

Log files docker

Flight-service

Bookings-service

Api-gateway

Eureka-server

Email sending using SMTP

The screenshot shows an email message from "nsvsujay@gmail.com" to "nsvsujay@gmail.com". The message content is as follows:

Dear Sujay NSVSU,

Your flight booking has been confirmed!

Booking Details:

PNR: PNR09D91A93
Flight: SpiceJet-5
From: Mumbai
To: Goa
Departure: 2025-12-23T07:00
Number of Seats: 2
Seat Numbers: A1, A2
Total Price: ₹5000.00

Thank you for booking with us!

Best Regards,
Flight Booking Team

Postman Testing

The screenshot shows the Postman application interface. On the left, there is a sidebar with a tree view of collections and items, including "Flight Requests" which is currently selected. The main area shows a "Post Request to add airfare" collection. Under this collection, there is a "Post Request to create a flight" item. The "Body" tab of this item is expanded, showing a JSON payload:

```
[{"id": 1, "name": "testflight1", "seat": "A1", "price": 5000}].
```

The status bar at the bottom indicates a successful response with a status code of 200 OK.

The screenshot shows the Microsoft API Management interface. On the left, the 'My Workspaces' sidebar lists various workspaces, including 'Flight Management System'. The main area displays a 'Create Flight' operation under the 'Flight Management System' workspace. The 'HTTP' tab is selected, showing the URL: `https://api.flightservice.com/api/flights`. The request body contains the following JSON payload:

```
1 "airlineId": "6420822042108220871001",
2 "airlineName": "Spiralair",
3 "flightNumber": "56-298",
4 "fromPlace": "Newark",
5 "toPlace": "Dot",
6 "departureTime": "2025-12-22T07:00:00",
7 "arrivalTime": "2025-12-23T08:39:00",
8 "price": 2500.00,
9 "totalSeats": 100,
10 "availableSeats": 100,
11 "tripType": "WOWOW_TRIP"
```

Below the code editor, there are tabs for 'Body', 'Headers', 'Signed ID', 'Metrics', and 'Logs'. The 'Logs' tab is currently active, showing a single log entry:

```
1 FlightId: "V220822042108220871001"
2
3
```

File Window Help

Home Workspaces API Network Search Process

My Workspaces

+ Create workspace

1 Service Plan

- system-test-workspace

- └ Post Request to add airtime
- └ Get Request to see all airtime
- └ Post Request to create a flight
- └ Post Request to search for flights
- └ Get Request to create a booking
- └ Get Request to see bookings by id
- └ Get Request to see booking history
- └ Delete request to remove booking

2 Automations

3 Flight Management System

4 JobPlan

5 JobCollector

6 Workflow

7 Workflows - GitHub

8 Resolving

9 New Collection

10 New Collection

11 New Collection

12 New Collection

13 NewCollection-1

14 Resolved Message

15 SWIIS

16 Tutorial-Mother

POST → http://localhost:8080/api/test

Headers

Body

Request Response Headers Test Results

200 OK 10ms 200 OK View Response

Step	Value	Headers	Test Results
1	[{"id": "632db27ca428c0380f3e430", "arrivalId": "A022111111111111111", "arrLocation": "Delhi", "flightNumber": "98-123", "inFlight": "Mumbai", "outLocation": "Hyderabad"}]	[{"Content-Type": "application/json"}]	[{"Status": "Success", "Message": "Response status code was 200 and response time was 10ms"}]

0 rows of data 0 rows 0 rows 0 rows 0 rows 0 rows

File Window Help

Home Workspaces API Network Search Process

My Workspaces

+ Create workspace

1 Service Plan

- system-test-workspace

- └ Post Request to add airtime
- └ Get Request to see all airtime
- └ Post Request to create a flight
- └ Post Request to search for flights
- └ Post Request to search for booking
- └ Get Request to see bookings by id
- └ Get Request to see booking history
- └ Delete request to remove booking

2 Automations

3 Flight Management System

4 JobPlan

5 JobCollector

6 Workflow

7 Workflows - GitHub

8 Resolving

9 New Collection

10 New Collection

11 New Collection

12 New Collection

13 NewCollection-1

14 Resolved Message

15 SWIIS

16 Tutorial-Mother

POST → http://localhost:8080/api/test

Headers

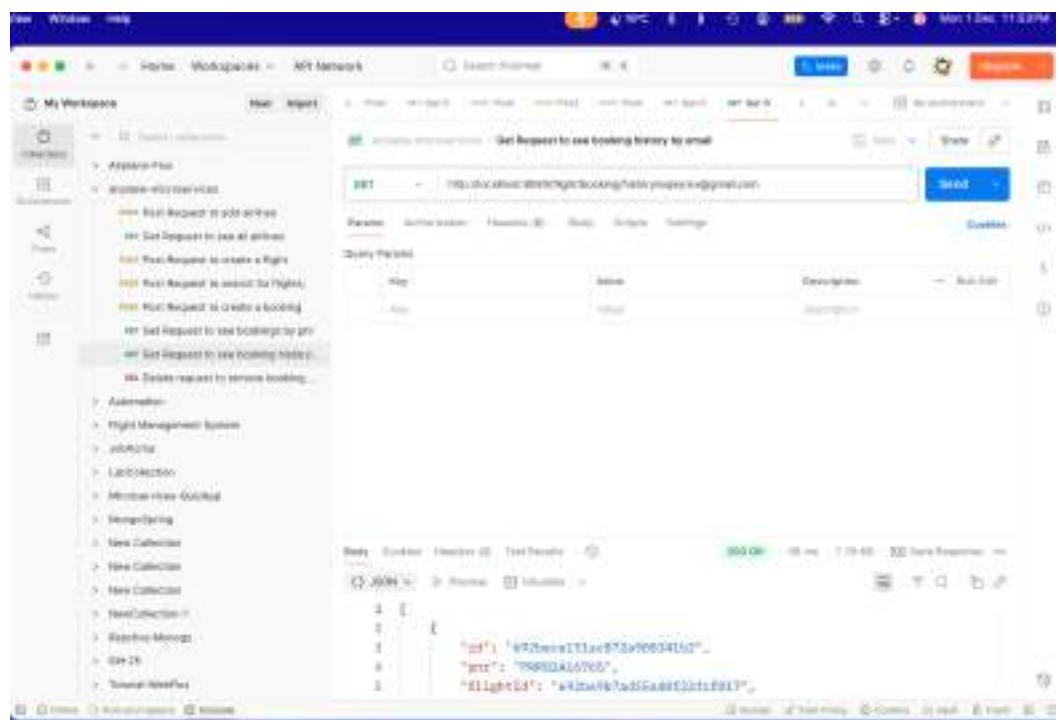
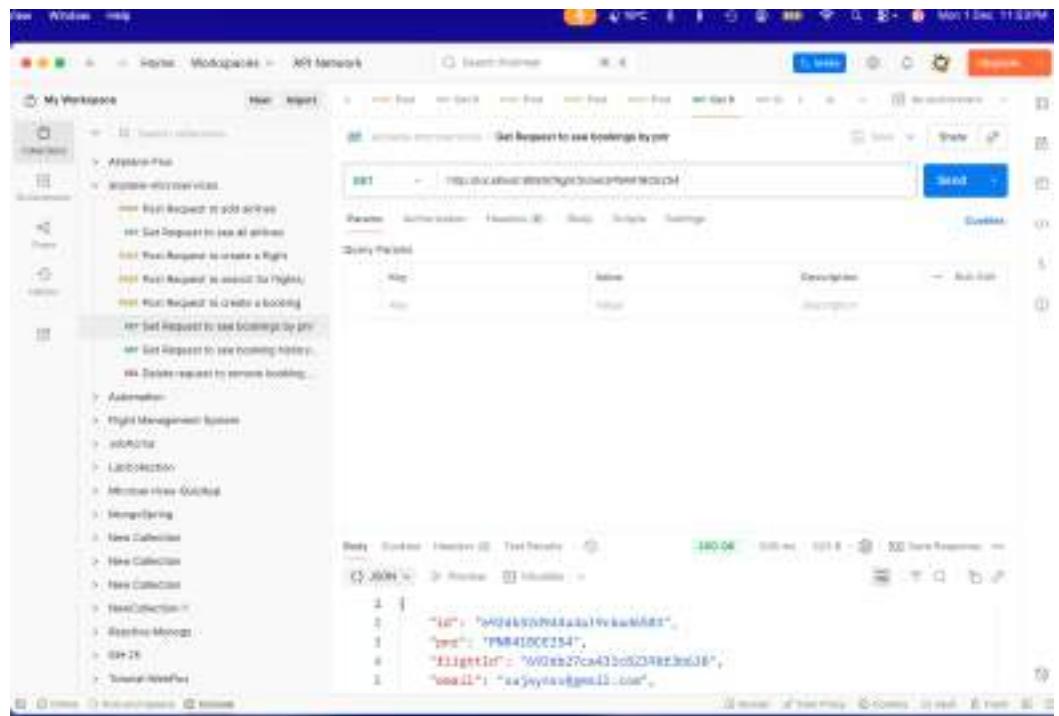
Body

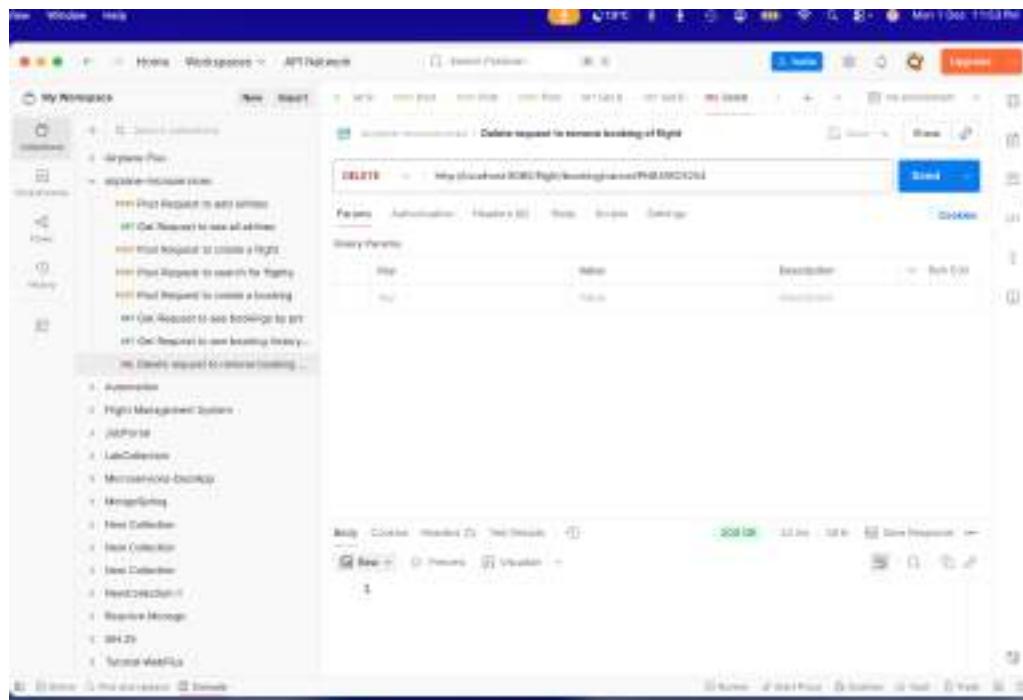
Request Response Headers Test Results

200 OK 10ms 200 OK View Response

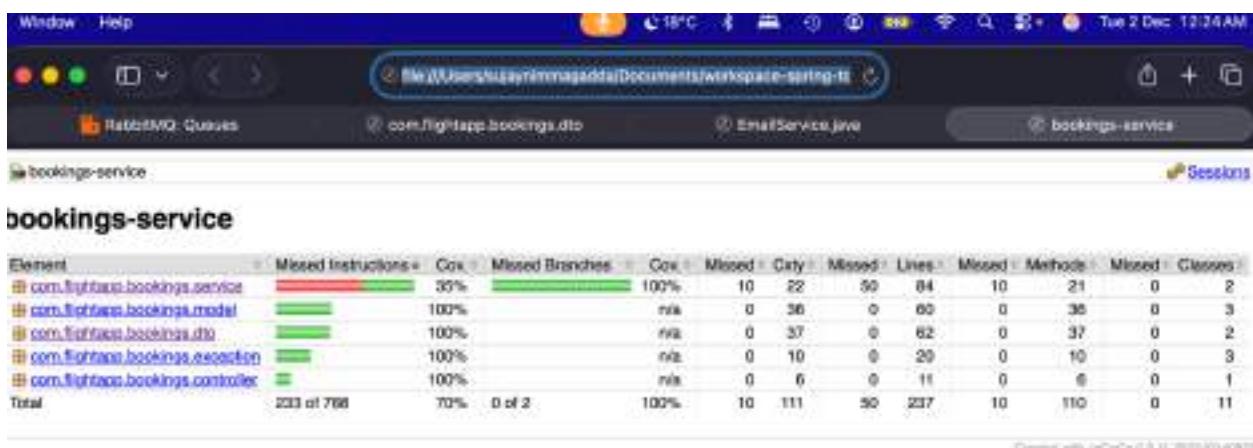
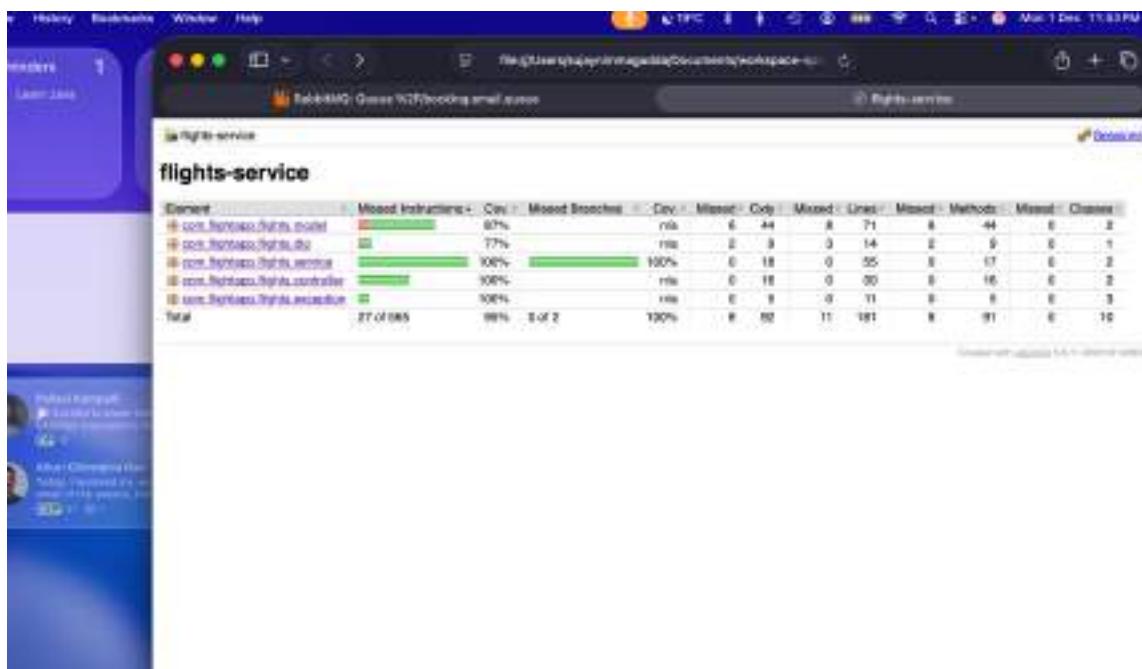
Step	Value	Headers	Test Results
1	{"post": "986410C325A1"}	[{"Content-Type": "application/json"}]	[{"Status": "Success", "Message": "Response status code was 200 and response time was 10ms"}]

0 rows of data 0 rows 0 rows 0 rows 0 rows 0 rows





Jacoco Report



RabbitMq

The screenshot shows the RabbitMQ Management UI with the title "RabbitMQ - RabbitMQ 4.2.1 899242.0". The main navigation tabs are "Overview", "Connections", "Channels", and "Exchanges", with "Queues and Streams" being the active tab. The "Overview" section displays metrics for queues and streams. Under "Queues", there is one queue named "bookings-streams-class" with 0 messages. Under "Streams", there is one stream named "bookings-streams-class" with 0 messages. The "Details" section shows the configuration for this stream.

Attribute	Value
Algorithm	RoundRobin
Broker	bookings-streams-class
Consumer count	0
Consumer backlog	0
Consumer rejects	0

The screenshot shows the "Queues and Streams" page of the RabbitMQ Management UI. The title is "RabbitMQ - RabbitMQ 4.2.1 899242.0". The "Queues" section lists one queue named "bookings-streams-class" with 0 messages. The "Streams" section lists one stream named "bookings-streams-class" with 0 messages. At the bottom, there is a link to "Add a new queue".

Virtual host	Name	Type	Plugins	Status	Reads	Writes	Total	Message rate	Delivery / deliver / get ack
/	bookings-streams-class	0		Normal	0	0	0	0.00/s	0.00/s

Safari Help

localhost

RabbitMQ Queue N2 (BookingEmailQueue) 2020-12-01 10:25 AM

Refreshed 2020-12-01 09:25:38 [Refresh every 5 seconds] Virtual host: / User guest [Log out]

RabbitMQ 4.2.2 Erlang 26.2 Cluster rabbit@dale-MacBook-Air-2

Overview Connections Channels Exchanges Queues and Streams **Queues and Streams**

Actions

Warning: getting messages from a queue is a destructive action.

Ack mode: Multi message receive (auto) Manual

Polling: Auto polling (every 5s) Manual

Messages: 4

Get messages

Message 1

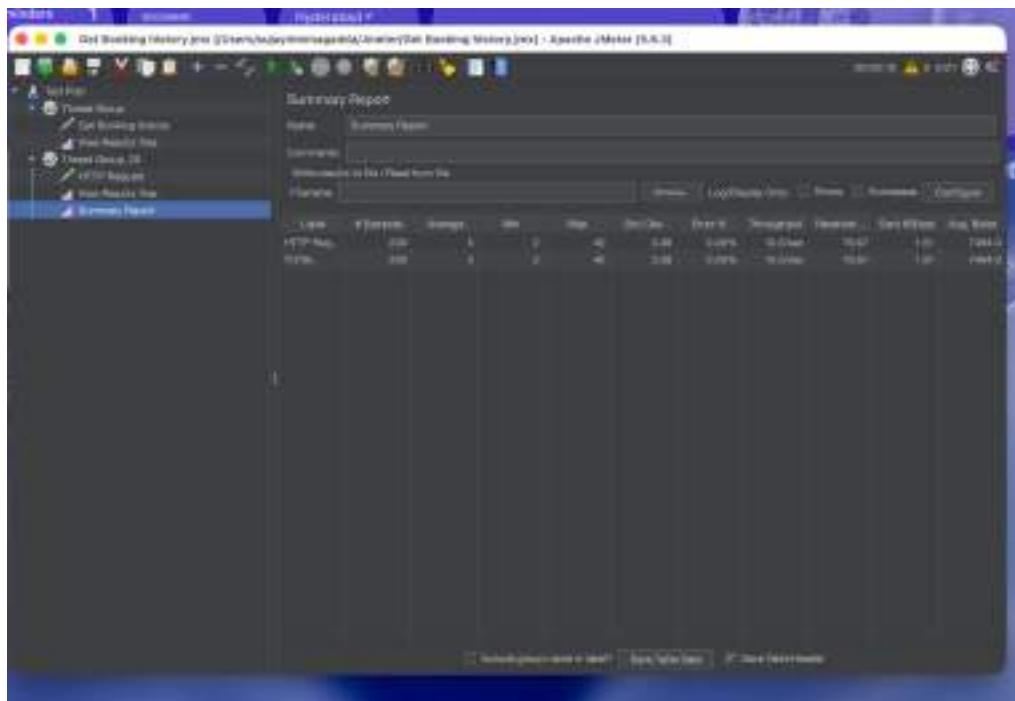
The server reported 22 messages remaining.

Exchange	(AMQP default)
Routing key	booking_email_queue
Delivery count	1
Properties	<pre>priority: 0 delivery_mode: 2 headers: {\"Typeid\": \"com.RabbitApp.bookings.v1.EmailNotification\"} content_encoding: UTF-8 content_type: application/json</pre>
Payload	<pre>{"to": "testuser@gmail.com", "subject": "Flight Booking Confirmation - 3991-8800354310", "body": "Dear User, Your flight booking has been confirmed."}</pre>

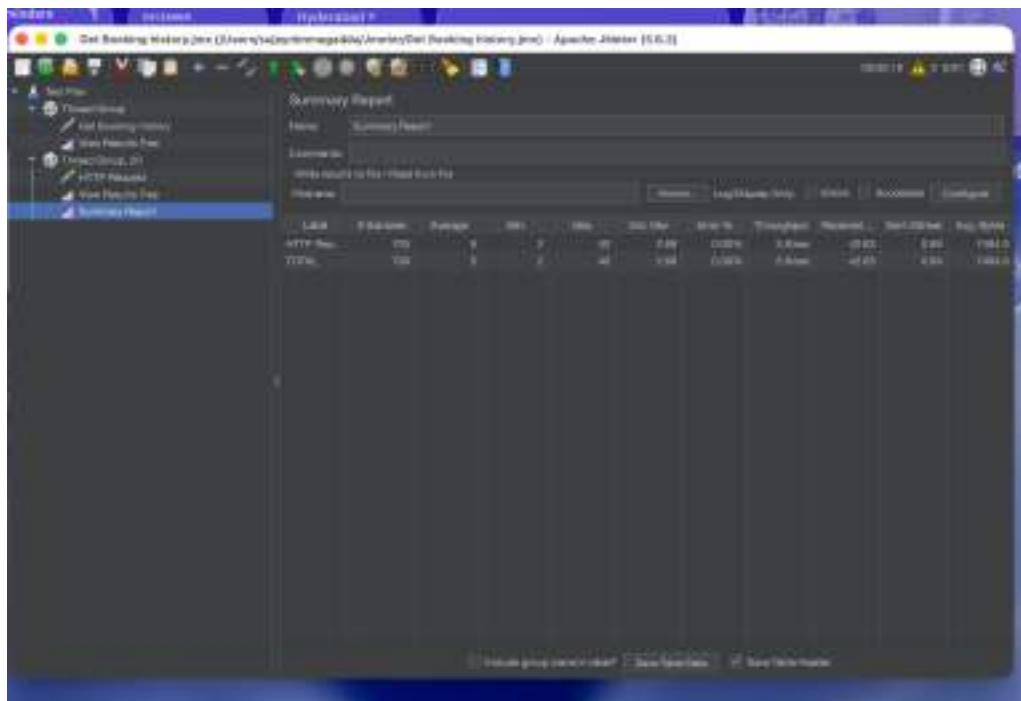
Move messages

Details

Jmeter



20 requests



50 requests

SonarQube

The screenshot shows the SonarQube web interface for the project "Right-bounding-boxes". The main header includes links for "My Projects", "My Issues", "Explore", and a "Logout" button. Below the header, the project name "Right-bounding-boxes" is displayed along with a "Scan start" date of "2019-05-10 10:00:00 UTC" and a "Scan end" date of "2019-05-10 10:00:00 UTC".

The dashboard summary indicates the project has passed its quality gate. Key metrics shown include:

- Security:** 11 Open issues (Severity: **B**)
- Reliability:** 0 Open issues (Severity: **A**)
- Maintainability:** 5 Open issues (Severity: **A**)
- Associated Issues:** 0
- Coverage:** A few extra steps are needed for SonarQube Cloud to analyze your static coverage. [See the coverage analysis](#).
- Duplications:** 10.6% (70,000/659,981 lines of code)

At the bottom, there is a link to "Security Hotspots".

