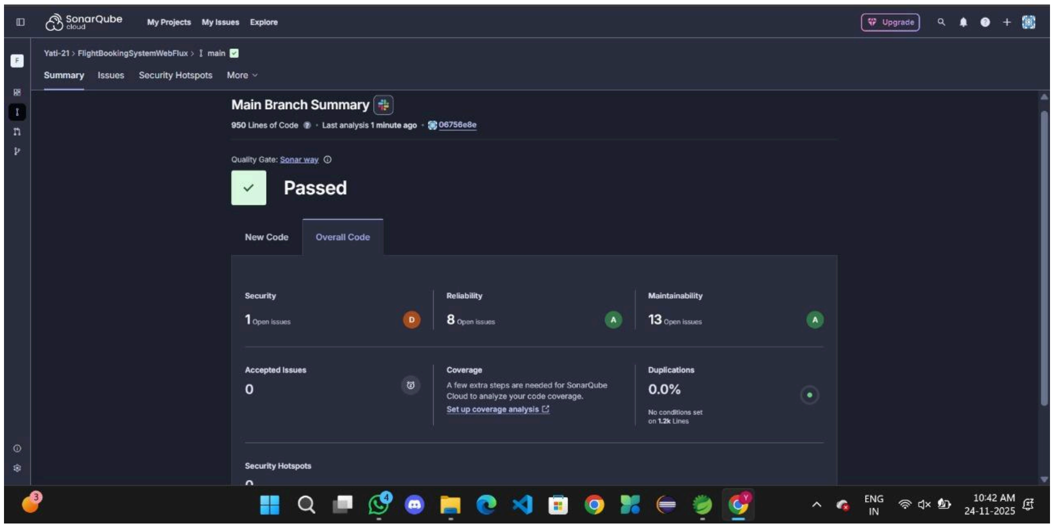
Accepted Issues: 0 — Valid issues that were not fixed

Coverage: A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis

Duplications (Failed): 6.42% — Required: ≤ 3.0% on 358 New Lines

Security Hotspots: 0 — No conditions set

Search · ENG IN · 07:31 PM 23-11-2025



SonarQube cloud · My Projects · My Issues · Explore · Upgrade

Yati-21 > FlightBookingSystemWebFlux > main

Summary · Issues · Security Hotspots · More

Main Branch Summary

950 Lines of Code · Last analysis 1 minute ago · 06756e8e

Quality Gate: Sonar way

Passed

New Code · Overall Code

Security: 1 Open issues (D)

Reliability: 8 Open issues (A)

Maintainability: 13 Open issues (A)

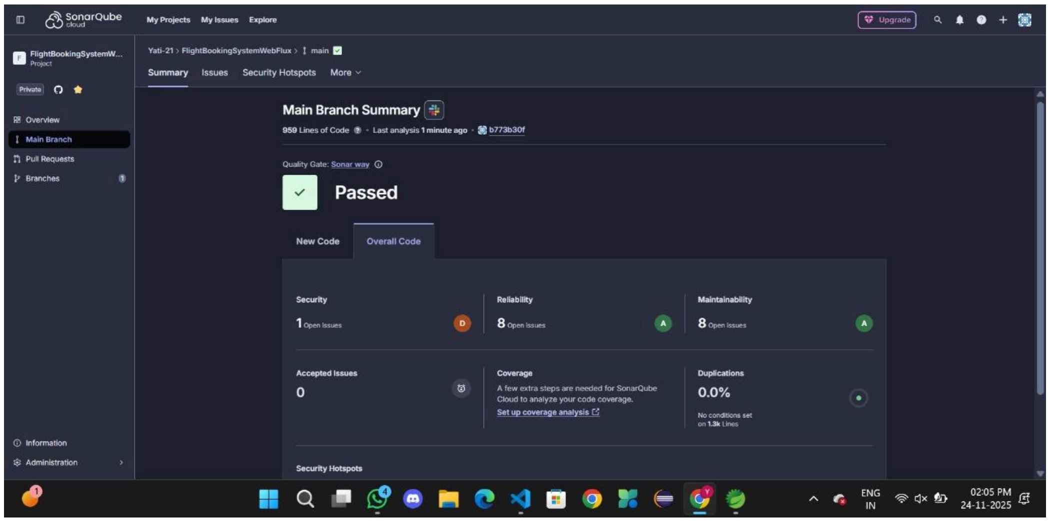
Accepted Issues: 0

Coverage: A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis

Duplications: 0.0% — No conditions set on 1.2k Lines

Security Hotspots: 0

ENG IN · 10:42 AM 24-11-2025



Main Branch Summary

959 Lines of Code · Last analysis 1 minute ago · b773b30f

Quality Gate: Sonar way

Passed

Security 1 Open issues D	Reliability 8 Open issues A	Maintainability 8 Open issues A
Accepted Issues 0	Coverage A few extra steps are needed for SonarQube Cloud to analyze your code coverage. Set up coverage analysis	Duplications 0.0% No conditions set on 1.3k Lines

Security Hotspots

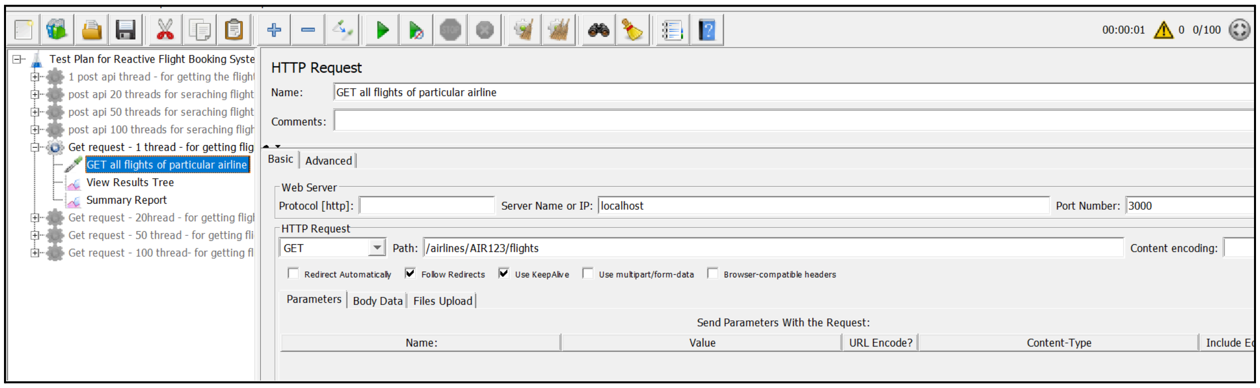
8. JMeter Load Tests

Thread Counts

- 1 Thread
- 20 Threads
- 50 Threads
- 100 Threads

Endpoints Tested:

GET Load Test : airlines/{airline code}/flights :: search all flights of a particular airline



Test Plan for Reactive Flight Booking Syste
- 1 post api thread - for getting the flight
- post api 20 threads for searching flight
- post api 50 threads for searching flight
- post api 100 threads for searching fligh
- Get request - 1 thread - for getting flig
  - GET all flights of particular airline
  - View Results Tree
  - Summary Report
- Get request - 20hread - for getting fligl
- Get request - 50 thread - for getting fli
- Get request - 100 thread- for getting fl

HTTP Request

Name: GET all flights of particular airline

Comments:

Web Server

Protocol [http]:   Server Name or IP: localhost   Port Number: 3000

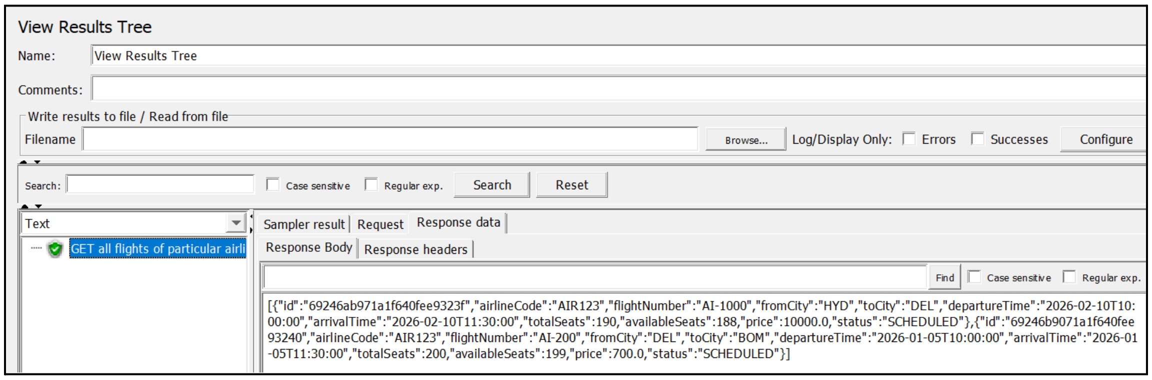
HTTP Request

GET   Path: /airlines/AIR123/flights   Content encoding:

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include E



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename   Log/Display Only: ☐ Errors ☐ Successes

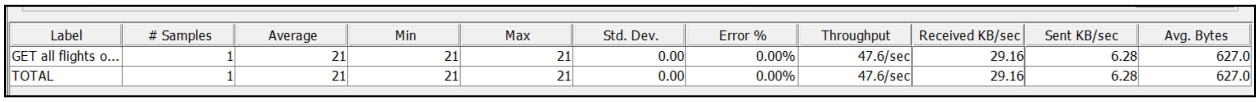
Search: ☐ Case sensitive ☐ Regular exp.

Text

GET all flights of particular airli

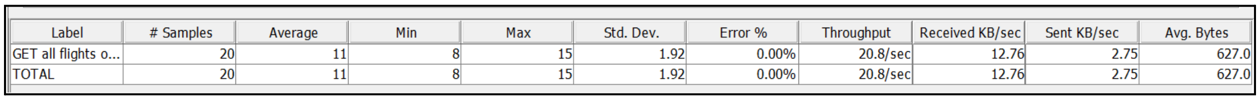
```
[{"id":"69246ab971a1f640fee9323f","airlineCode":"AIR123","flightNumber":"AI-1000","fromCity":"HYD","toCity":"DEL","departureTime":"2026-02-10T10:
00:00","arrivalTime":"2026-02-10T11:30:00","totalSeats":190,"availableSeats":188,"price":10000.0,"status":"SCHEDULED"},{"id":"69246b9071a1f640fee
93240","airlineCode":"AIR123","flightNumber":"AI-200","fromCity":"DEL","toCity":"BOM","departureTime":"2026-01-05T10:00:00","arrivalTime":"2026-01
-05T11:30:00","totalSeats":200,"availableSeats":199,"price":700.0,"status":"SCHEDULED"}]
```

1 thread:



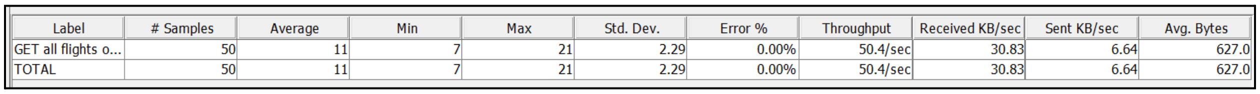
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
GET all flights o...	1	21	21	21	0.00	0.00%	47.6/sec	29.16	6.28	627.0
TOTAL	1	21	21	21	0.00	0.00%	47.6/sec	29.16	6.28	627.0

20 threads:



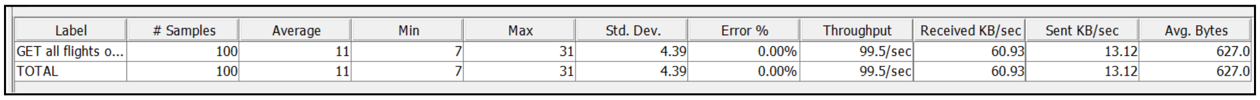
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
GET all flights o...	20	11	8	15	1.92	0.00%	20.8/sec	12.76	2.75	627.0
TOTAL	20	11	8	15	1.92	0.00%	20.8/sec	12.76	2.75	627.0

50 threads:



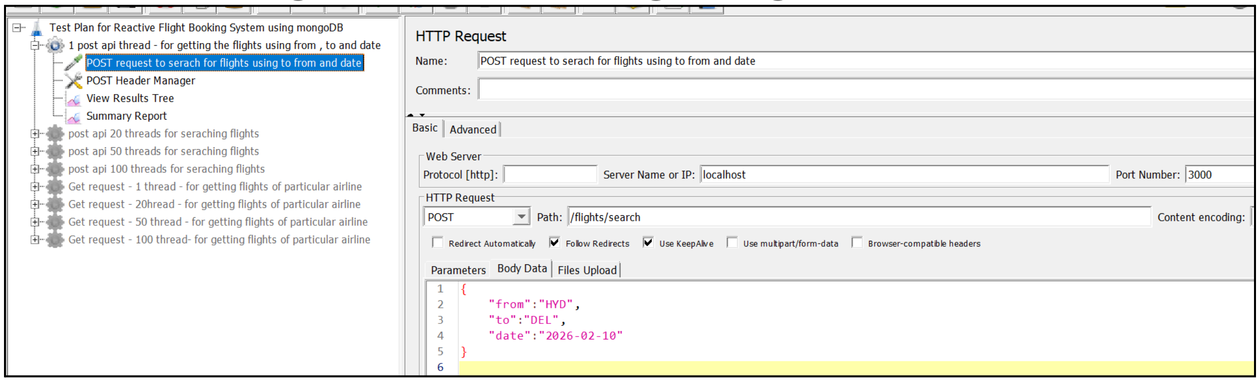
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
GET all flights o...	50	11	7	21	2.29	0.00%	50.4/sec	30.83	6.64	627.0
TOTAL	50	11	7	21	2.29	0.00%	50.4/sec	30.83	6.64	627.0

100 threads:



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
GET all flights o...	100	11	7	31	4.39	0.00%	99.5/sec	60.93	13.12	627.0
TOTAL	100	11	7	31	4.39	0.00%	99.5/sec	60.93	13.12	627.0

POST Load Test : /flights/search :: search all flights using to, from and date



Test Plan for Reactive Flight Booking System using mongoDB
- 1 post api thread - for getting the flights using from , to and date
  - POST request to serach for flights using to from and date
  - POST Header Manager
  - View Results Tree
  - Summary Report
- post api 20 threads for seraching flights
- post api 50 threads for seraching flights
- post api 100 threads for seraching flights
- Get request - 1 thread - for getting flights of particular airline
- Get request - 20hread - for getting flights of particular airline
- Get request - 50 thread - for getting flights of particular airline
- Get request - 100 thread- for getting flights of particular airline

HTTP Request

Name: POST request to search for flights using to from and date

Comments:

Basic | Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 3000

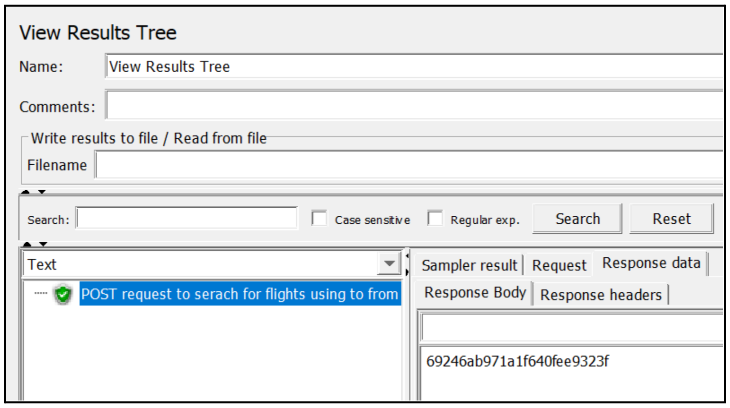
HTTP Request

POST Path: /flights/search Content encoding:

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters | Body Data | Files Upload

```
{
    "from":"HYD",
    "to":"DEL",
    "date":"2026-02-10"
}
```



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Search: ☐ Case sensitive ☐ Regular exp. Search Reset

Text

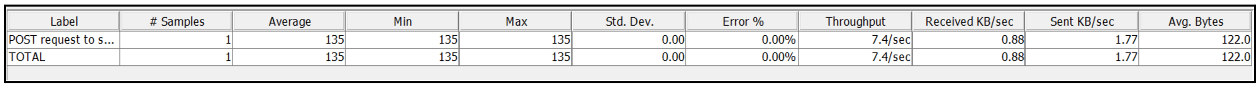
POST request to serach for flights using to from

Sampler result | Request | Response data

Response Body | Response headers

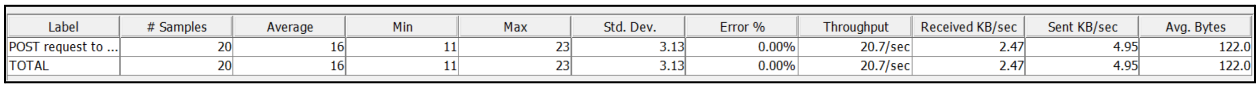
```
69246ab971a1f640fee9323f
```

1 thread :



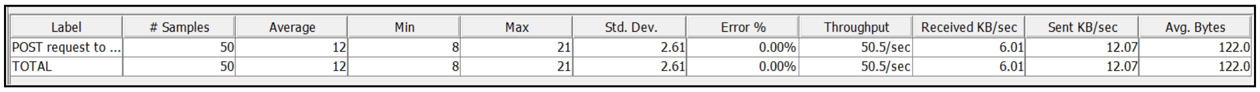
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST request to s...	1	135	135	135	0.00	0.00%	7.4/sec	0.88	1.77	122.0
TOTAL	1	135	135	135	0.00	0.00%	7.4/sec	0.88	1.77	122.0

20 threads:



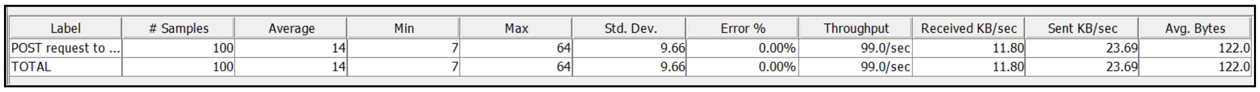
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST request to ...	20	16	11	23	3.13	0.00%	20.7/sec	2.47	4.95	122.0
TOTAL	20	16	11	23	3.13	0.00%	20.7/sec	2.47	4.95	122.0

50 threads:



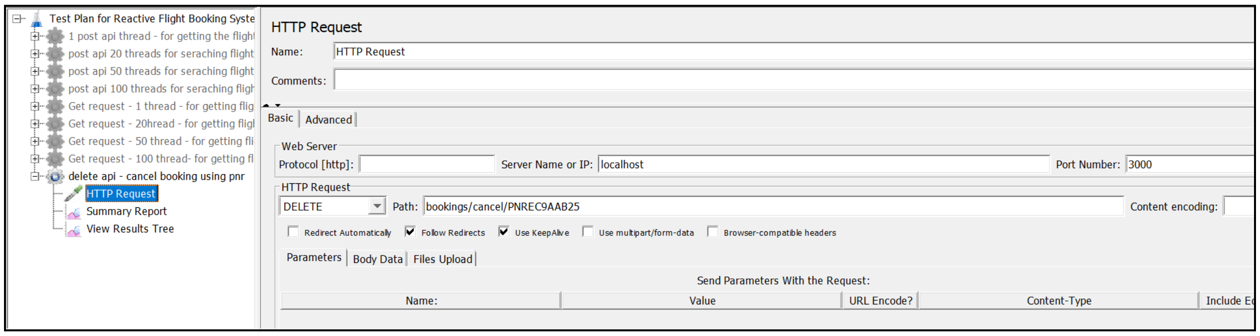
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST request to ...	50	12	8	21	2.61	0.00%	50.5/sec	6.01	12.07	122.0
TOTAL	50	12	8	21	2.61	0.00%	50.5/sec	6.01	12.07	122.0

100 threads:



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST request to ...	100	14	7	64	9.66	0.00%	99.0/sec	11.80	23.69	122.0
TOTAL	100	14	7	64	9.66	0.00%	99.0/sec	11.80	23.69	122.0

DELETE Load Test : /bookings/cancel/{PNR} :: cancel booking using pnr number



```
Test Plan for Reactive Flight Booking Syste
  1 post api thread - for getting the flight
  post api 20 threads for seraching flight
  post api 50 threads for seraching flight
  post api 100 threads for seraching fligt
  Get request - 1 thread - for getting flig
  Get request - 20hread - for getting fligl
  Get request - 50 thread - for getting fli
  Get request - 100 thread- for getting fl
  delete api - cancel booking using pnr
    HTTP Request
    Summary Report
    View Results Tree
```

HTTP Request

Name: HTTP Request

Comments:

Basic | Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 3000

HTTP Request

DELETE Path: bookings/cancel/PNREC9AAB25 Content encoding:

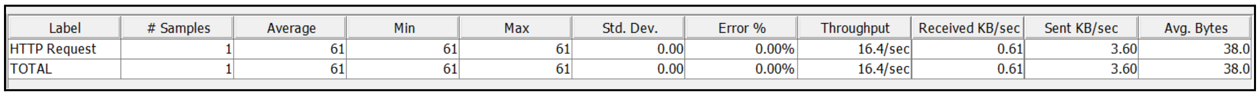
☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters | Body Data | Files Upload

Send Parameters With the Request:

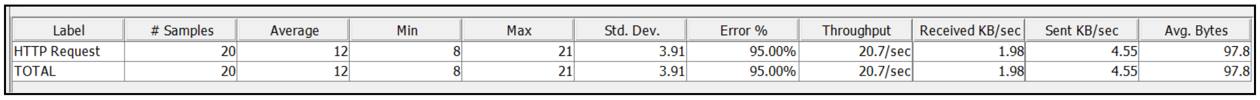
Name:	Value	URL Encode?	Content-Type	Include E

1 thread :

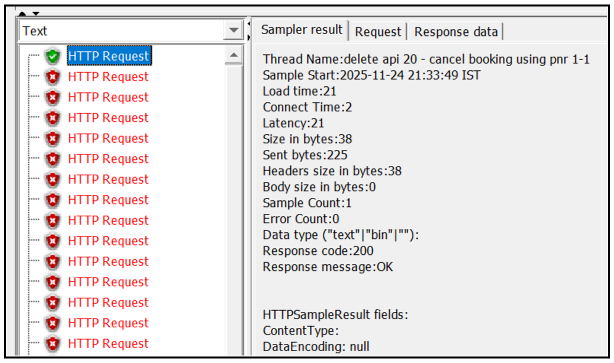


Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	1	61	61	61	0.00	0.00%	16.4/sec	0.61	3.60	38.0
TOTAL	1	61	61	61	0.00	0.00%	16.4/sec	0.61	3.60	38.0

20 threads :



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	20	12	8	21	3.91	95.00%	20.7/sec	1.98	4.55	97.8
TOTAL	20	12	8	21	3.91	95.00%	20.7/sec	1.98	4.55	97.8

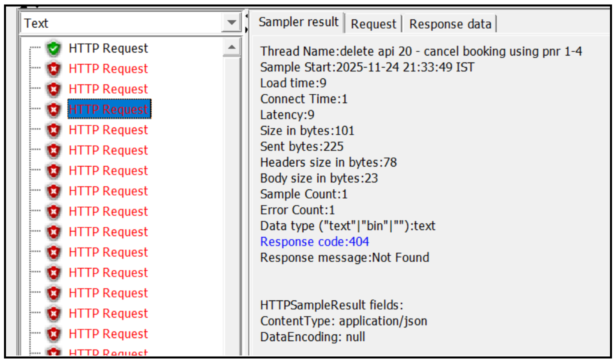


Text

✓ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request

Sampler result | Request | Response data

```
Thread Name:delete api 20 - cancel booking using pnr 1-1
Sample Start:2025-11-24 21:33:49 IST
Load time:21
Connect Time:2
Latency:21
Size in bytes:38
Sent bytes:225
Headers size in bytes:38
Body size in bytes:0
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):
Response code:200
Response message:OK


HTTPSampleResult fields:
ContentType:
DataEncoding: null
```



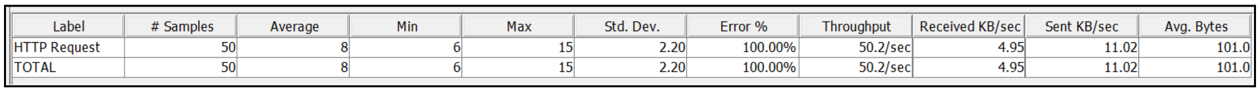
Text

✓ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request
✗ HTTP Request

Sampler result | Request | Response data

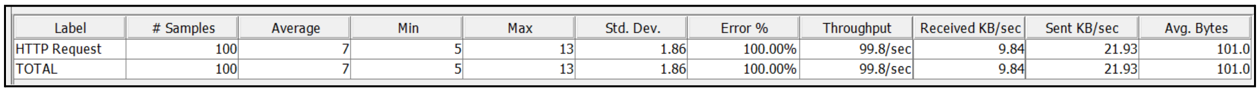
```
Thread Name:delete api 20 - cancel booking using pnr 1-4
Sample Start:2025-11-24 21:33:49 IST
Load time:9
Connect Time:1
Latency:9
Size in bytes:101
Sent bytes:225
Headers size in bytes:78
Body size in bytes:23
Sample Count:1
Error Count:1
Data type ("text"|"bin"|""):text
Response code:404
Response message:Not Found


HTTPSampleResult fields:
ContentType: application/json
DataEncoding: null
```

50 threads :



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	50	8	6	15	2.20	100.00%	50.2/sec	4.95	11.02	101.0
TOTAL	50	8	6	15	2.20	100.00%	50.2/sec	4.95	11.02	101.0

100 threads :



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	100	7	5	13	1.86	100.00%	99.8/sec	9.84	21.93	101.0
TOTAL	100	7	5	13	1.86	100.00%	99.8/sec	9.84	21.93	101.0

9. Conclusion

This project demonstrates a fully reactive, scalable backend system using Spring WebFlux and Reactive MongoDB, covering:

- Robust business logic
- Reactive patterns
- Clean architecture
- Strong validation
- High test coverage
- Performance testing
- Complete API design