

Get all airlines

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8765/flight-service/api/flight/airline/get
- Status:** 200 OK
- Time:** 762 ms
- Body Size:** 572 B
- Headers:** (9)
- Body:** JSON (Preview and Visualize tabs available)
- Responses:** 200 OK (762 ms, 572 B, Save Response, etc.)

```
1 [  
2 {  
3   "id": "AI001",  
4   "name": "Air India",  
5   "logoUrl": "https://upload.wikimedia.org/airindia/logo.png",  
6   "flightIds": null  
7 },  
8 {  
9   "id": "IN002",  
10  "name": "IndiGo",  
11  "logoUrl": "https://upload.wikimedia.org/indigo/logo.png",  
12  "flightIds": null  
13 }]
```

Get airline by ID

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8765/flight-service/api/flight/airline//get/AI0 ...
- Status:** 200 OK
- Time:** 82 ms
- Body Size:** 225 B
- Headers:** (7)
- Body:** JSON (Preview and Visualize tabs available)
- Responses:** 200 OK (82 ms, 225 B, Save Response, etc.)

```
1 {  
2   "id": "AI001",  
3   "name": "Air India",  
4   "logoUrl": "https://upload.wikimedia.org/airindia/logo.png",  
5   "flightIds": null  
6 }
```

Get if airline by ID not found

The screenshot shows a Postman request for a GET operation at `http://localhost:8765/flight-service/api/flight/airline/get/A10C...`. The response status is 404 Not Found, with a duration of 36 ms and a body size of 82 B. The response body is a single character '1'.

Add flight to inventory

The screenshot shows a Postman request for a POST operation at `http://localhost:8765/flight-service/api/flight/add`. The request body is a JSON object representing a flight with various fields like fromPlace, toPlace, departureTime, arrivalTime, availableSeats, price, oneWay, roundTrip, and airlineId. The response status is 201 Created, with a duration of 199 ms and a body size of 80 B. The response body is a single character '1'.

```
1 {  
2   "fromPlace": "Delhi",  
3   "toPlace": "Mumbai",  
4   "departureTime": "2025-12-10T07:30:00",  
5   "arrivalTime": "2025-12-10T10:00:00",  
6   "availableSeats": 120,  
7   "price": {  
8     "oneWay": 3200,  
9     "roundTrip": 6000  
10    },  
11  "airlineId": "SJ003"  
12 }
```

Add flight to inventory where airline is absent

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** <http://localhost:8765/flight-service/api/flight/add>
- Body:** JSON (Raw view)
- Request Body Content:**

```
1 {
2     "fromPlace": "Delhi",
3     "toPlace": "Mumbai",
4     "departureTime": "2025-12-10T07:30:00",
5     "arrivalTime": "2025-12-10T10:00:00",
6     "availableSeats": 120,
7     "price": {
8         "oneWay": 3200,
9         "roundTrip": 6000
10    },
11    "airlineId": "SJ0003"
12 }
```

- Response Status:** 400 Bad Request
- Response Time:** 30 ms
- Response Size:** 84 B
- Save Response:** Available

Get all flights

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:8765/flight-service/api/flight/get/flights>
- Body:** JSON (Raw view)
- Request Body Content:**

```
{}
```

- Response Status:** 200 OK
- Response Time:** 47 ms
- Response Size:** 583 B
- Save Response:** Available
- Response Preview:** Available
- Response Visualize:** Available

The response body is displayed as:

```
1 [
2     {
3         "id": "692b2874e0335bee85bc7dd9",
4         "fromPlace": "Delhi",
5         "toPlace": "Mumbai",
6         "arrivalTime": "2025-12-10T10:00:00",
7         "departureTime": "2025-12-10T07:30:00",
8         "availableSeats": 120,
9         "price": {
10             "oneWay": 3200.0,
11             "roundTrip": 6000.0
12         },
13         "airlineId": "SJ003"
14     },
```

Get by flightId

HTTP Get Request

GET http://localhost:8765/flight-service/api/flight/get/692b2874e0335bee85bc7dd9

Send

Params

Query Params

	Key	Value	Description	...	Bulk Edit
Body	{ } JSON	Preview Visualize	200 OK	24 ms	348 B

```
1 {
2   "id": "692b2874e0335bee85bc7dd9",
3   "fromPlace": "Delhi",
4   "toPlace": "Mumbai",
5   "arrivalTime": "2025-12-10T10:00:00",
6   "departureTime": "2025-12-10T07:30:00",
7   "availableSeats": 120,
8   "price": {
9     "oneWay": 3200.0,
10    "roundTrip": 6000.0
11  },
12  "airlineId": "SJ003"
13 }
```

Get by flightId if not found

flight microservice / Flight by id not found

HTTP Get Request

GET http://localhost:8765/flight-service/api/flight/get/602b2874e0335bee85bc7dd9

Send

Params

Body

404 Not Found

Raw Preview Debug with AI

Save Response

1

Get flights by searchRequest

The screenshot shows a Postman request for "By searchrequest". The URL is `http://localhost:8765/flight-service/api/flight/search`. The request method is POST. The Body tab shows a JSON payload:

```
1 {
2   "fromPlace": "Delhi",
3   "toPlace": "Mumbai",
4   "date": "2025-12-10"
5 }
```

The response status is 200 OK, with a response time of 51 ms and a response size of 583 B. The response body is a JSON array of flight details:

```
1 [
2   {
3     "id": "692b2874e0335bee85bc7dd9",
4     "fromPlace": "Delhi",
5     "toPlace": "Mumbai",
6     "arrivalTime": "2025-12-10T10:00:00",
7     "departureTime": "2025-12-10T07:30:00",
8     "availableSeats": 120,
9     "price": {
10       "oneWay": 3200.0,
11       "roundTrip": 6000.0
12     },
13     "airlineId": "SJ003"
14   }
]
```

Get flights by searchRequest if not found

The screenshot shows a Postman request for "By searchrequest not found". The URL is `http://localhost:8765/flight-service/api/flight/search`. The request method is POST. The Body tab shows a JSON payload:

```
1 {
2   "fromPlace": "Hyderabad",
3   "toPlace": "Mumbai",
4   "date": "2025-12-10"
5 }
```

The response status is 204 No Content, with a response time of 25 ms and a response size of 64 B.

Adding Booking

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** <http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62b>
- Body Content:**

```
1 {
2   "email": "pranitha@example.com",
3   "seatCount": 2,
4   "tripType": "ROUND_TRIP",
5   "mealPreference": "VEG",
6   "passengerIds": [
7     "692bd52101c23eb0f615c421"
8   ],
9   "seatNumbers": [
10     "3A"
11   ]
12 }
```
- Response Status:** 201 Created
- Response Time:** 574 ms
- Response Size:** 80 B
- Actions:** Save Response, Copy, Find, Delete, Refresh

Add Booking when flight is not present

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** <http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62a>
- Body Content:**

```
1 {
2   "email": "pranitha@example.com",
3   "seatCount": 2,
4   "tripType": "ROUND_TRIP",
5   "mealPreference": "VEG",
6   "passengerIds": [
7     "692bd52101c23eb0f615c421"
8   ],
9   "seatNumbers": [
10     "3A"
11   ]
12 }
```
- Response Status:** 404 Not Found
- Response Time:** 72 ms
- Response Size:** 173 B
- Actions:** Save Response, Copy, Find, Delete, Refresh

The response body contains the message: "Flight not found with id: 692bb90943301a2c8088f62a".

When the seats are already booked

HTTP flight microservice / When seat not available

POST http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62b

Body (9)

none form-data x-www-form-urlencoded raw JSON

```
1 {
2     "email": "pranitha@example.com",
3     "seatCount": 2,
4     "tripType": "ROUND_TRIP",
5     "mealPreference": "VEG",
6     "passengerIds": [
7         "692bd52101c23eb0f615c421"
8     ],
9     "seatNumbers": [
10         "3A"
11     ]
12 }
```

400 Bad Request 42 ms 149 B Save Response

Body Cookies Headers (3) Test Results Raw Preview Debug with AI

1 Seat 3A is not available

Get Using PNR

HTTP flight microservice / Get using pnr

GET http://localhost:8765/booking-service/api/ticket/692bb90943301a2c8088f62b-251130110858

Body (3)

{} JSON

```
1 {
2     "id": "692bd8730339bf2654770c6c",
3     "pnz": "692bb90943301a2c8088f62b-251130110858",
4     "email": "pranitha@example.com",
5     "seatCount": 2,
6     "tripType": "ROUND_TRIP",
7     "mealPreference": "VEG",
8     "flightId": "692bb90943301a2c8088f62b",
9     "passengerIds": [
10         "692bd52101c23eb0f615c421"
11     ],
12     "seatNumbers": [
13         "3A"
14     ],
15     "totalAmount": 3200.0
16 }
```

200 OK 78 ms 410 B Save Response

Body Cookies Headers (3) Test Results Raw Preview Visualize

Get by PNR if not found

HTTP flight microservice / When pnr not found

GET http://localhost:8765/booking-service/api/ticket/692bb90943301a2c8088f62b-251130110859

Params (7)

Query Params

Key	Value	Description	Bulk Edit
Body	Cookies	Headers (2)	Test Results

404 Not Found 13 ms 82 B Save Response

Body Cookies Headers (2) Test Results Raw Preview Debug with AI

Get booking by Email

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** http://localhost:8765/booking-service/api//history/pranitha@example.com
- Body:** JSON response (expanded) showing a booking object with fields like id, pnr, email, seatCount, tripType, mealPreference, flightId, passengerIds, seatNumbers, and totalAmount.
- Status:** 200 OK
- Time:** 25 ms
- Size:** 454 B

When booking with email is not available

The screenshot shows a Postman interface with the following details:

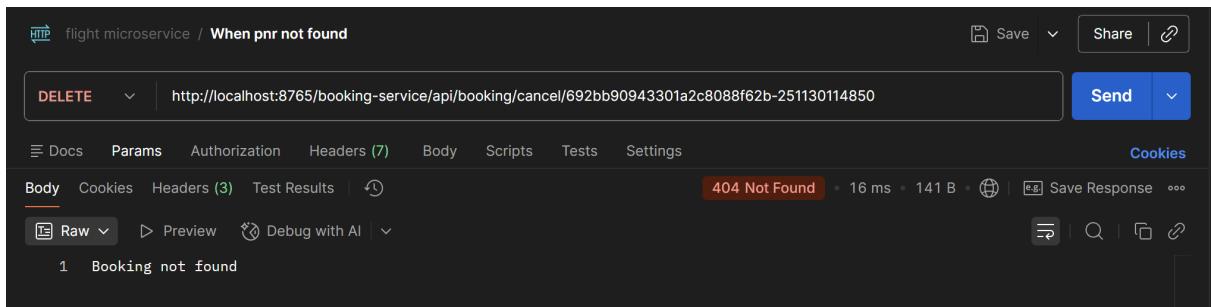
- Method:** GET
- URL:** http://localhost:8765/booking-service/api//history/pranitha1a@example.com
- Status:** 404 Not Found
- Time:** 15 ms
- Size:** 82 B

Delete by PNR

The screenshot shows a Postman interface with the following details:

- Method:** DELETE
- URL:** http://localhost:8765/booking-service/api/booking/cancel/692bb90943301a2c8088f62b-251130114857
- Body:** Raw response (expanded) showing a message: "Booking with PNR 692bb90943301a2c8088f62b-251130114857 deleted successfully. Seats released."
- Status:** 200 OK
- Time:** 203 ms
- Size:** 208 B

Delete if PNR not found



HTTP flight microservice / When pnr not found

DELETE http://localhost:8765/booking-service/api/booking/cancel/692bb90943301a2c8088f62b-251130114850

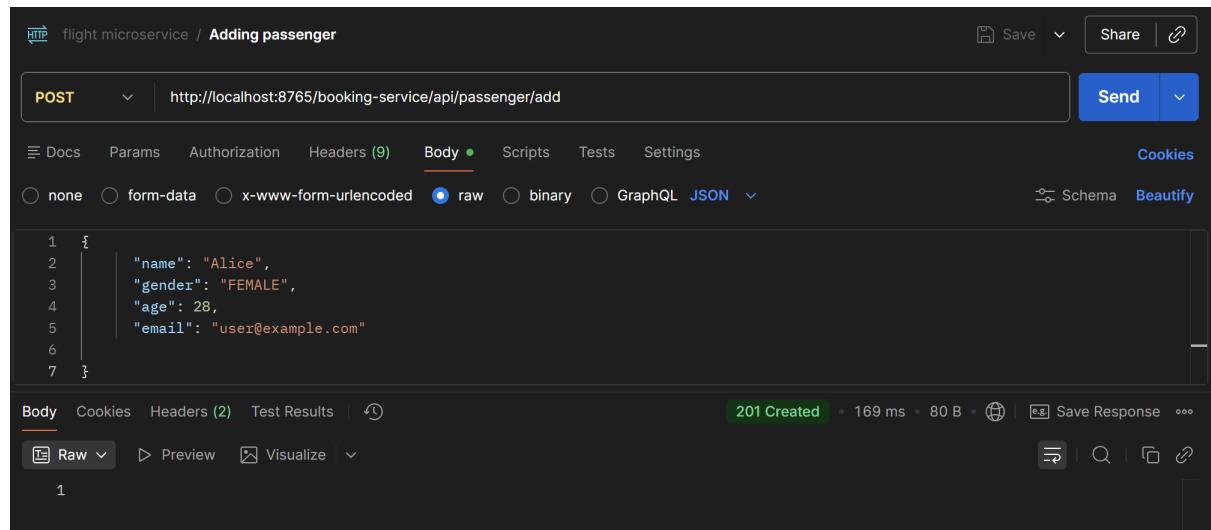
Send

404 Not Found 16 ms 141 B Save Response

Body Raw Preview Debug with AI

Booking not found

Adding Passenger



HTTP flight microservice / Adding passenger

POST http://localhost:8765/booking-service/api/passenger/add

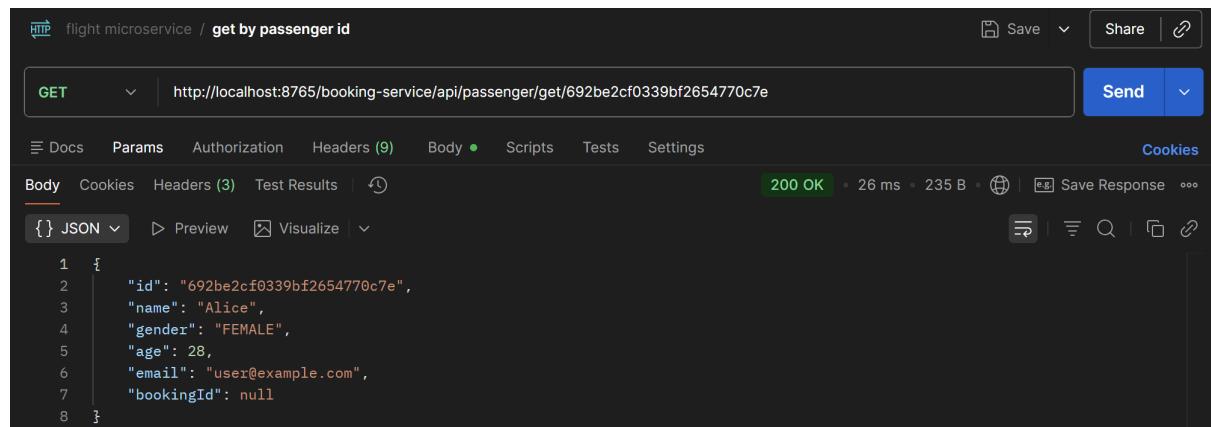
Send

201 Created 169 ms 80 B Save Response

Body Raw Preview Visualize

```
1 {  
2   "name": "Alice",  
3   "gender": "FEMALE",  
4   "age": 28,  
5   "email": "user@example.com"  
6 }  
7
```

Get passenger by id



HTTP flight microservice / get by passenger id

GET http://localhost:8765/booking-service/api/passenger/get/692be2cf0339bf2654770c7e

Send

200 OK 26 ms 235 B Save Response

Body JSON Preview Visualize

```
1 {  
2   "id": "692be2cf0339bf2654770c7e",  
3   "name": "Alice",  
4   "gender": "FEMALE",  
5   "age": 28,  
6   "email": "user@example.com",  
7   "bookingId": null  
8 }
```

Get passenger by id if not found

The screenshot shows the Postman interface with a failed API call. The URL is `http://localhost:8765/booking-service/api/passenger/get/692be2cf0339bf2654770c7e`. The response status is **404 Not Found**, with a response time of 13 ms and a body size of 82 B. The raw response content is empty (1 character).

Get passenger by email

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8765/booking-service/api/passenger/get/email/user@example.com`. The response status is **200 OK**, with a response time of 20 ms and a body size of 279 B. The response body is a JSON object:

```
1 {  
2   "id": "692be2cf0339bf2654770c7e",  
3   "name": "Alice",  
4   "gender": "FEMALE",  
5   "age": 28,  
6   "email": "user@example.com",  
7   "bookingId": null  
8 }
```

Get passenger but email not found

The screenshot shows the Postman interface with a failed API call. The URL is `http://localhost:8765/booking-service/api/passenger/get/email/pranithaa@example.com`. The response status is **404 Not Found**, with a response time of 18 ms and a body size of 82 B. The raw response content is empty (1 character).

Circuit breaker fallback

The screenshot shows the Postman application interface. The URL is `http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62b`. The request method is POST. The body contains a JSON object representing a booking with passenger details. The response status is 503 Service Unavailable, with a message indicating the flight service was unavailable.

```
1 {
2   "email": "pranitha@example.com",
3   "seatCount": 2,
4   "tripType": "ROUND_TRIP",
5   "mealPreference": "VEG",
6   "passengerIds": [
7     "692bd52101c23eb0f615c421"
8 ]
```

503 Service Unavailable • 15 ms • 208 B • ⓘ Save Response ⚙️

1 Flight service unavailable while booking flightId: 692bb90943301a2c8088f62b

Validation failed case

The screenshot shows the Postman application interface. The URL is `http://localhost:8765/booking-service/api/passenger/add`. The request method is POST. The body contains a JSON object with passenger details. The response status is 400 Bad Request, with a detailed error message about a validation failure for the email field.

```
1 {
2   "name": "Pranitha Varakantham",
3   "gender": "FEMALE",
4   "age": 20,
5   "email": "pranitha160805@gmail.com@example.com"
6 }
```

400 Bad Request • 223 ms • 738 B • ⓘ Save Response ⚙️

```
1 {
2   "timestamp": "2025-12-01T14:27:57.873+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "Validation failed for object='passenger'. Error count: 1",
6   "errors": [
7     {
8       "objectName": "passenger",
9       "field": "email",
10      "rejectedValue": "pranitha160805@gmail.com@example.com",
11      "codes": [
12        "Email.passenger.email",
13        "Email.email",
14        "Email.java.lang.String",
15        "Email"
16      ],
17      "arguments": [
18        {
19          "codes": [
20            ...
21          ]
22        }
23      ]
24    }
25  ]
26 }
```

