

Adding Airline

The screenshot shows the Postman interface for a POST request. The request is sent to `http://localhost:8080/api/v1.0/flight/airline/add`. The body is a JSON object: `{ "name": "Indigo", "logoUrl": "https://indigo.com/logo.png" }`. The response is a `201 Created` status with a response time of 28 ms and a body size of 43 B. The response is displayed in the 'Raw' tab.

```
POST http://localhost:8080/api/v1.0/flight/airline/add
```

```
{  
  "name": "Indigo",  
  "logoUrl": "https://indigo.com/logo.png"  
}
```

201 Created • 28 ms • 43 B • Save Response

Raw • Preview • Visualization

Trying to add Airline with same Name

The screenshot shows the Postman interface for a POST request. The request is sent to `http://localhost:8080/api/v1.0/flight/airline/add`. The body is a JSON object: `{ "name": "Indigo", "logoUrl": "https://indigo.com/logo.png" }`. The response is a `409 Conflict` status with a response time of 25 ms and a body size of 44 B. The response is displayed in the 'Raw' tab.

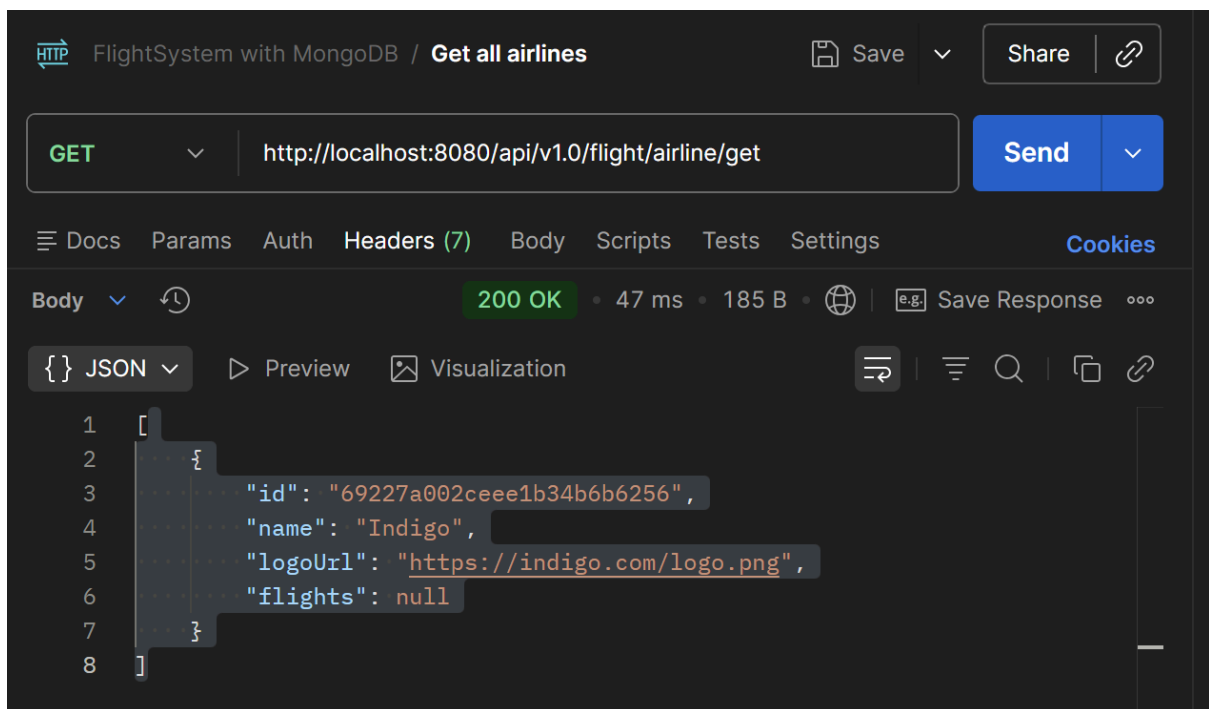
```
POST http://localhost:8080/api/v1.0/flight/airline/add
```

```
{  
  "name": "Indigo",  
  "logoUrl": "https://indigo.com/logo.png"  
}
```

409 Conflict • 25 ms • 44 B • Save Response

Raw • Preview • Visualization

Get all airlines



FlightSystem with MongoDB / **Get all airlines** Save Share

GET http://localhost:8080/api/v1.0/flight/airline/get Send

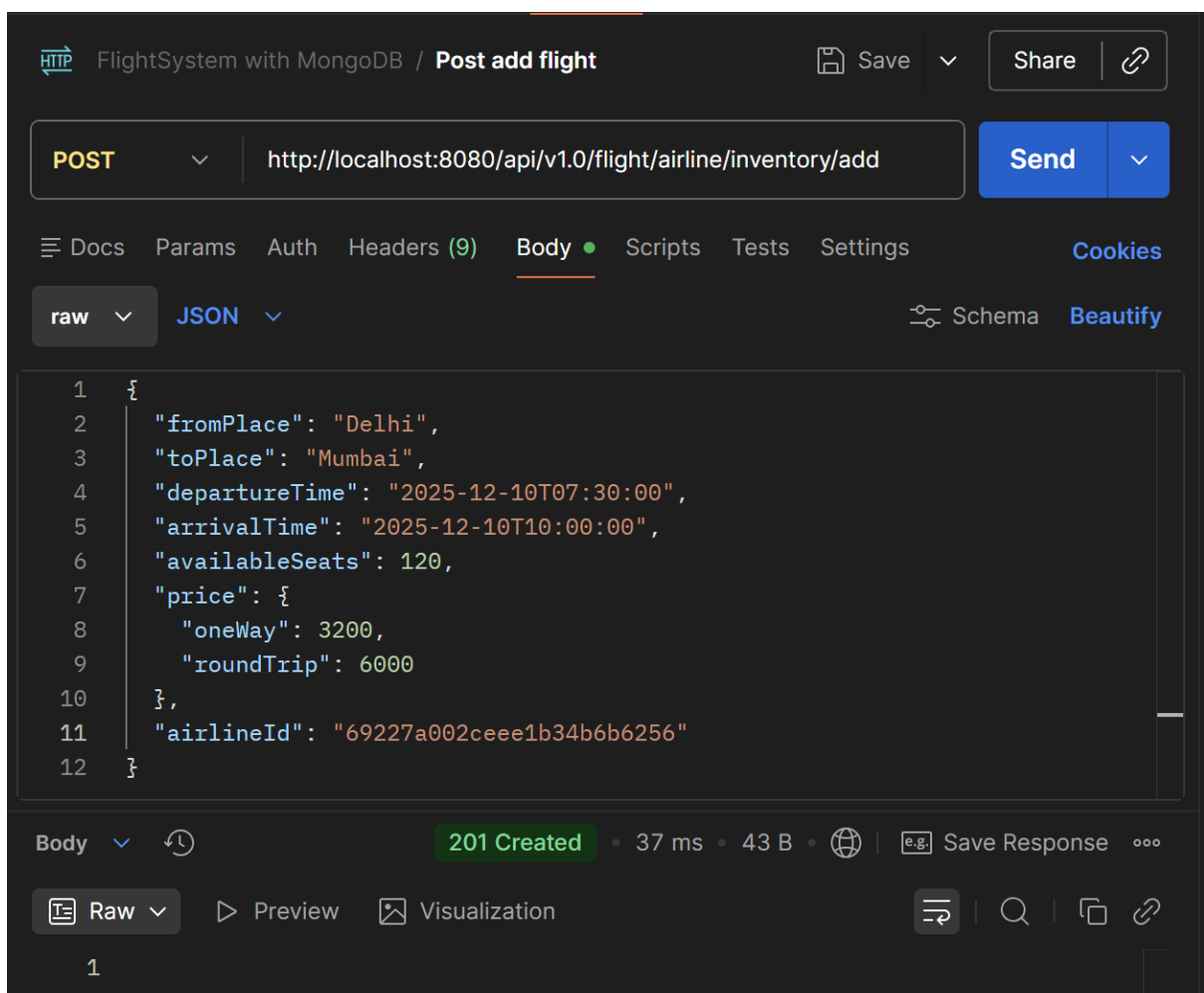
Docs Params Auth Headers (7) Body Scripts Tests Settings Cookies

Body 200 OK • 47 ms • 185 B Save Response

JSON Preview Visualization

```
1 [
2   {
3     "id": "69227a002ceee1b34b6b6256",
4     "name": "Indigo",
5     "logoUrl": "https://indigo.com/logo.png",
6     "flights": null
7   }
8 ]
```

Add flight to inventory



FlightSystem with MongoDB / **Post add flight** Save Share

POST http://localhost:8080/api/v1.0/flight/airline/inventory/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": "69227a002ceee1b34b6b6256"
12 }
```

Body 201 Created • 37 ms • 43 B Save Response

Raw Preview Visualization

1

Add flight to inventory where airline is absent

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

- URL:** `http://localhost:8080/api/v1.0/flight/airline/inventory/add`
- Method:** `POST`
- Body (JSON):**

```
{
  "fromPlace": "Delhi",
  "toPlace": "Mumbai",
  "departureTime": "2025-12-10T07:30:00",
  "arrivalTime": "2025-12-10T10:00:00",
  "availableSeats": 120,
  "price": {
    "oneWay": 3200,
    "roundTrip": 6000
  },
  "airlineId": "69227a002ceee1b34b6b6259"
}
```
- Response:** `400 Bad Request` (11 ms, 47 B)

Get all flights

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

- URL:** `http://localhost:8080/api/v1.0/flight/get/flights`
- Method:** `GET`
- Response:** `200 OK` (20 ms, 348 B)
- Body (JSON):**

```
[
  {
    "id": "6922aa8e2ceee1b34b6b629b",
    "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": "69227a002ceee1b34b6b6256",
    "bookings": null
  }
]
```

Get by flightId

The screenshot shows a REST client interface for a project named "FlightSystem with MongoDB". The active tab is "Get by flightId". The request method is "GET" and the URL is "http://localhost:8080/api/v1.0/flight/get/6922aa8e2ceee1b34...". The response status is "200 OK" with a response time of 19 ms and a body size of 339 B. The response body is displayed in JSON format, showing flight details for a flight from Delhi to Mumbai.

```
{  "id": "6922aa8e2ceee1b34b6b629b",  "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": "69227a002ceee1b34b6b6256",  "bookings": null}
```

Get by flightId if not found

The screenshot shows the same REST client interface, but for a request titled "Get by flightId if not present". The request method is "GET" and the URL is "http://localhost:8080/api/v1.0/flight/get/692285912ceee1a34...". The response status is "404 Not Found" with a response time of 16 ms and a body size of 45 B. The response body is currently 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 project named "FlightSystem with MongoDB". A new POST request is configured to the endpoint `http://localhost:8080/api/v1.0/flight/search`. The request body is a JSON object with the following fields:

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

The response is a 200 OK status with a response time of 51 ms and a body size of 348 B. The response body is a JSON array containing one flight object:

```
1 [
2   {
3     "id": "692285912ceee1b34b6b6274",
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": 0.0,
11      "roundTrip": 0.0
12    },
13     "airlineId": "69227a002ceee1b34b6b6256",
14     "bookings": null,
15     "seats": null
16   }
17 ]
```

Get flights by searchRequest if not found

The screenshot shows the same REST client interface. The request body is updated to use the date "2025-12-11":

```
1 {
2   "fromPlace": "Delhi",
3   "toPlace": "Mumbai",
4   "date": "2025-12-11"
5 }
```

The response is a 204 No Content status with a response time of 16 ms and a body size of 27 B. The response body is empty, as indicated by the "Raw" view showing only a line number "1".

Adding Booking

The screenshot shows a REST client interface for a project named "FlightSystem with MongoDB". The active tab is "Add booking". The request method is "POST" and the URL is "http://localhost:8080/api/v1.0/flight/booking/6922aa8e2ceee...". The request body is a JSON object representing a booking with two passengers, Alice and Bob. The status bar at the bottom indicates a successful response: "201 Created" with a response time of 176 ms and a body size of 43 B.

```
1 {
2   "email": "user@example.com",
3   "seatCount": 2,
4   "tripType": "ROUND_TRIP",
5   "mealPreference": "VEG",
6   "seatNumbers": ["12C", "12D"],
7   "passengers": [
8     {
9       "name": "Alice",
10      "gender": "FEMALE",
11      "age": 28,
12      "email": "user@example.com"
13    },
14    {
15      "name": "Bob",
16      "gender": "MALE",
17      "age": 32,
18      "email": "user1@example.com"
19    }
20  ]
21 }
22 }
23 }
```

Body ▾ ⌛ 201 Created • 176 ms • 43 B • 🌐 ⚙️ Save Response ⋮

Raw ▾ ▶ Preview 🖼 Visualization 🔍 🔍 🔍 🔍

1

Add Booking when flight is not present

The screenshot shows the same REST client interface, but the title is "When flight not found". The request is identical to the previous one. However, the status bar at the bottom indicates a "400 Bad Request" response with a response time of 13 ms and a body size of 47 B.

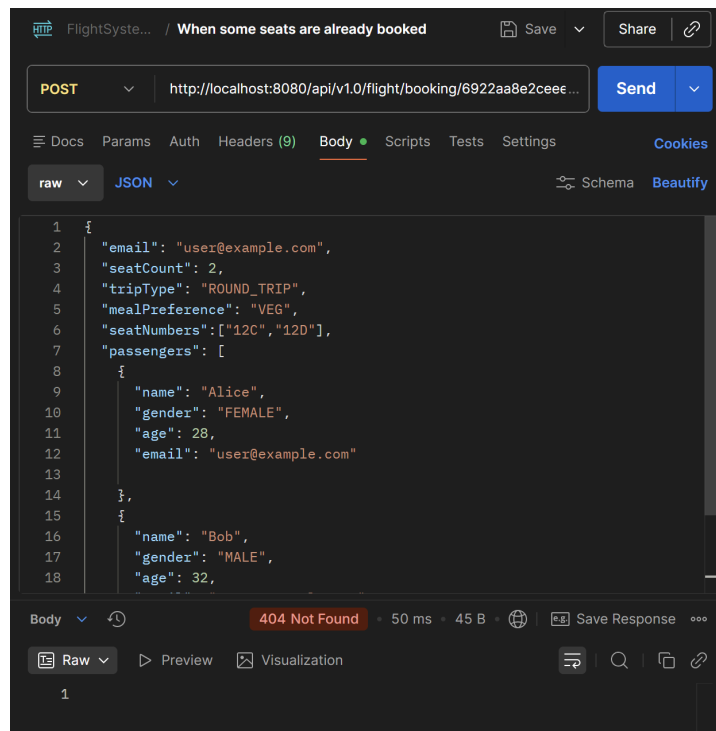
```
1 {
2   "email": "user@example.com",
3   "seatCount": 2,
4   "tripType": "ROUND_TRIP",
5   "mealPreference": "VEG",
6   "seatNumbers": ["12C", "12D"],
7   "passengers": [
8     {
9       "name": "Alice",
10      "gender": "FEMALE",
11      "age": 28,
12      "email": "user@example.com"
13    },
14    {
15      "name": "Bob",
16      "gender": "MALE",
17      "age": 32,
18      "email": "user1@example.com"
19    }
20  ]
21 }
22 }
23 }
24 }
```

Body ▾ ⌛ 400 Bad Request • 13 ms • 47 B • 🌐 ⚙️ Save Response ⋮

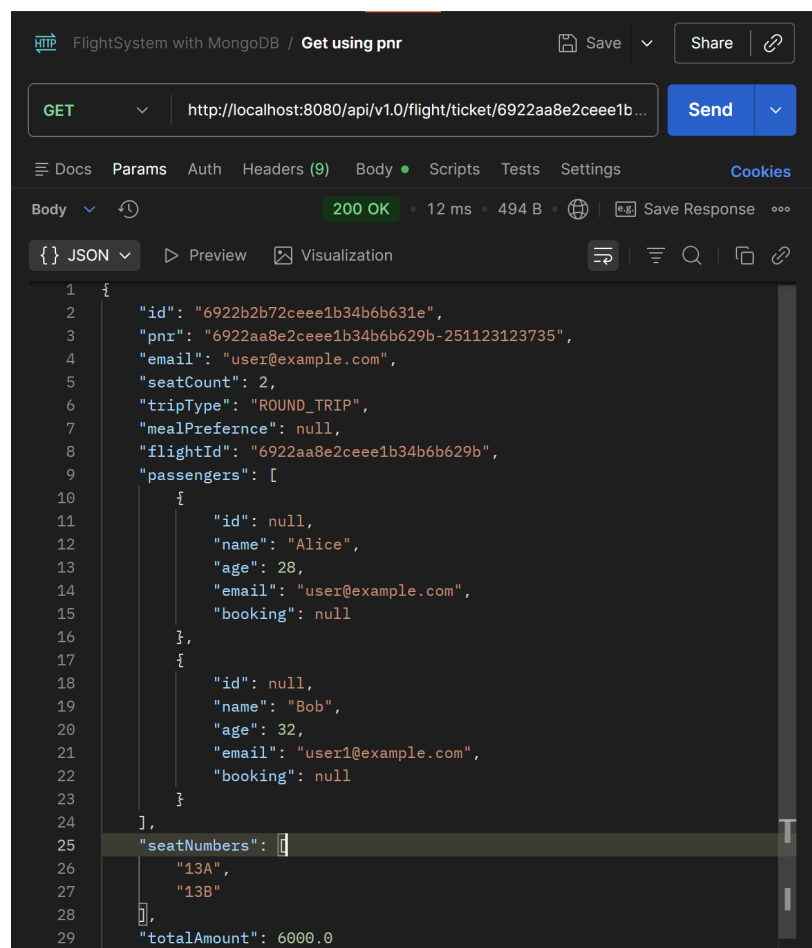
Raw ▾ ▶ Preview 🖼 Visualization 🔍 🔍 🔍 🔍

1

When the seats are already booked



Get Using PNR



Get by PNR if not found

The screenshot shows a REST client interface for a project named "FlightSystem with MongoDB". The active tab is "Get if PNR not found". The request method is "GET" and the URL is "http://localhost:8080/api/v1.0/flight/ticket/6922aa8e2ceee1b...". The response status is "404 Not Found" with a response time of 22 ms and a body size of 45 B. The response body is currently empty, showing only the line number "1".

Get booking by Email

The screenshot shows a REST client interface for a project named "FlightSystem with MongoDB". The active tab is "Get by email". The request method is "GET" and the URL is "http://localhost:8080/api/v1.0/flight/history/user@example.cc...". The response status is "200 OK" with a response time of 28 ms and a body size of 494 B. The response body is displayed in JSON format:

```
1 {
2   "id": "6922b2b72ceee1b34b6b631e",
3   "pnr": "6922aa8e2ceee1b34b6b629b-251123123735",
4   "email": "user@example.com",
5   "seatCount": 2,
6   "tripType": "ROUND_TRIP",
7   "mealPreference": null,
8   "flightId": "6922aa8e2ceee1b34b6b629b",
9   "passengers": [
10    {
11      "id": null,
12      "name": "Alice",
13      "age": 28,
14      "email": "user@example.com",
15      "booking": null
16    },
17    {
18      "id": null,
19      "name": "Bob",
20      "age": 32,
21      "email": "user1@example.com",
22      "booking": null
23    }
24  ],
25   "seatNumbers": [
26     "13A",
27     "13B"
28   ],
29   "totalAmount": 6000.0
```


When booking with email is not available

FlightSystem with MongoDB / **When email not found** [Save] [Share]

GET `http://localhost:8080/api/v1.0/flight/history/user@example1.c...` [Send]

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

Body **404 Not Found** • 9 ms • 45 B • [Save Response]

Raw Preview Visualization

1

Delete by PNR

FlightSystem with MongoDB / **Delete by PNR** [Save] [Share]

DELETE `http://localhost:8080/api/v1.0/flight/booking/cancel/6922aa8...` [Send]

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

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body **204 No Content** • 30 ms • 27 B • [Save Response]

Raw Preview Visualization

1

Delete if PNR not found

FlightSystem with MongoDB / **Delete if PNR not found** [Save] [Share]

DELETE `http://localhost:8080/api/v1.0/flight/booking/cancel/9922aa8...` [Send]

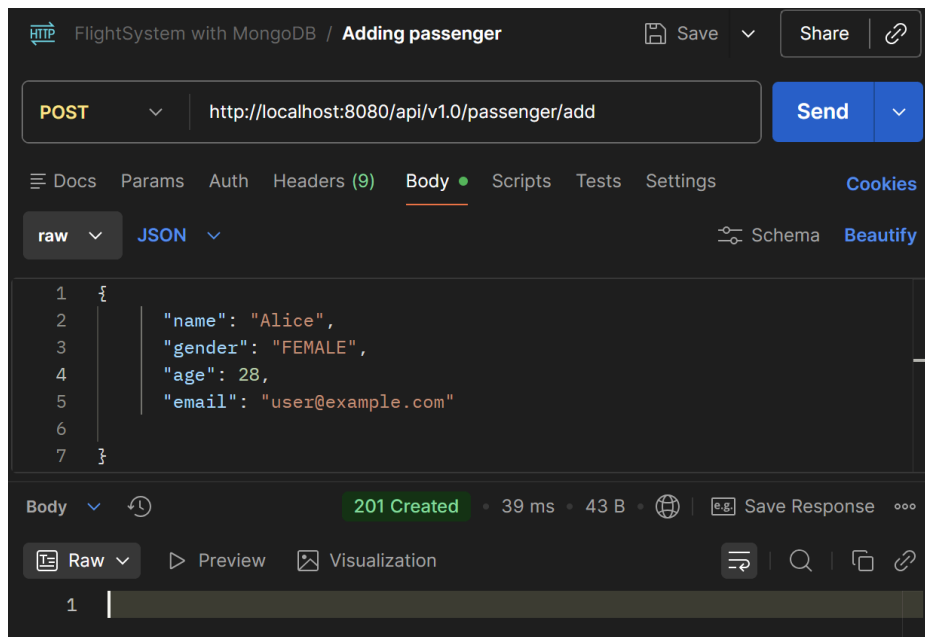
Docs Params Auth Headers (9) Body Scripts Tests Settings Cookies

Body **404 Not Found** • 17 ms • 45 B • [Save Response]

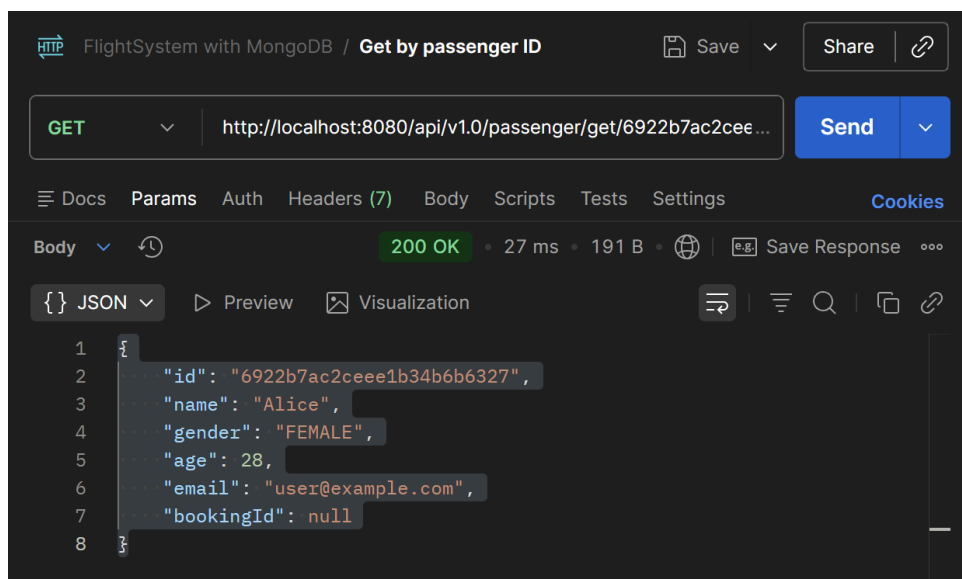
Raw Preview Visualization

1

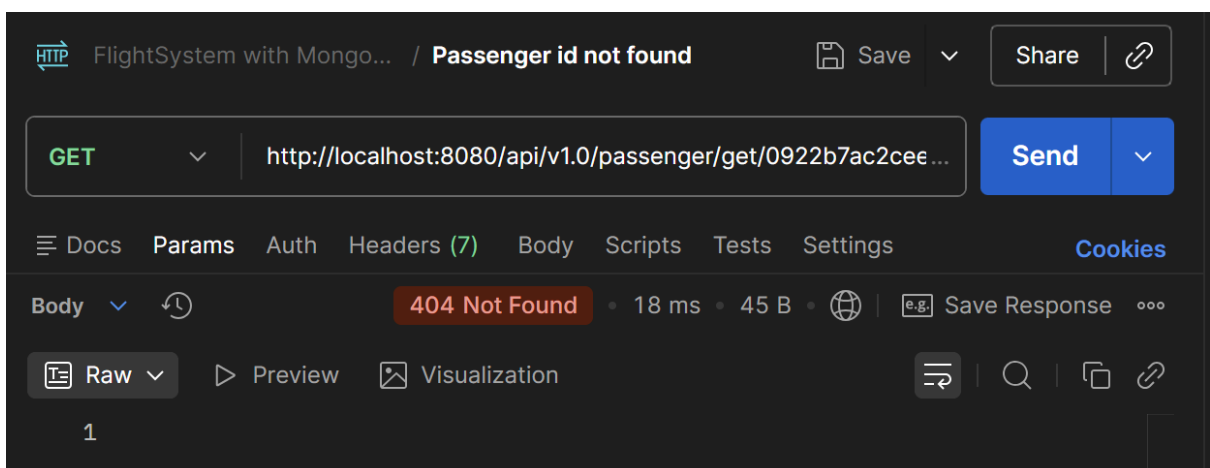
Adding Passenger



Get passenger by id



Get passenger by id if not found



Get passenger by email

The screenshot shows a REST client interface for a project named "FlightSystem with MongoDB". The active tab is "Passenger by email". The request is a GET to `http://localhost:8080/api/v1.0/passenger/get/email/user@ex...`. The response is a 200 OK status with a response time of 12 ms and a body size of 191 B. The response body is displayed in JSON format:

```
1 {
2   "id": "6922b7ac2ceee1b34b6b6327",
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 same REST client interface, but the request is now `http://localhost:8080/api/v1.0/passenger/get/email/user1@e...`. The response is a 404 Not Found status with a response time of 13 ms and a body size of 45 B. The response body is currently empty, as indicated by the "Raw" view showing a single line with a cursor.