

Get all airlines

flight microservice / **Get airlines** [Save] [Share]

GET `http://localhost:8765/flight-service/api/flight/airline/get` [Send]

Docs Params Auth Headers (9) Body • Scripts Tests Settings Cookies

Body [200 OK • 762 ms • 572 B • Save Response]

[JSON] [Preview] [Visualize]

```
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   },
14 ]
```

Get airline by ID

flight microservice / **Get airline by id** [Save] [Share]

GET `http://localhost:8765/flight-service/api/flight/airline//get/AI0...` [Send]

Docs Params Auth Headers (7) Body Scripts Tests Settings Cookies

Body [200 OK • 82 ms • 225 B • Save Response]

[JSON] [Preview] [Visualize]

```
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 REST client interface for a "flight microservice". The request is a GET to the endpoint `http://localhost:8765/flight-service/api/flight/airline/get/AI0C...`. The response is a **404 Not Found** status, with a response time of 36 ms and a body size of 82 B. The interface includes tabs for Docs, Params, Auth, Headers (7), Body, Scripts, Tests, and Settings. The Body tab is selected, showing the raw response. A "Send" button is visible next to the request URL.

Add flight to inventory

The screenshot shows a REST client interface for a "flight microservice". The request is a POST to the endpoint `http://localhost:8765/flight-service/api/flight/add`. The request body is a JSON object:

```
{  "fromPlace": "Delhi",  "toPlace": "Mumbai",  "departureTime": "2025-12-10T07:30:00",  "arrivalTime": "2025-12-10T10:00:00",  "availableSeats": 120,  "price": {    "oneWay": 3200,    "roundTrip": 6000  },  "airlineId": "SJ003"}
```

. The response is a **201 Created** status, with a response time of 199 ms and a body size of 80 B. The interface includes tabs for Docs, Params, Auth, Headers (9), Body, Scripts, Tests, and Settings. The Body tab is selected, showing the raw request body. A "Send" button is visible next to the request URL.

Add flight to inventory where airline is absent

flight microservice / **New Request** Save Share

POST ▼ `http://localhost:8765/flight-service/api/flight/add` Send ▼

Docs Params Auth Headers (9) **Body** Scripts Tests Settings Cookies

raw ▼ **JSON** ▼ Schema Beautify

```
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 }
```

Body ▼ 🕒 **400 Bad Request** • 30 ms • 84 B • 🌐 Save Response ⋮

Raw ▼ ▶ Preview 🐞 Debug with AI ▼ ↺ 🔍 📄 🔗

1

Get all flights

flight microservice / **Get all flights** Save Share

GET ▼ `http://localhost:8765/flight-service/api/flight/get/flights` Send ▼

Docs Params Auth Headers (7) **Body** Scripts Tests Settings Cookies

Body ▼ 🕒 **200 OK** • 47 ms • 583 B • 🌐 Save Response ⋮

{} **JSON** ▼ ▶ Preview 🖼 Visualize ▼ ↺ 🔍 📄 🔗

```
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  },
15 ]
```

Get by flightId

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

- Method:** GET
- URL:** `http://localhost:8765/flight-service/api/flight/get/692b2874e...`
- Status:** 200 OK
- Response Body (JSON):**

```
{
  "id": "692b2874e0335bee85bc7dd9",
  "fromPlace": "Delhi",
  "toPlace": "Mumbai",
  "arrivalTime": "2025-12-10T10:00:00",
  "departureTime": "2025-12-10T07:30:00",
  "availableSeats": 120,
  "price": {
    "oneWay": 3200.0,
    "roundTrip": 6000.0
  },
  "airlineId": "SJ003"
}
```

Get by flightId if not found

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

- Method:** GET
- URL:** `http://localhost:8765/flight-service/api/flight/get/602b2874e...`
- Status:** 404 Not Found
- Response Body:** (Empty, as indicated by the 'Raw' view showing only a line number '1')

Get flights by searchRequest

The screenshot shows a REST client interface for a flight microservice. The request is a POST to `http://localhost:8765/flight-service/api/flight/search`. The request body is a JSON object with the following fields:

```
{
  "fromPlace": "Delhi",
  "toPlace": "Mumbai",
  "date": "2025-12-10"
}
```

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

```
[
  {
    "id": "692b2874e0335bee85bc7dd9",
    "fromPlace": "Delhi",
    "toPlace": "Mumbai",
    "arrivalTime": "2025-12-10T10:00:00",
    "departureTime": "2025-12-10T07:30:00",
    "availableSeats": 120,
    "price": {
      "oneWay": 3200.0,
      "roundTrip": 6000.0
    },
    "airlineId": "SJ003"
  }
]
```

Get flights by searchRequest if not found

The screenshot shows the same REST client interface, but the request body is now:

```
{
  "fromPlace": "Hyderabad",
  "toPlace": "Mumbai",
  "date": "2025-12-10"
}
```

The response is a 204 No Content status with a response time of 25 ms and a body size of 64 B. The response body is empty, as indicated by the 'Raw' tab showing a single line with the number 1.

Adding Booking

The screenshot shows a REST client interface for a flight microservice. The request is a POST to `http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62b`. The body is a JSON object with the following structure:

```
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 }
13
```

The response is a 201 Created status with a response time of 574 ms and a body size of 80 B. The response is saved and can be viewed in raw, preview, or visualize format.

Add Booking when flight is not present

The screenshot shows a REST client interface for a flight microservice. The request is a POST to `http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62a`. The body is a JSON object with the following structure:

```
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 }
```

The response is a 404 Not Found status with a response time of 72 ms and a body size of 173 B. The response is saved and can be viewed in raw, preview, or debug with AI format.

1 Flight not found with id: 692bb90943301a2c8088f62a

When the seats are already booked

flight microservice / **When seat not available**

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

Docs Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```
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 }
```

Body Cookies Headers (3) Test Results 400 Bad Request • 42 ms • 149 B Save Response

Raw Preview Debug with AI

```
1 Seat 3A is not available
```

Get Using PNR

flight microservice / **Get using pnr**

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

Docs Params Authorization Headers (7) **Body** Scripts Tests Settings Cookies

Body Cookies Headers (3) Test Results 200 OK • 78 ms • 410 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "id": "692bd8730339bf2654770c6c",
3   "pnr": "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 }
```

Get by PNR if not found

flight microservice / **When pnr not found**

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

Docs **Params** Authorization Headers (7) **Body** Scripts Tests Settings Cookies

Query Params

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (2) Test Results 404 Not Found • 13 ms • 82 B Save Response

Raw Preview Debug with AI

```
1
```

Get booking by Email

flight microservice / Using email

GET

http://localhost:8765/booking-service/api/history/pranitha@example.com

Send

Docs

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

Body

Cookies

Headers (4)

Test Results

200 OK • 25 ms • 454 B • Save Response

JSON

Preview

Visualize

```
1 {
2   "id": "692bd8730339bf2654770c6c",
3   "pnr": "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 }
```

When booking with email is not available

flight microservice / When email not found

GET

http://localhost:8765/booking-service/api/history/pranith1a@example.com

Send

Docs

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

Body

Cookies

Headers (2)

Test Results

404 Not Found • 15 ms • 82 B • Save Response

Raw

Preview

Debug with AI

```
1
```

Delete by PNR

flight microservice / Delete using pnr

DELETE

http://localhost:8765/booking-service/api/booking/cancel/692bb90943301a2c8088f62b-251130114857

Send

Docs

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

Query Params

Key	Value	Description	Bulk Edit
-----	-------	-------------	-----------

Body

Cookies

Headers (3)

Test Results

200 OK • 203 ms • 208 B • Save Response

Raw

Preview

Visualize

```
1 Booking with PNR 692bb90943301a2c8088f62b-251130114857 deleted successfully. Seats released.
```


Delete if PNR not found

The screenshot shows a REST client interface for a "flight microservice". The request is a **DELETE** method to the URL `http://localhost:8765/booking-service/api/booking/cancel/692bb90943301a2c8088f62b-251130114850`. The response status is **404 Not Found**, with a response time of 16 ms and a body size of 141 B. The response body is displayed in "Raw" format as `Booking not found`.

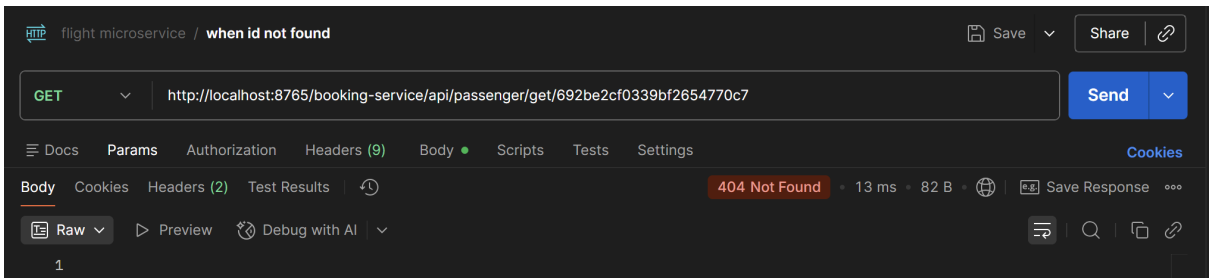
Adding Passenger

The screenshot shows a REST client interface for a "flight microservice". The request is a **POST** method to the URL `http://localhost:8765/booking-service/api/passenger/add`. The request body is in "raw" format, containing a JSON object: `{ "name": "Alice", "gender": "FEMALE", "age": 28, "email": "user@example.com" }`. The response status is **201 Created**, with a response time of 169 ms and a body size of 80 B. The response body is displayed in "Raw" format as `1`.

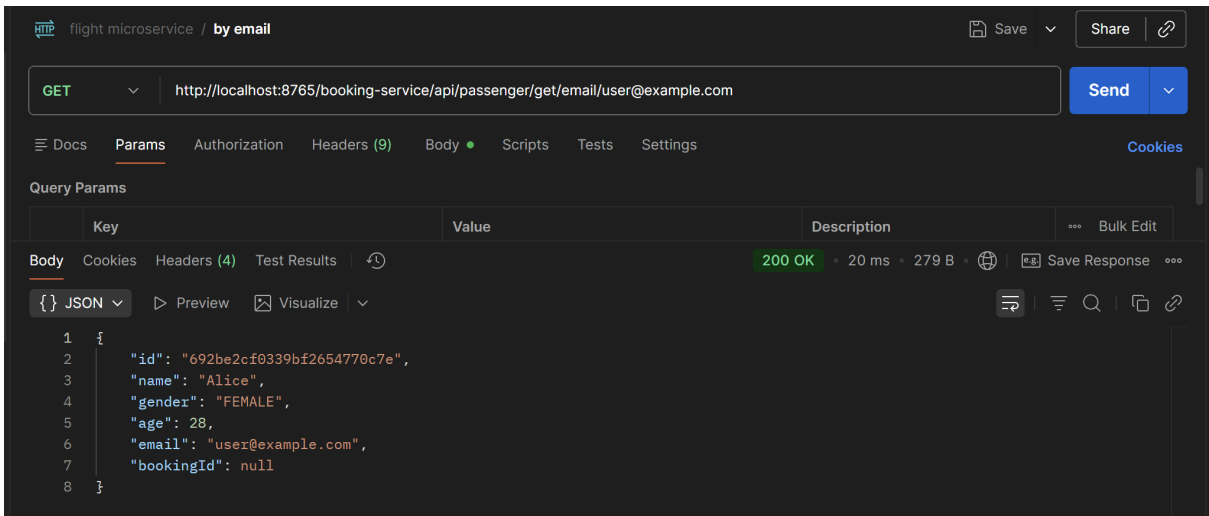
Get passenger by id

The screenshot shows a REST client interface for a "flight microservice". The request is a **GET** method to the URL `http://localhost:8765/booking-service/api/passenger/get/692be2cf0339bf2654770c7e`. The response status is **200 OK**, with a response time of 26 ms and a body size of 235 B. The response body is displayed in "JSON" format, showing a JSON object: `{ "id": "692be2cf0339bf2654770c7e", "name": "Alice", "gender": "FEMALE", "age": 28, "email": "user@example.com", "bookingId": null }`.

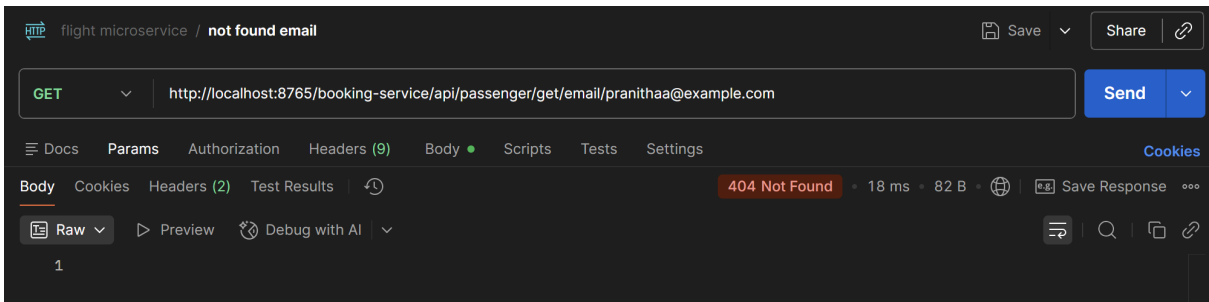
Get passenger by id if not found



Get passenger by email



Get passenger but email not found



Circuit breaker fallback

The screenshot shows a REST client interface for a flight microservice. The request is a POST to `http://localhost:8765/booking-service/api/booking/692bb90943301a2c8088f62b`. The request body is a JSON object with the following fields: `email` (pranitha@example.com), `seatCount` (2), `tripType` (ROUND_TRIP), `mealPreference` (VEG), and `passengerIds` (an array containing "692bd52101c23eb0f615c421"). The response is a 503 Service Unavailable status with a 15 ms duration and 208 B of data. The response body is raw text: "Flight service unavailable while booking flightId: 692bb90943301a2c8088f62b".

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

Body Cookies Headers (3) Test Results 503 Service Unavailable • 15 ms • 208 B Save Response

Raw Preview Debug with AI

1 Flight service unavailable while booking flightId: 692bb90943301a2c8088f62b

Validation failed case

The screenshot shows a REST client interface for a flight microservice. The request is a POST to `http://localhost:8765/booking-service/api/passenger/add`. The request body is a JSON object with the following fields: `name` (Pranitha Varakantham), `gender` (FEMALE), `age` (20), and `email` (pranitha160805@gmail.com@example.com). The response is a 400 Bad Request status with a 223 ms duration and 738 B of data. The response body is a JSON object with the following fields: `timestamp` (2025-12-01T14:27:57.873+00:00), `status` (400), `error` (Bad Request), `message` (Validation failed for object='passenger'. Error count: 1), and `errors` (an array of error objects). The first error object has the following fields: `objectName` (passenger), `field` (email), `rejectedValue` (pranitha160805@gmail.com@example.com), and `codes` (an array of error codes: Email.passenger.email, Email.email, Email.java.lang.String, Email).

```
1 {
2   "name": "Pranitha Varakantham",
3   "gender": "FEMALE",
4   "age": 20,
5   "email": "pranitha160805@gmail.com@example.com"
6 }
7
```

Body Cookies Headers (3) Test Results 400 Bad Request • 223 ms • 738 B Save Response

{ JSON Preview Debug with AI

```
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": [
```

