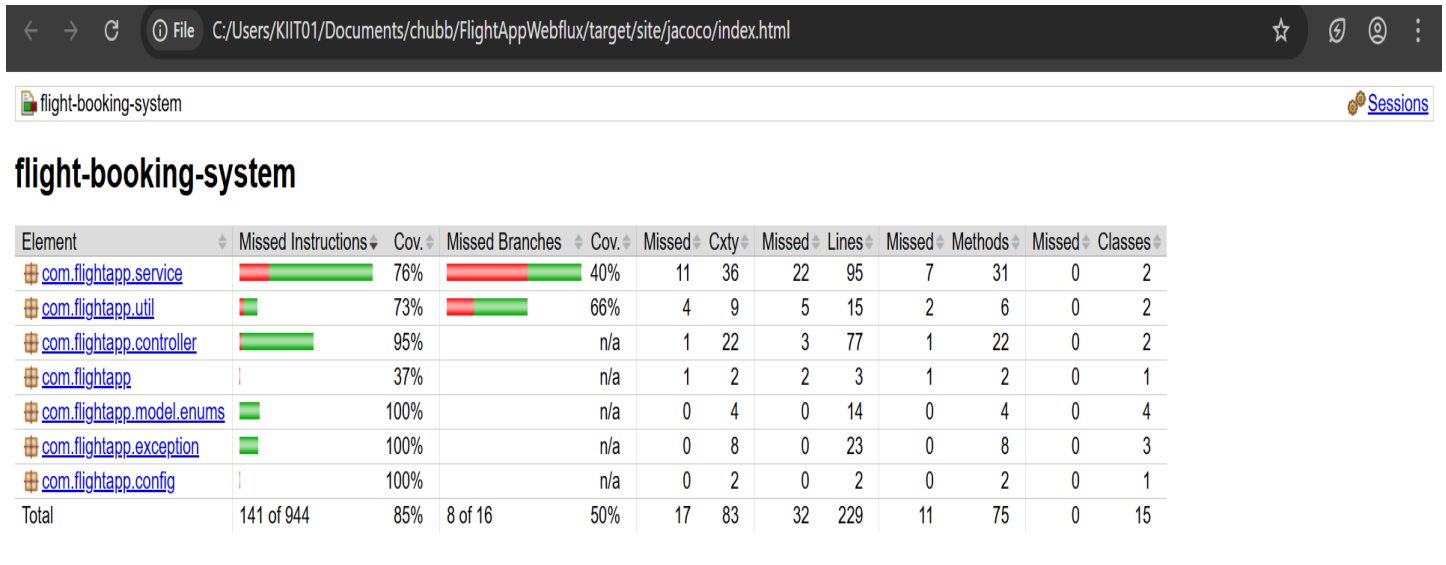


# Flight Management System using webflux and mongodb:

## Jacoco Report:

Code Coverage: 85%

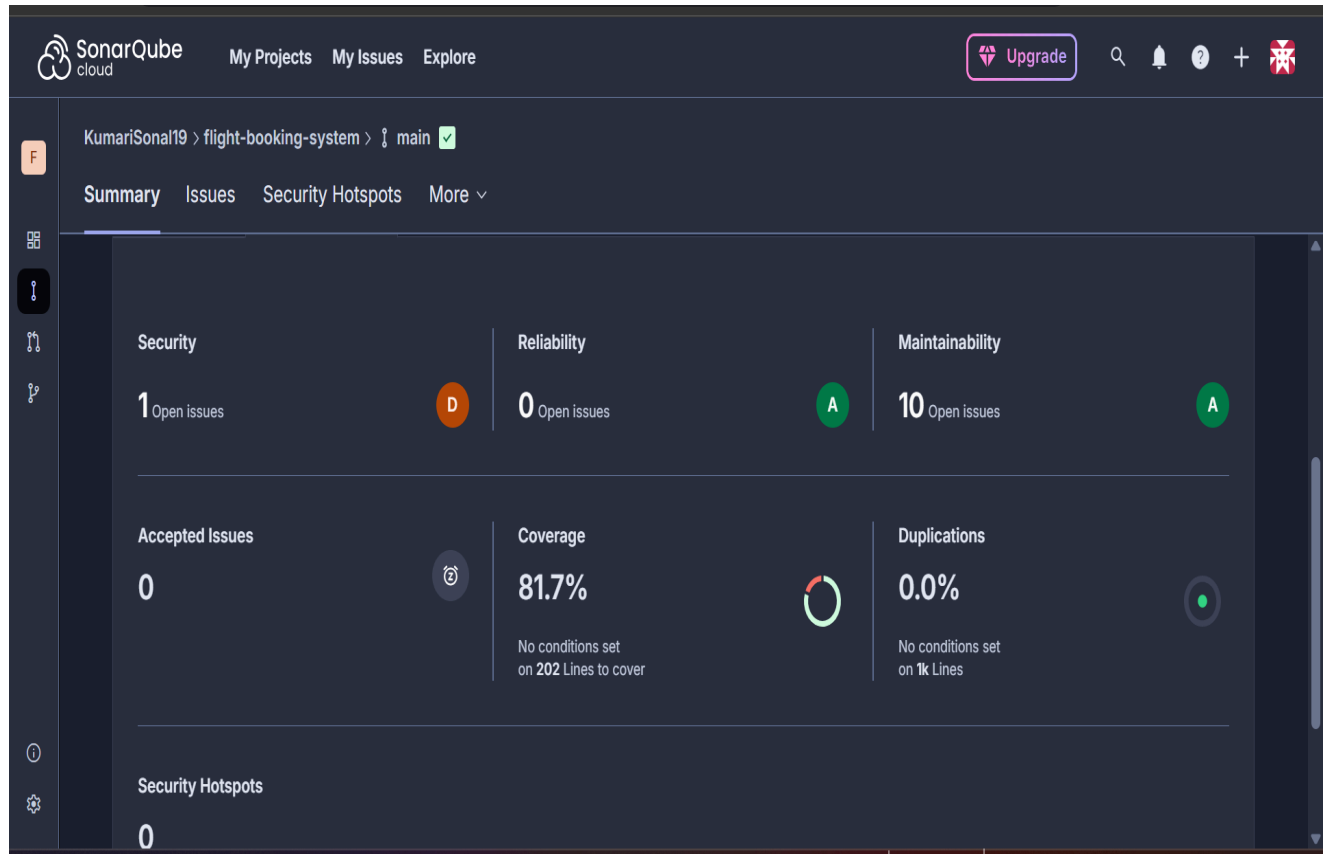


Created with JaCoCo 0.8.11.202310140853

## SONARQUBE REPORT (Before):

Code Coverage: 81.7

Total Issues: 11



## ISSUES:

KumariSonal19 > flight-booking-system > main

Summary Issues Security Hotspots More

Filters

Software quality

Security1

Reliability0

Maintainability10

Severity ?

Blocker0

High1

Medium1

Low2

src/.../java/com/flightapp/controller/FlightController.java

☐ Replace this persistent entity with a simple POJO or DTO object.

Consistency

SecurityHigh

cwe spring +

Open KumariSonal19 L27 10min effort 8 hours ago Vulnerability Critical

src/.../java/com/flightapp/exception/GlobalExceptionHandler.java

☐ Remove the parentheses around the "error" parameter

Intentionality

MaintainabilityLow

java8 +

Open KumariSonal19 L35 2min effort 6 minutes ago Code Smell Minor

src/main/java/com/flightapp/util/ValidationUtil.java

KumariSonal19 > flight-booking-system > main

Summary Issues Security Hotspots More

### Filters

Software quality

- Security 1
- Reliability 0
- Maintainability 10

Severity ?

- Blocker 0
- High 1
- Medium 1
- Low 2

src/.../java/com/flightapp/model/dto/BookingRequestTest.java

☐ Remove this 'public' modifier. Intentionality

Maintainability Info junit tests +

☐ Open KumariSonal19 L12 • 2min effort • 1 hour ago • Code Smell • Info

src/.../java/com/flightapp/model/dto/BookingResponseTest.java

☐ Remove this 'public' modifier. Intentionality

Maintainability Info junit tests +

☐ Open KumariSonal19 L14 • 2min effort • 1 hour ago • Code Smell • Info

src/.../java/com/flightapp/model/dto/FlightSearchRequestTest.java

☐ Remove this 'public' modifier. Intentionality

KumariSonal19 > flight-booking-system > main

Summary Issues Security Hotspots More

### Filters

Software quality

- Security 1
- Reliability 0
- Maintainability 10

Severity ?

- Blocker 0
- High 1
- Medium 1
- Low 2

src/.../java/com/flightapp/model/dto/FlightSearchRequestTest.java

☐ Remove this 'public' modifier. Intentionality

Maintainability Info junit tests +

☐ Open KumariSonal19 L10 • 2min effort • 1 hour ago • Code Smell • Info

src/.../com/flightapp/model/entity/BookingEntityTest.java

☐ Remove this 'public' modifier. Intentionality

Maintainability Info junit tests +

☐ Open KumariSonal19 L13 • 2min effort • 1 hour ago • Code Smell • Info

src/.../com/flightapp/model/entity/FlightEntityTest.java

☐ Remove this 'public' modifier. Intentionality

Filters

Software quality

Security	1
Reliability	0
Maintainability	10

Severity

Blocker	0
High	1
Medium	1
Low	2

Open KumariSonal19 L9 2min effort 1 hour ago Code Smell Info

src/.../java/com/flightapp/service/BookingServiceImplTest.java

☐ Remove this unnecessary import: same package classes are always implicitly imported. Intentionality

Maintainability Low unused +

Open KumariSonal19 L13 1min effort 6 minutes ago Code Smell Minor

11 of 11 shown

## SONARQUBE REPORT (After):

All issues resolved

SummaryIssuesSecurity HotspotsMore ▾

Filters

▼ Software quality

Security0

Reliability0

Maintainability0

▼ Severity ⓘ

Blocker0

High0

Medium0

Low0

Select issues for bulk actions

☐ Change

Select issues ▾ ▲

Navigate to issue ◀ ▶

↻ 0 issues 0 effort

No Issues. Hooray!

© 2018-2025

[SonarSource Sàrl.](#)

All rights reserved.

Terms

Pricing

Privacy

Cookie

Security

Community

Documentation

Contact

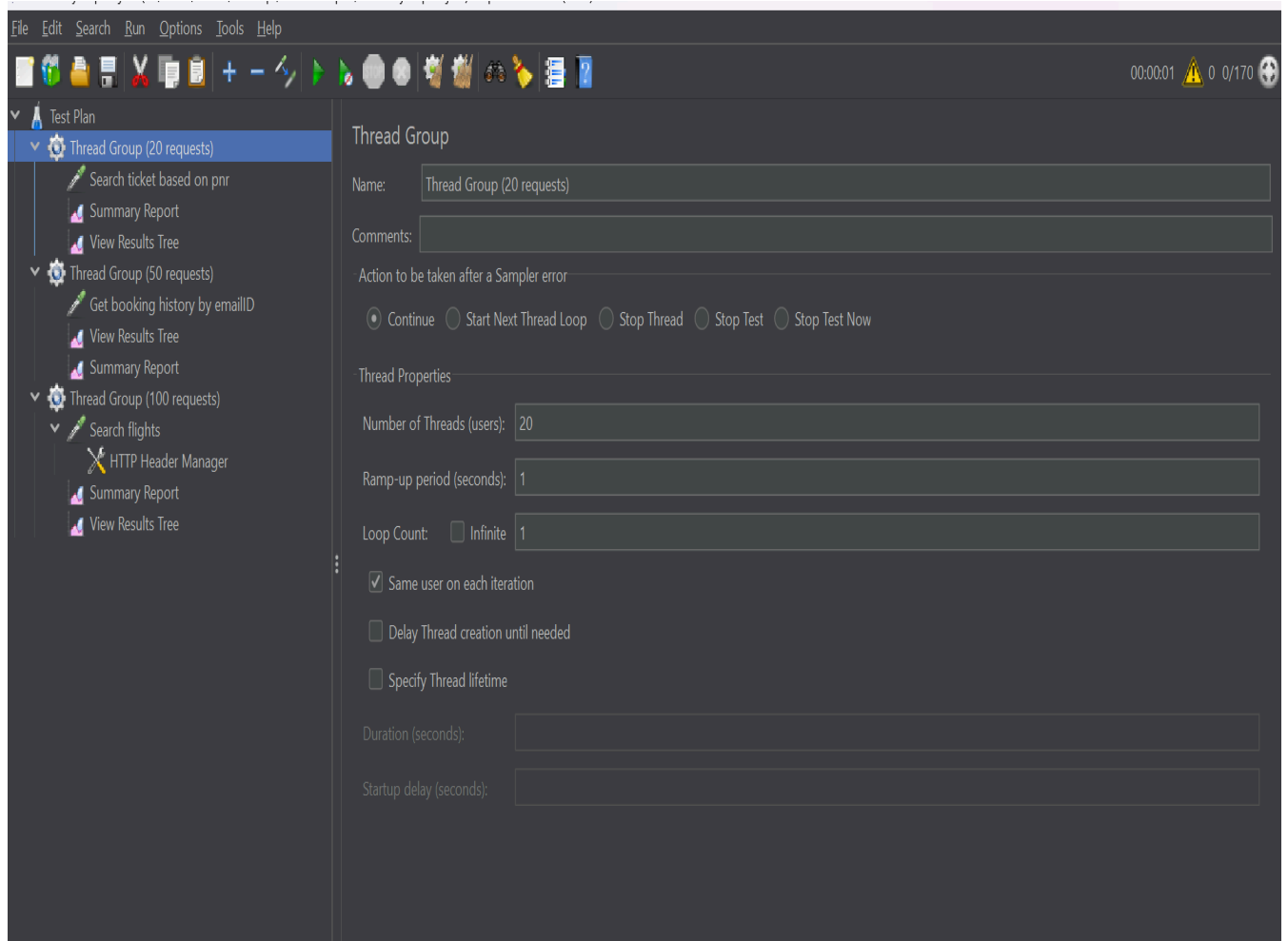
Status

About

us

# JMETER TESTING

## Case 1: 20 requests



Test Plan

Thread Group (20 requests)

Search ticket based on pnr

Summary Report

View Results Tree

Thread Group (50 requests)

Get booking history by emailID

View Results Tree

Summary Report

Thread Group (100 requests)

Search flights

HTTP Header Manager

Summary Report

View Results Tree

HTTP Request

Name: Search ticket based on pnr

Comments:

Basic

Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8081

HTTP Request

GET Path: api/flight/ticket/FLT25112588EB65 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 Equals?
-------	-------	-------------	--------------	-----------------

Detail

Add

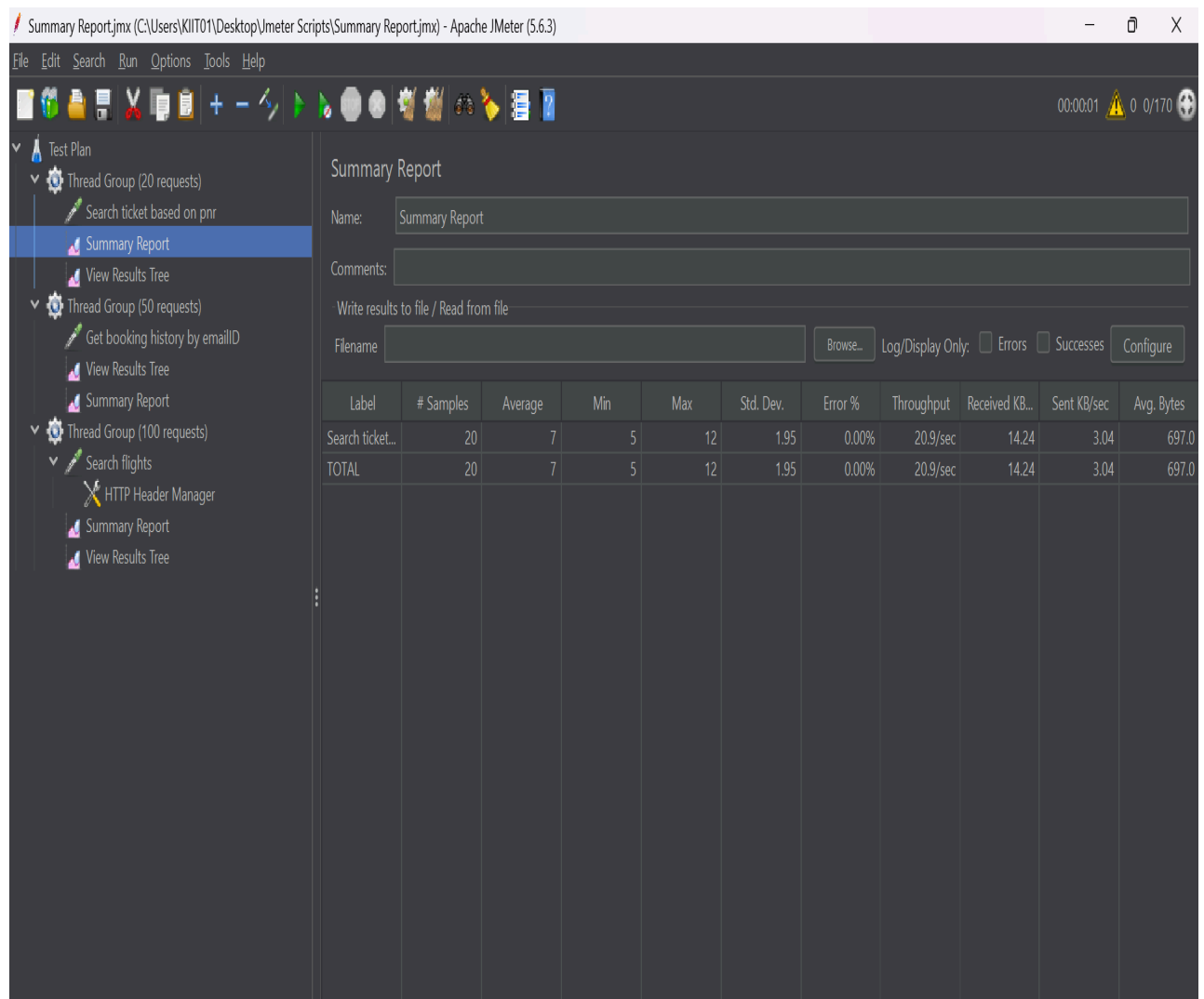
Add from Clipboard

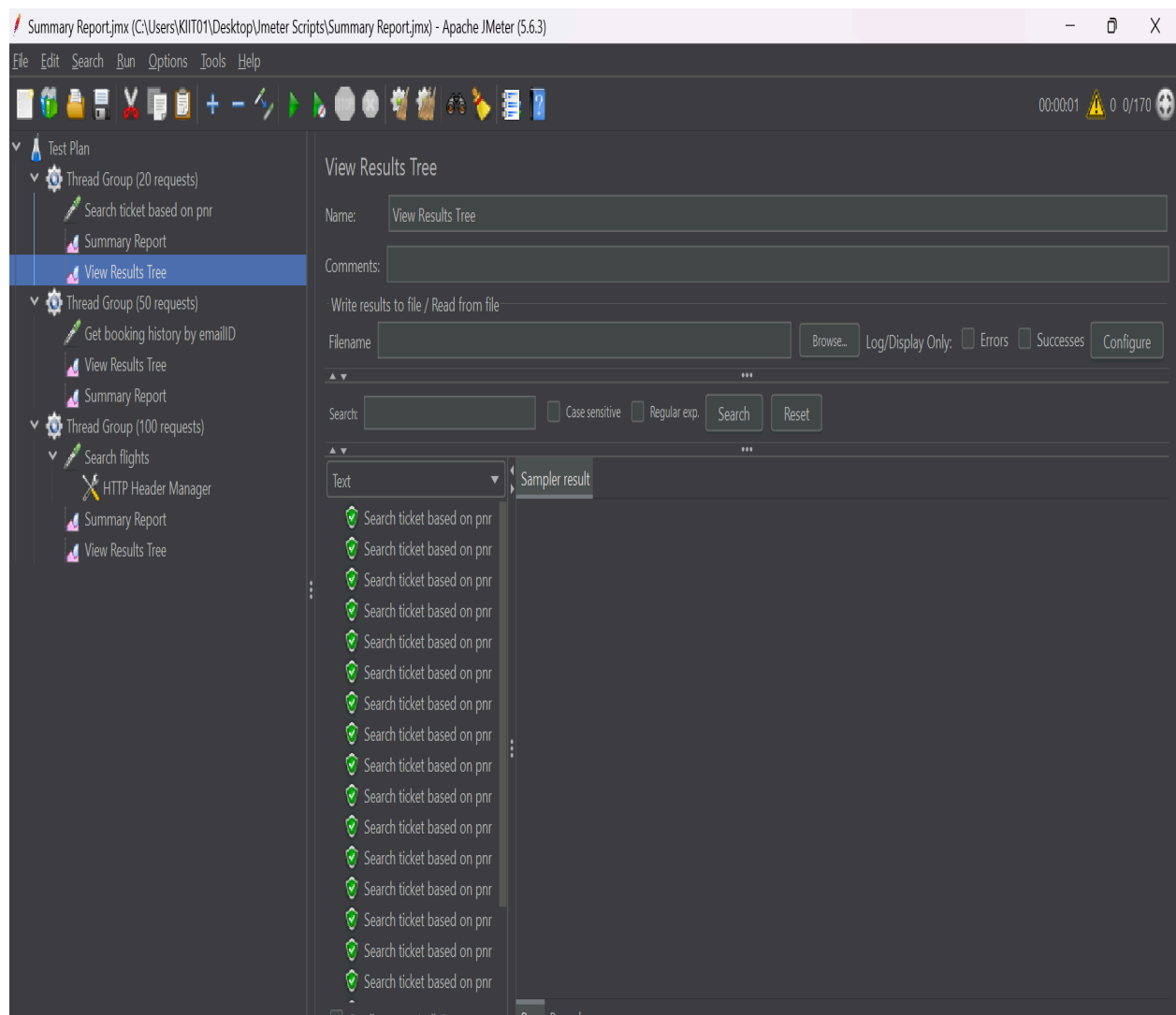
Delete

Up

Down







## Case 2: 50 requests

Summary Report.jmx (C:\Users\KMIT01\Desktop\Jmeter Scripts\Summary Report.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:01 0 0/170

Test Plan

- Thread Group (20 requests)
  - Search ticket based on pnr
  - Summary Report
  - View Results Tree
- Thread Group (50 requests)
  - Get booking history by emailID**
  - View Results Tree
  - Summary Report
- Thread Group (100 requests)
  - Search flights
    - HTTP Header Manager
    - Summary Report
    - View Results Tree

### HTTP Request

Name: Get booking history by emailID

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8081

HTTP Request

GET Path: /api/flight/booking/history/simran.sharma@example.com Content encoding:

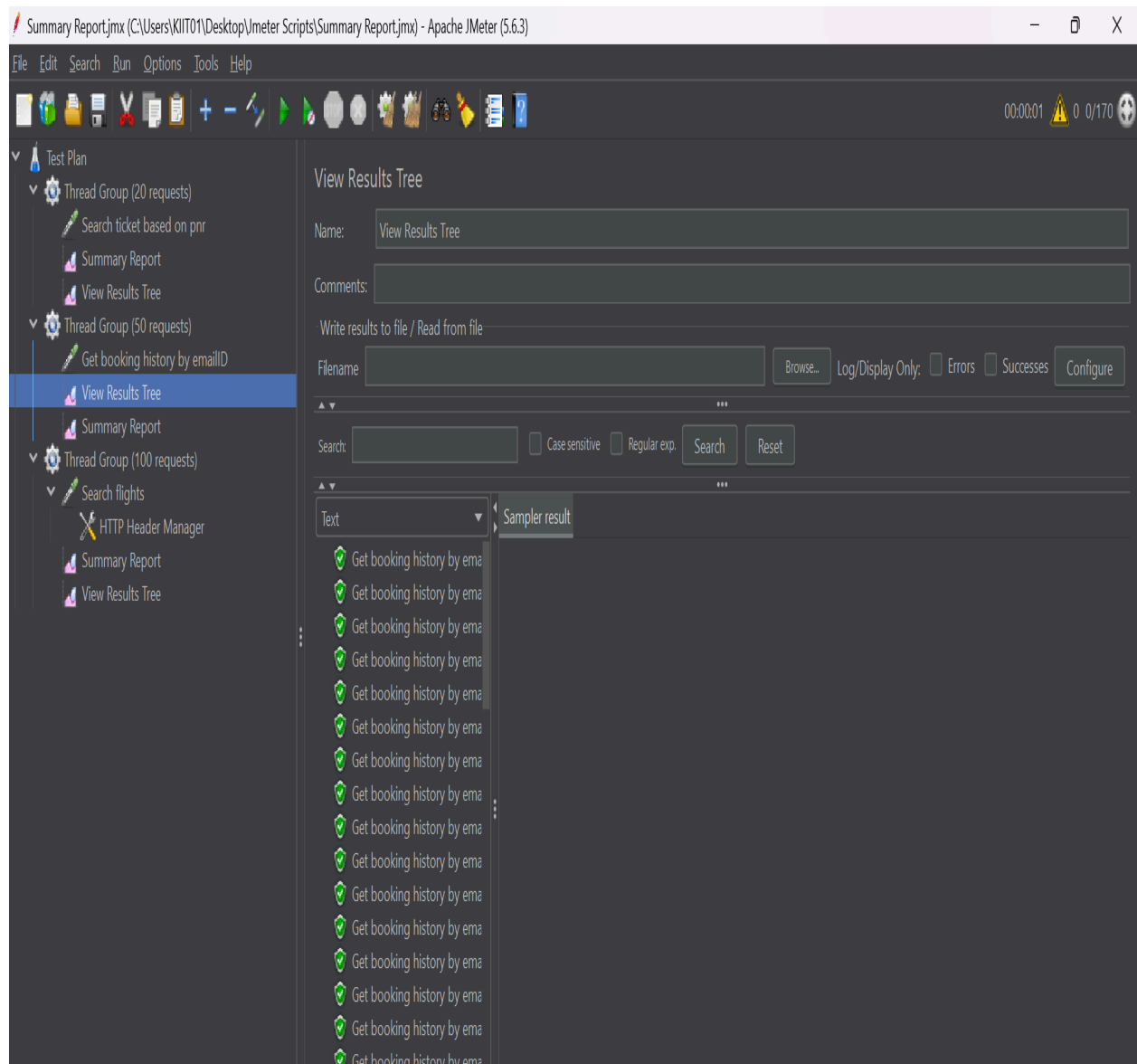
☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

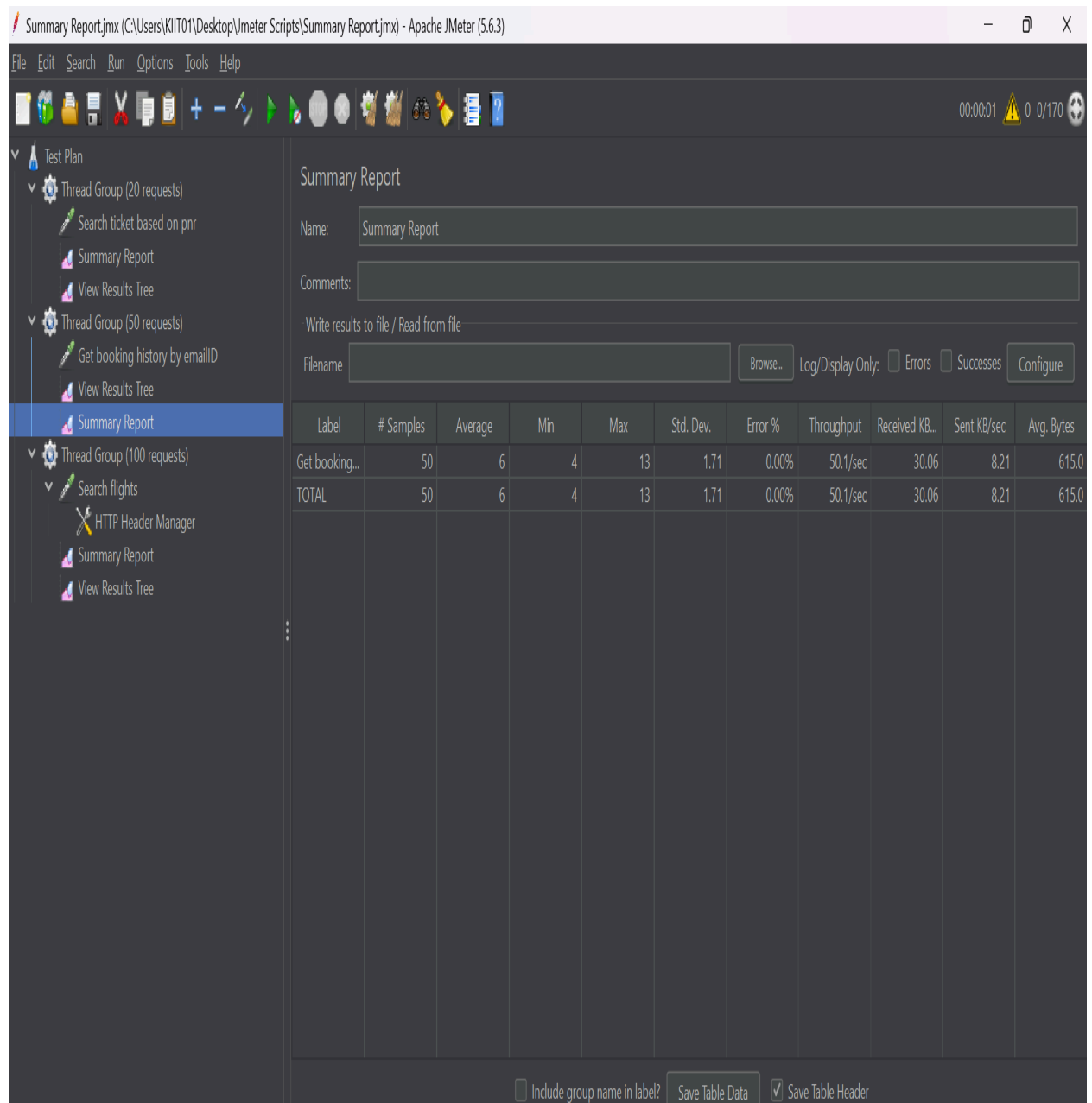
Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

Detail Add Add from Clipboard Delete Up Down





### Case 3: 100 requests

Summary Report.jmx (C:\Users\KIITO1\Desktop\Jmeter Scripts\Summary Report.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:01 0 0/170

Test Plan

- Thread Group (20 requests)
  - Search ticket based on pnr
  - Summary Report
  - View Results Tree
- Thread Group (50 requests)
  - Get booking history by emailID
  - View Results Tree
  - Summary Report
- Thread Group (100 requests)**
  - Search flights
  - HTTP Header Manager
  - Summary Report
  - View Results Tree

Thread Group

Name: Thread Group (100 requests)

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users): 100

Ramp-up period (seconds): 1

Loop Count: ☐ Infinite 1

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Summary Report.jmx (C:\Users\KIIT01\Desktop\Jmeter Scripts\Summary Report.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:01 0 0/170

Test Plan

- Thread Group (20 requests)
  - Search ticket based on pnr
  - Summary Report
  - View Results Tree
- Thread Group (50 requests)
  - Get booking history by emailID
  - View Results Tree
  - Summary Report
- Thread Group (100 requests)
  - Search flights**
  - HTTP Header Manager
  - Summary Report
  - View Results Tree

### HTTP Request

Name: Search flights

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: localhost Port Number: 8081

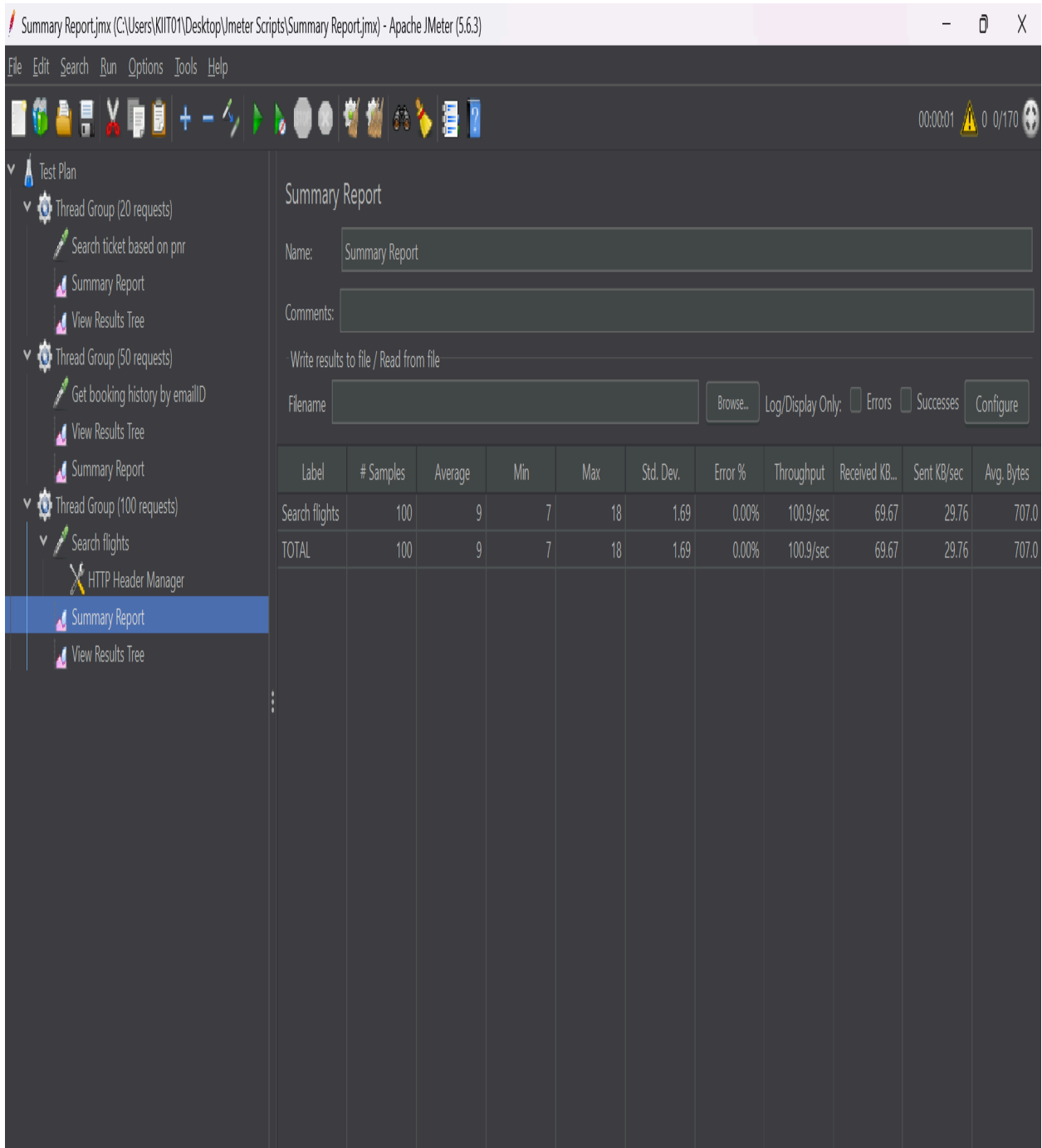
HTTP Request

POST Path: /api/flight/search Content encoding:

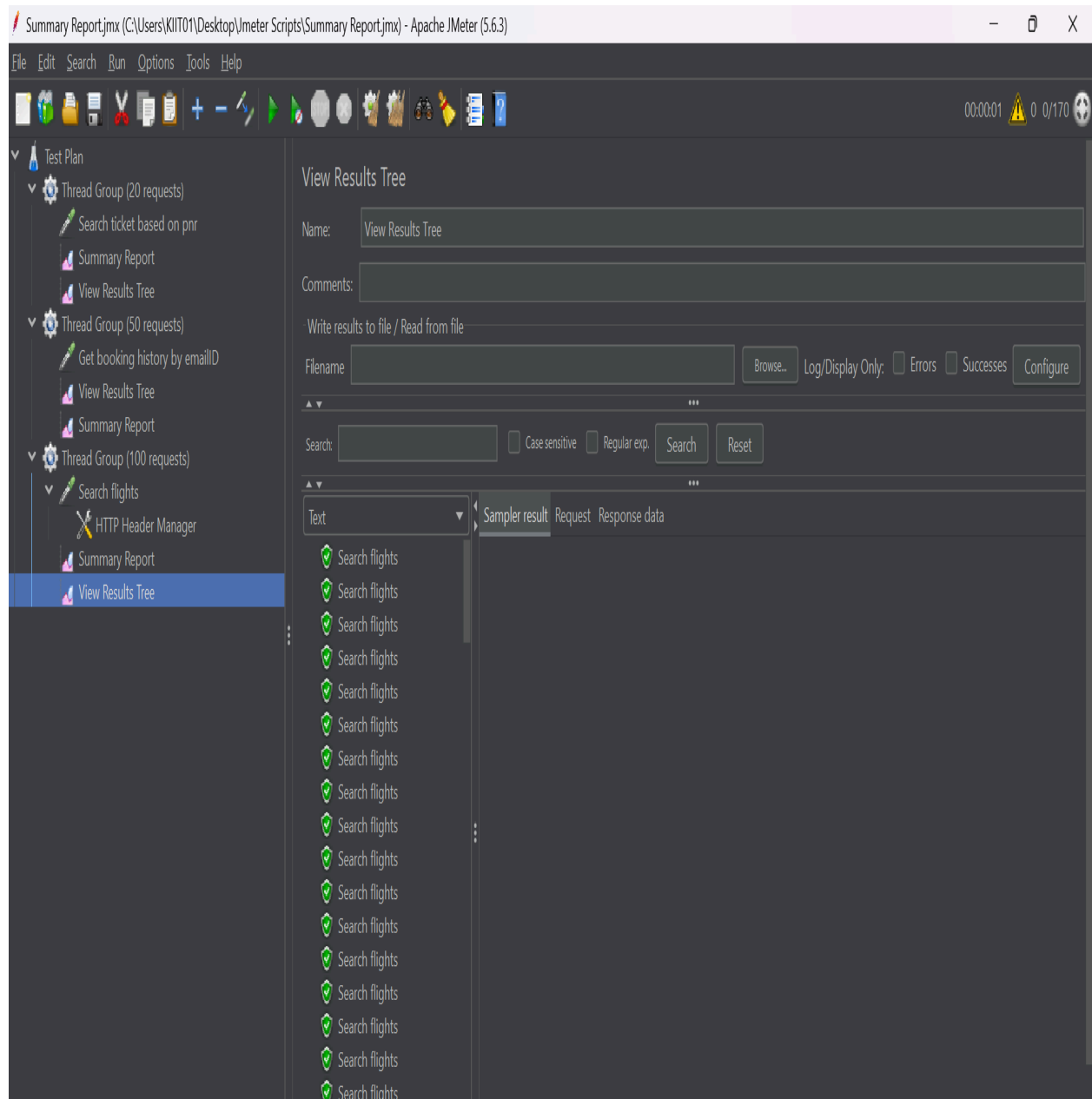
☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

```
1 {
2   "fromLocation": "Delhi",
3   "toLocation": "Mumbai",
4   "travelDate": "2025-11-26",
5   "tripType": "ONE_WAY"
6 }
```







# POSTMAN API ENDPOINT TESTING

## 1. POST:

<http://localhost:8081/api/flight/airline/inventory>

The screenshot displays the Postman interface for a POST request. The top bar shows the collection path 'My Collection / FlightAppWebflux / Add flights to inventory' and a 'Save' button. The request is configured with the method 'POST' and the URL 'http://localhost:8081/api/flight/airline/inventory'. A 'Send' button is visible on the right. The 'Body' tab is selected, showing a raw JSON payload with flight details: airline name, logo, locations, departure/arrival times, price, seats, and flight number. Below the request, the response section shows a '201 Created' status with a response time of 721 ms and a body size of 157 B. The response body is a JSON object containing a success message and a unique ID. The interface includes various tabs for request configuration (Params, Headers, Body, Scripts, Settings) and response viewing (Body, Cookies, Headers, Test Results, Preview, Visualize).

```
1 {
2   "airlineName": "SpiceJet",
3   "airlineLogo": "https://example.com/spicejet.png",
4   "fromLocation": "Bangalore",
5   "toLocation": "Chennai",
6   "departureTime": "2025-11-26T08:00:00",
7   "arrivalTime": "2025-11-26T09:00:00",
8   "price": 3000.0,
9   "availableSeats": 150,
10  "totalSeats": 150,
11  "flightNumber": "SG302"
12 }
```

**201 Created** • 721 ms • 157 B • Save Response

```
1 {
2   "message": "Flight inventory added successfully",
3   "id": "6924ed46234fb520a88ccdc4"
4 }
```

## 2. POST:

<http://localhost:8081/api/flight/search>

The screenshot displays the Postman API client interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. A search bar labeled 'Search Postman' and a keyboard shortcut 'Ctrl K' are present. On the right, there are buttons for 'Invite', 'Upgrade', and a minus sign. The main workspace is titled 'SONAL KUMARI\_3114's Workspace' and contains a 'New' button and an 'Import' button. Below this, a breadcrumb trail shows 'My Collection / FlightAppWebflux / Search flights'. The left sidebar is divided into 'Collections', 'Environments', 'History', and 'Flows'. Under 'Collections', 'My Collection' is expanded, showing a list of endpoints: 'FlightApp', 'SpringBootWebfluxExample', 'MongoSpring', 'SpringMongodbReactive', and 'FlightAppWebflux'. The 'FlightAppWebflux' collection is expanded, showing several endpoints: 'POST Add flights to inventory', 'POST Search flights' (which is selected), 'POST Book tickets', 'GET Get ticket by PNR', 'GET Get Booking history by email', and 'DEL Cancel Ticket'. The main panel shows the configuration for the selected 'POST Search flights' endpoint. The URL is 'http://localhost:8081/api/flight/search'. The 'Body' tab is selected, and the 'raw' radio button is chosen. The body content is a JSON object: 

```
{  "fromLocation": "Delhi",  "toLocation": "Mumbai",  "travelDate": "2025-11-26",  "tripType": "ONE_WAY"}
```

HomeWorkspacesAPI Network

Search PostmanCtrl K

Invite

Upgrade

SONAL KUMARI\_3114's WorkspaceNewImport

Collections

My Collection

FlightApp

SpringBootWebfluxExample

MongoSpring

SpringMongodbReactive

FlightAppWebflux

POST Add flights to inventory

POST Search flights

POST Book tickets

GET Get ticket by PNR

GET Get Booking history by email

DEL Cancel Ticket

DEL Delet

Flight

POST Sear

POST Boo

GET Get ti

GET Get B

POST Add

My Collection / FlightAppWebflux / Search flights

Save

Share

POST

http://localhost:8081/api/flight/search

Send

Docs

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Body

Cookies

Headers (2)

Test Results

200 OK • 131 ms • 683 B • Save Response

JSON

Preview

Visualize

```
1 [
2   {
3     "id": "692411b9702bc266c6879fde",
4     "airlineName": "Air India",
5     "airlineLogo": "https://example.com/airindia.png",
6     "fromLocation": "Delhi",
7     "toLocation": "Mumbai",
8     "departureTime": "2025-11-26T10:00:00",
9     "arrivalTime": "2025-11-26T12:30:00",
10    "price": 5500.0,
11    "totalSeats": 180,
12    "availableSeats": 180,
13    "flightNumber": "AI101"
14  },
15  {
16    "id": "6924137c702bc266c6879fdf",
17    "airlineName": "IndiGo",
18    "airlineLogo": "https://example.com/indigo.png",
19    "fromLocation": "Delhi",
```

### 3. POST:

<http://localhost:8081/api/flight/booking/692413c1702bc266c6879fe0>

The screenshot shows the Postman interface with the following details:

- Collection:** My Collection / FlightAppWebflux / Book tickets
- Method:** POST
- URL:** `http://localhost:8081/api/flight/booking/692413c1702bc266c6879fe0`
- Body Type:** raw (selected), JSON
- Body Content:**

```
1 {
2   "userName": "Simran Sharma",
3   "email": "simran.sharma@example.com",
4   "passengers": [
5     {
6       "name": "Simran Sharma",
7       "gender": "MALE",
8       "age": 28,
9       "seatNumber": "A1",
10      "mealPreference": "VEG"
11    },
12    {
13      "name": "Priya Sharma",
14      "gender": "FEMALE",
15      "age": 26,
16      "seatNumber": "A2",
17      "mealPreference": "VEG"
18    }
19  ]
20 }
```

The screenshot shows the Postman interface with the following details:

- Collection:** My Collection / FlightAppWebflux / Book tickets
- Method:** POST
- URL:** `http://localhost:8081/api/flight/booking/692413c1702bc266c6879fe0`
- Status:** 201 Created
- Response Type:** JSON
- Response Content:**

```
1 {
2   "success": true,
3   "message": "Ticket booked successfully",
4   "data": {
5     "pnr": "FLT25112588EB65",
6     "flightId": "692413c1702bc266c6879fe0",
7     "userName": "Simran Sharma",
8     "email": "simran.sharma@example.com",
9     "passengers": [
10      {
11        "name": "Simran Sharma",
12        "gender": "MALE",
13        "age": 28,
14        "seatNumber": "A1",
15        "mealPreference": "VEG"
16      },
17      {
18        "name": "Priya Sharma",
19        "gender": "FEMALE",
20        "age": 26,
21        "seatNumber": "A2",
22        "mealPreference": "VEG"
23      }
24    ]
25  }
26 }
```

#### 4. GET: <http://localhost:8080/api/flight/booking/ticket/FLT25112588EB65>

The screenshot displays a REST client interface with a sidebar on the left containing a collection of endpoints under 'My Collection'. The main panel shows a GET request to `http://localhost:8081/api/flight/ticket/FLT25112588EB65` which has been successfully executed, returning a 200 OK status. The response body is shown in JSON format, detailing flight information and passenger data.

**Left Sidebar (Collections):**

- My Collection
  - FlightApp
  - SpringBootWebfluxExample
  - MongoSpring
  - SpringMongoDbReactive
  - FlightAppWebflux
    - POST Add flights to inventory
    - POST Search flights
    - POST Book tickets
    - GET Get ticket by PNR**
    - GET Get Booking history by email
    - DEL Cancel Ticket

**Main Panel (Request/Response):**

**Request:** GET `http://localhost:8081/api/flight/ticket/FLT25112588EB65`

**Response:** 200 OK • 363 ms • 697 B

**Body (JSON):**

```
{
  "pnr": "FLT25112588EB65",
  "flightId": "692413c1702bc266c6879fe0",
  "userName": "Simran Sharma",
  "email": "simran.sharma@example.com",
  "passengers": [
    {
      "name": "Simran Sharma",
      "gender": "MALE",
      "age": 28,
      "seatNumber": "A1",
      "mealPreference": "VEG"
    },
    {
      "name": "Priya Sharma",
      "gender": "FEMALE",
      "age": 26,
      "seatNumber": "A2",
      "mealPreference": "VEG"
    }
  ]
}
```

## 5. GET:

<http://localhost:8080/api/flight/booking/history/>

The screenshot shows a REST client interface with a sidebar on the left containing a project tree. The main area displays a GET request to `http://localhost:8081/api/flight/booking/history/simran.sharma@example.com`. The response is a JSON object with a status of 200 OK, a response time of 92 ms, and a size of 598 B. The JSON body contains flight details for two passengers: Simran Sharma and Priya Sharma.

**Request:**

```
GET http://localhost:8081/api/flight/booking/history/simran.sharma@example.com
```

**Response:**

```
{
  "pnr": "FLT25112588EB65",
  "flightId": "692413c1702bc266c6879fe0",
  "userName": "Simran Sharma",
  "email": "simran.sharma@example.com",
  "passengers": [
    {
      "name": "Simran Sharma",
      "gender": "MALE",
      "age": 28,
      "seatNumber": "A1",
      "mealPreference": "VEG"
    },
    {
      "name": "Priya Sharma",
      "gender": "FEMALE",
      "age": 26,
      "seatNumber": "A2",
      "mealPreference": "VEG"
    }
  ]
}
```

## 6. DELETE: <http://localhost:8080/api/flight/booking/cancel/FLT25112588EB65>

The screenshot displays a REST client interface with the following details:

- Method:** DELETE
- URL:** <http://localhost:8080/api/flight/booking/cancel/FLT25112588EB65>
- Send Button:** A blue button labeled "Send".
- Navigation Tabs:** Docs, Params, Authorization, Headers (6), Body, Scripts, Settings. The "Params" tab is currently selected.
- Query Params Table:**

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		
- Response Status:** 200 OK, 99 ms, 185 B. A "Save Response" button is visible.
- Body Tab:** Selected, showing the response in JSON format.

```
1 {  
2   "success": true,  
3   "message": "Booking cancelled successfully",
```