

Flight Management using webflux and MongoDB(Screenshots)

1) Post Request (add flight through inventory)

<http://localhost:8080/api/v1.0/flight/airline/inventory/add>

The screenshot shows a terminal window with a POST request to <http://localhost:8080/api/v1.0/flight/airline/inventory/add>. The request body contains the following JSON:

```
1 {  
2   "airlineName": "Indigo",  
3   "flightNumber": "AE101",  
4   "fromPlace": "DELHI",  
5   "toPlace": "MUMBAI",  
6   "departureTime": "2025-11-25T09:00:00",  
7   "arrivalTime": "2025-11-25T11:00:00",  
8 }  
9
```

The response status is 201 Created, with a duration of 2.16 s and a size of 3. The response body is identical to the request body.

2) Post request (to search for flights)

<http://localhost:8080/api/v1.0/flight/search>

The screenshot shows a terminal window with a POST request to <http://localhost:8080/api/v1.0/flight/search>. The request body contains the following JSON:

```
1 {  
2   "fromPlace": "DELHI",  
3   "toPlace": "MUMBAI",  
4   "travelDate": "2025-11-25",  
5   "travelTime": "",  
6   "tripType": "ROUND_TRIP",  
7   "returnDate": "2025-11-28"  
8 }  
9
```

The response status is 201 Created, with a duration of 1.23 s and a size of 709 B. The response body is:

```
1 [  
2   {  
3     "airlineName": "Indigo",  
4     "arrivalTime": "2025-11-25T11:00:00",  
5     "departureTime": "2025-11-25T09:00:00",  
6     "flightNumber": "AE101",  
7     "message": "Only onward flight available. Return flight not found.",  
8     "oneWayPrice": 4500.0,  
9     "returnArrivalTime": null,  
10    "returnDepartureTime": null,  
11  }  
12 ]  
13
```

3) POST request (make bookings)

<http://localhost:8080/api/v1.0/flight/booking/692333797aa12acb5c314279>

The screenshot shows a Postman interface with a POST request to <http://localhost:8080/api/v1.0/flight/booking/692333797aa12acb5c314279>. The request body is JSON:

```
1 {  
2   "email": "amankumar@gmail.com",  
3   "numberOfSeats": 2,  
4   "passengers": [  
5     {  
6       "name": "aman",  
7       "age": 31,  
8       "fareApplied": 3500.0,  
9       "fareCategory": "STUDENT",  
10      }  
11    ]  
12  }  
13 }
```

The response is 201 Created with a duration of 2.65 s.

4) GET request(ticket by PNR)

<http://localhost:8080/api/v1.0/flight/ticket/L7V0HA>

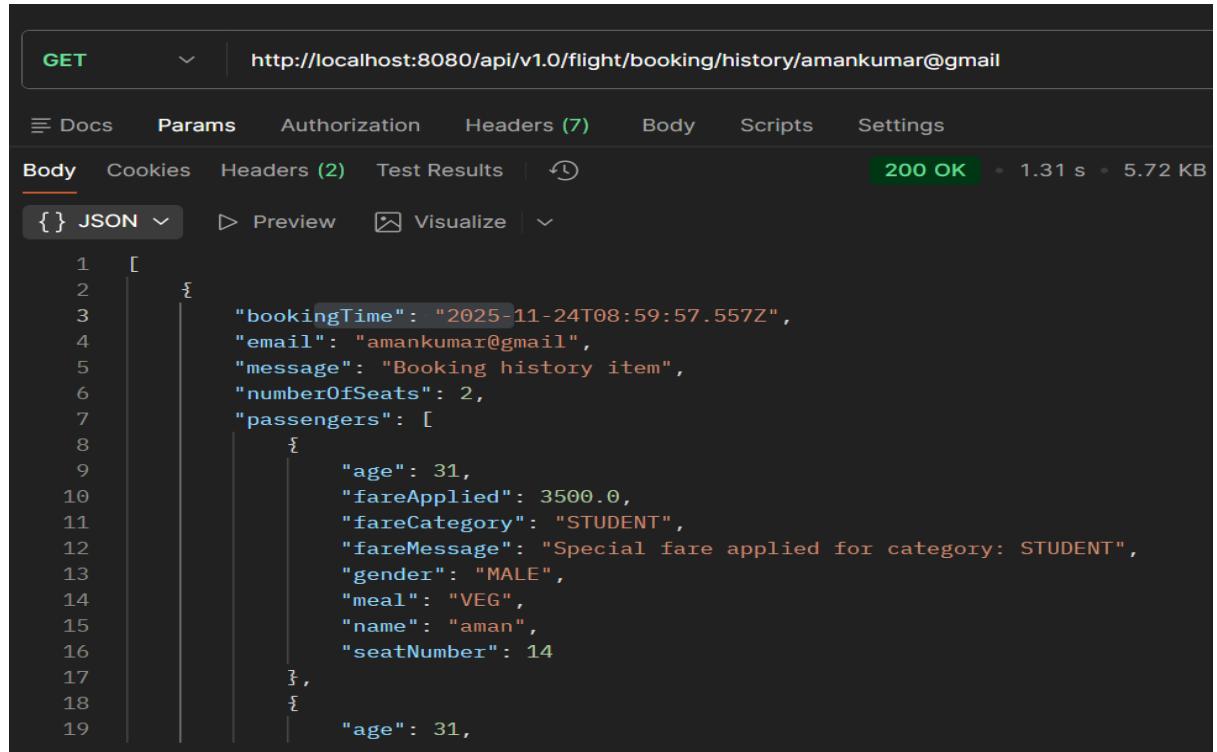
The screenshot shows a Postman interface with a GET request to <http://localhost:8080/api/v1.0/flight/ticket/L7V0HA>. The request body is JSON:

```
1 {  
2   "bookingTime": "2025-11-24T08:59:57.524Z",  
3   "email": "amankumar@gmail.com",  
4   "message": "Booking retrieved",  
5   "numberOfSeats": 2,  
6   "passengers": [  
7     {  
8       "age": 31,  
9       "fareApplied": 3500.0,  
10      "fareCategory": "STUDENT",  
11      "fareMessage": "Special fare applied for category: STUDENT",  
12      "gender": "MALE",  
13      "meal": "VEG",  
14      "name": "aman",  
15      "seatNumber": 14  
16    }  
17  ]  
18 }
```

The response is 200 OK with a duration of 573 ms.

5) GET request(see history by email)

<http://localhost:8080/api/v1.0/flight/booking/history/amankumar@gmail>



GET http://localhost:8080/api/v1.0/flight/booking/history/amankumar@gmail

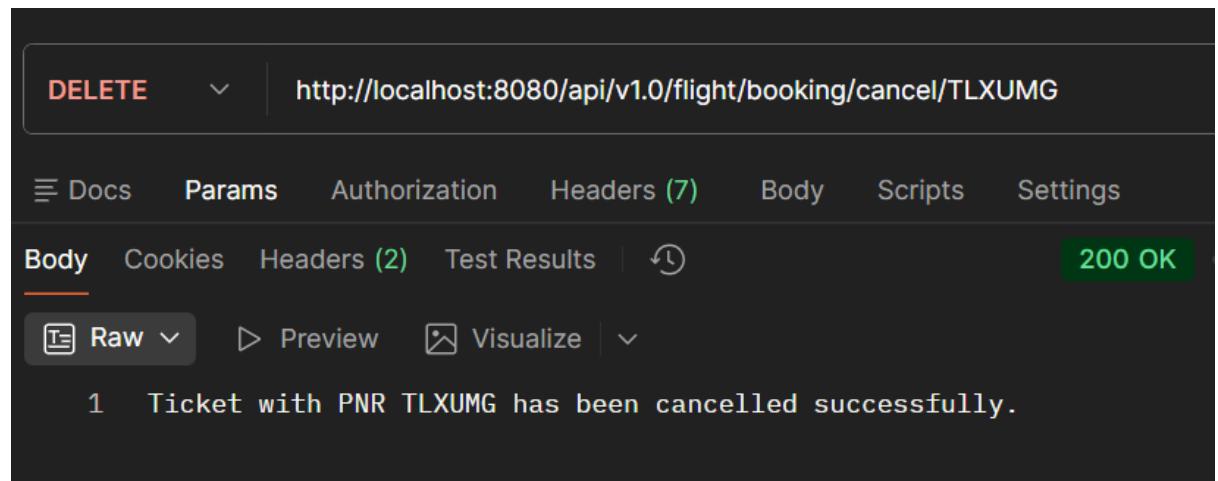
Headers (7): Content-Type: application/json, User-Agent: Postman/1.29.0, Host: localhost:8080, Connection: keep-alive, Cache-Control: no-cache, Postman-Token: 1234567890, Accept: */*

Body (JSON):

```
[{"bookingTime": "2025-11-24T08:59:57.557Z", "email": "amankumar@gmail", "message": "Booking history item", "numberOfSeats": 2, "passengers": [{"age": 31, "fareApplied": 3500.0, "fareCategory": "STUDENT", "fareMessage": "Special fare applied for category: STUDENT", "gender": "MALE", "meal": "VEG", "name": "aman", "seatNumber": 14}], "age": 31, "fareApplied": 3500.0, "fareCategory": "STUDENT", "fareMessage": "Special fare applied for category: STUDENT", "gender": "MALE", "meal": "VEG", "name": "aman", "seatNumber": 14}]
```

6) DELETE request (delete the bookings through pnr)

<http://localhost:8080/api/v1.0/flight/booking/cancel/TLXUMG>



DELETE http://localhost:8080/api/v1.0/flight/booking/cancel/TLXUMG

Headers (7): Content-Type: application/json, User-Agent: Postman/1.29.0, Host: localhost:8080, Connection: keep-alive, Cache-Control: no-cache, Postman-Token: 1234567890, Accept: */*

Body (Raw):

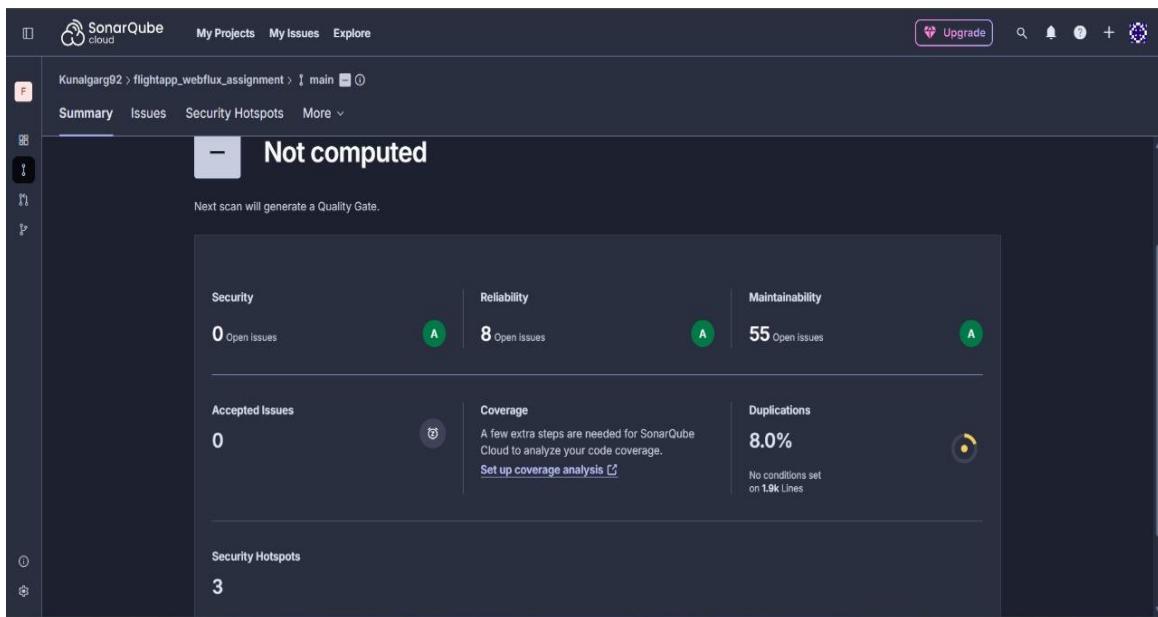
```
1 Ticket with PNR TLXUMG has been cancelled successfully.
```

Jacoco report (code coverage)

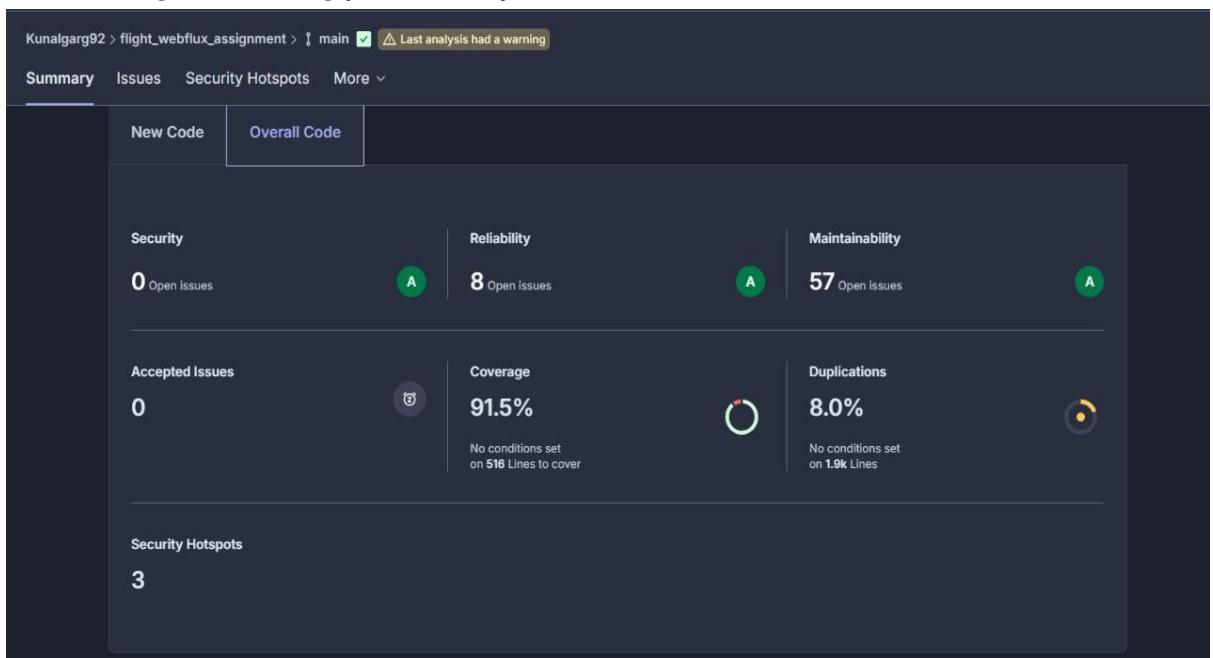
flight_webflux_assignment												Sessions	
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Ctry	Missed	Lines	Missed	Methods	Missed	Classes	
com.flighthapp.webflux.ServiceImplementation	90%	82%	8	52	20	196	1	32	0	32	0	2	
com.flighthapp.webflux.exception	92%	50%	10	30	11	93	7	27	0	3	0	3	
com.flighthapp.webflux.model	92%	n/a	6	68	7	81	6	68	0	4	0	4	
com.flighthapp.webflux.DTO	100%	n/a	0	90	0	132	0	90	0	7	0	7	
com.flighthapp.webflux.Controller	100%	n/a	0	10	0	11	0	10	0	4	0	4	
com.flighthapp.webflux	100%	n/a	0	2	0	3	0	2	0	0	0	1	
Total	127 of 1,787	92%	10 of 46	78%	24	252	38	516	14	229	0	21	

Created with JaCoCo 0.8.10-202304240956

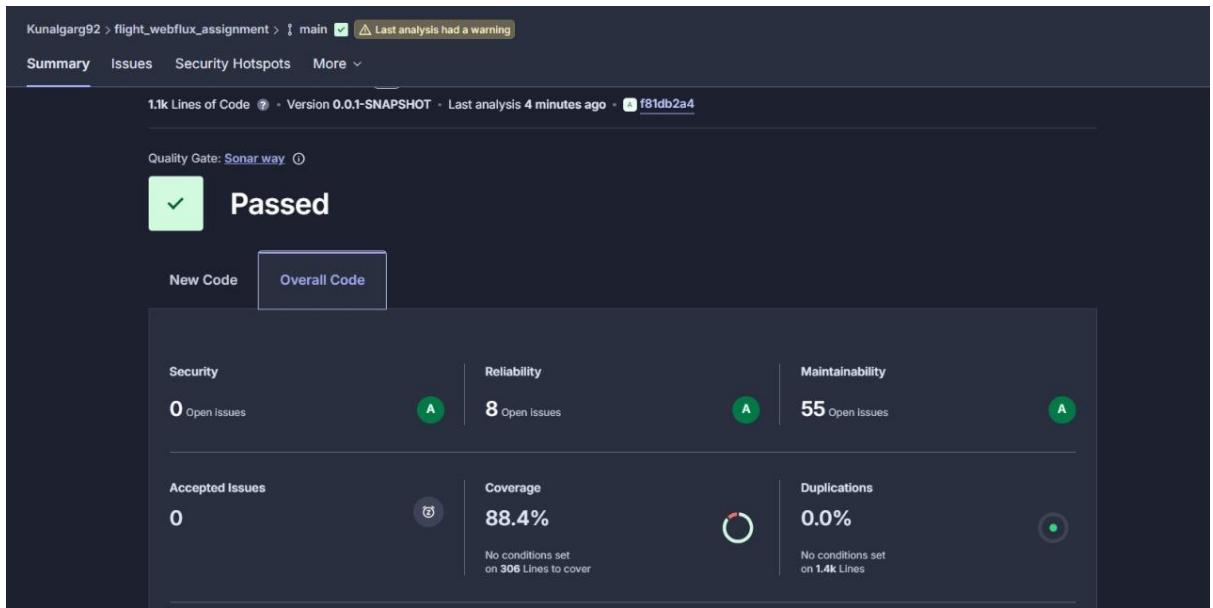
SonarQube attachment (before)



After running in cmd using { sonartoken }

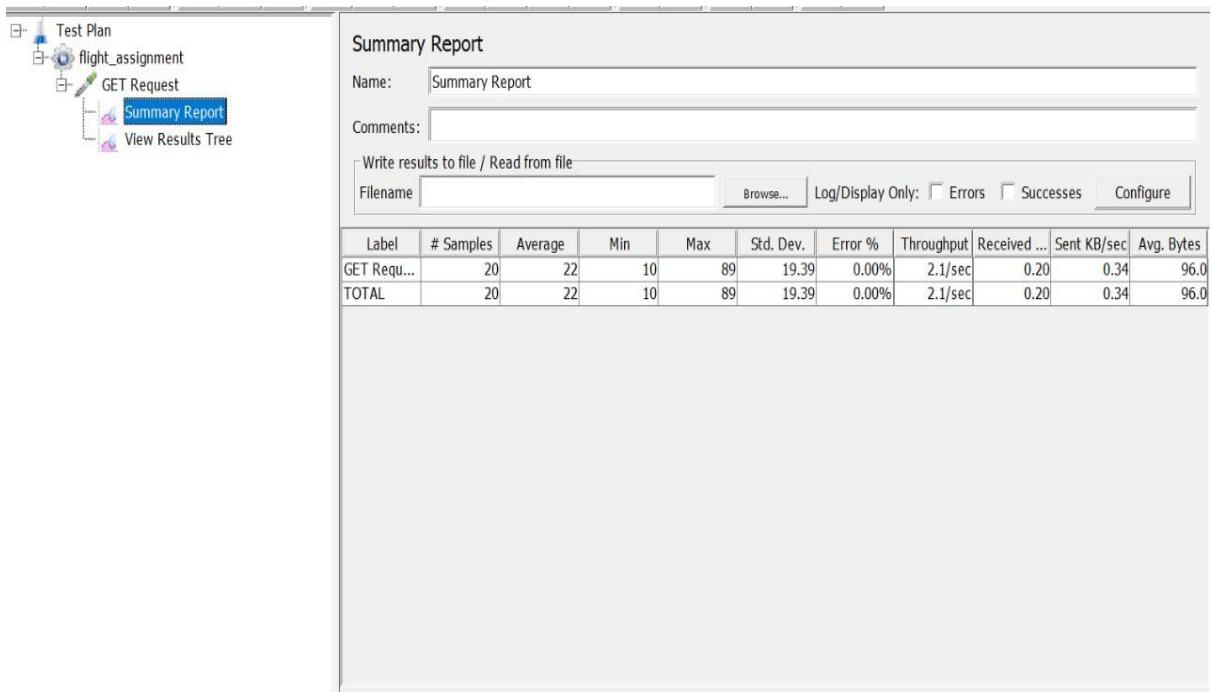


After (removed all duplications using Lombok and code coverage)

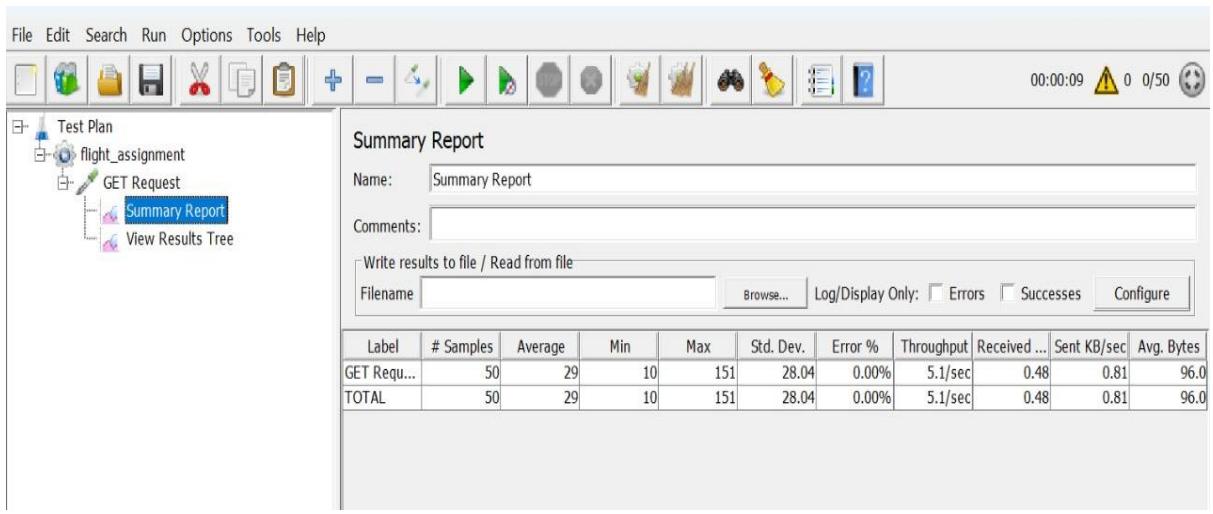


Jmeter Testing (for 20,50,100)

1) for 20 :



2) for 50:



3) For 100:

