

Adding Airline

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/v1.0/flight/airline/add`. The request body is JSON:

```
1 {
2   "name": "Indigo",
3   "logoUrl": "https://indigo.com/logo.png"
4 }
```

The response is **201 Created** with a time of 28 ms and a size of 43 B. The response body is empty (1).

Trying to add Airline with same Name

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/v1.0/flight/airline/add`. The request body is JSON:

```
1 {
2   "name": "Indigo",
3   "logoUrl": "https://indigo.com/logo.png"
4 }
```

The response is **409 Conflict** with a time of 25 ms and a size of 44 B. The response body is empty (1).

Get all airlines

The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/v1.0/flight/airline/get
- Status:** 200 OK
- Body:** JSON response (Pretty-Printed)

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

Add flight to inventory

The screenshot shows the Postman interface with the following details:

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

```
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 }
```
- Status:** 201 Created

Add flight to inventory where airline is absent

The screenshot shows a POST request to `http://localhost:8080/api/v1.0/flight/airline/inventory/add`. The request body is a JSON object representing a flight with an absent airline ID:

```
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": "69227a002ceee1b34b6b6259"  
12 }
```

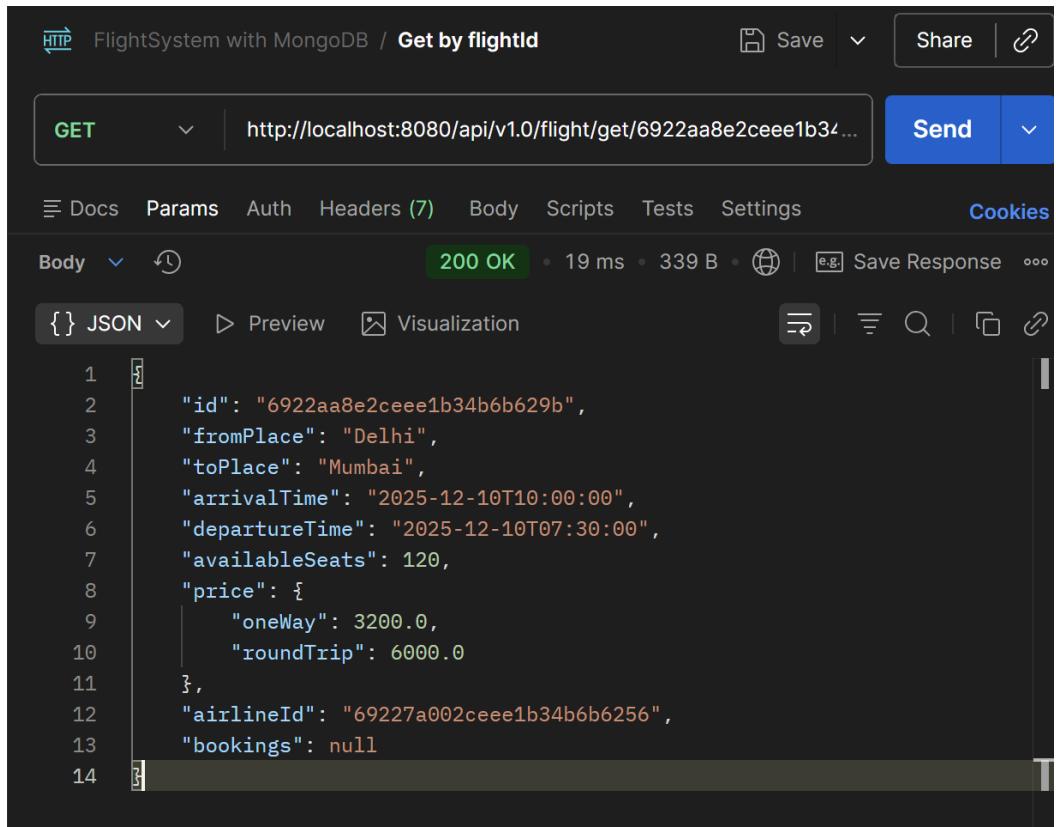
The response is a 400 Bad Request, indicating validation errors.

Get all flights

The screenshot shows a GET request to `http://localhost:8080/api/v1.0/flight/get/flights`. The response is a 200 OK, returning a list of flight documents:

```
1 [  
2   {  
3     "id": "6922aa8e2ceee1b34b6b629b",  
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": "69227a002ceee1b34b6b6256",  
14     "bookings": null  
15   }  
16 ]
```

Get by flightId



HTTP FlightSystem with MongoDB / Get by flightId

GET http://localhost:8080/api/v1.0/flight/get/6922aa8e2ceee1b34b6b629b... Send

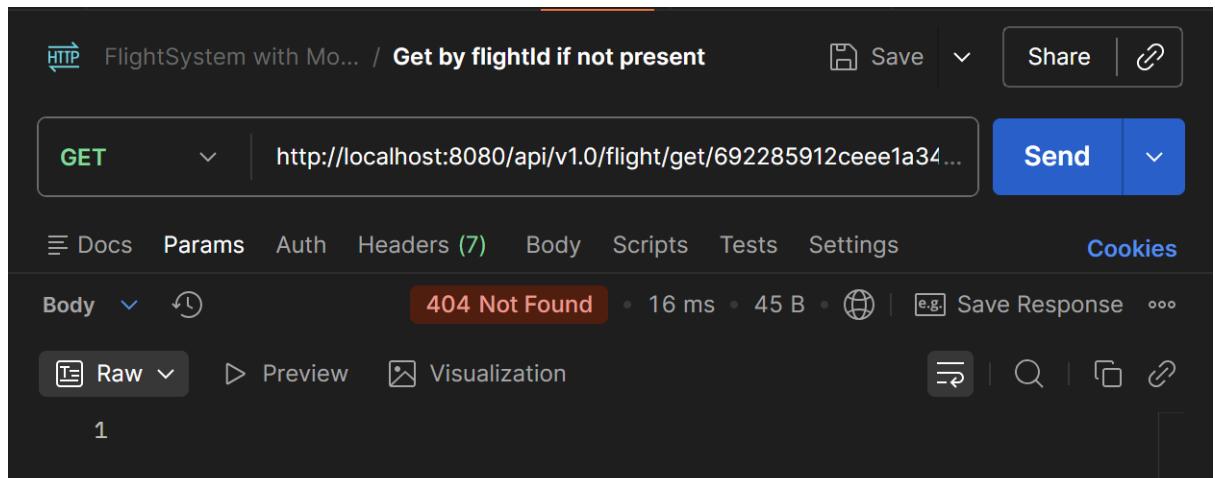
Docs Params Auth Headers (7) Body Scripts Tests Settings Cookies

Body Preview Visualization

200 OK • 19 ms • 339 B • Save Response ⚙️

```
{ } JSON ▾ 1 "id": "6922aa8e2ceee1b34b6b629b", 2 "fromPlace": "Delhi", 3 "toPlace": "Mumbai", 4 "arrivalTime": "2025-12-10T10:00:00", 5 "departureTime": "2025-12-10T07:30:00", 6 "availableSeats": 120, 7 "price": { 8 | "oneWay": 3200.0, 9 | "roundTrip": 6000.0 10 }, 11 "airlineId": "69227a002ceee1b34b6b6256", 12 "bookings": null 13 14
```

Get by flightId if not found



HTTP FlightSystem with Mo... / Get by flightId if not present

GET http://localhost:8080/api/v1.0/flight/get/692285912ceee1a34b6b6256... Send

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

Body Raw ▾ Preview Visualization

404 Not Found • 16 ms • 45 B • Save Response ⚙️

```
1
```

Get flights by searchRequest

The screenshot shows the Postman interface with a successful API call. The request URL is `http://localhost:8080/api/v1.0/flight/search`. The response status is `200 OK` 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 Postman interface with a failed API call. The request URL is `http://localhost:8080/api/v1.0/flight/search`. The response status is `204 No Content` with a response time of 16 ms and a body size of 27 B. The response body is empty.

Adding Booking

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/api/v1.0/flight/booking/6922aa8e2ceee...`. The request body is a JSON object representing a booking with two passengers. The response status is `201 Created` with a duration of `176 ms` and a 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 }
```

Add Booking when flight is not present

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/api/v1.0/flight/booking/6922aa8e2ceee...`. The request body is identical to the one in the previous screenshot. The response status is `400 Bad Request` with a duration of `13 ms` and a 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 }
```

When the seats are already booked

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** <http://localhost:8080/api/v1.0/flight/booking/6922aa8e2ceee1b...>
- Body:** JSON (shown below)
- Response Status:** 404 Not Found
- Response Time:** 50 ms
- Response Size:** 45 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    }  
19  ]  
20}  
21
```

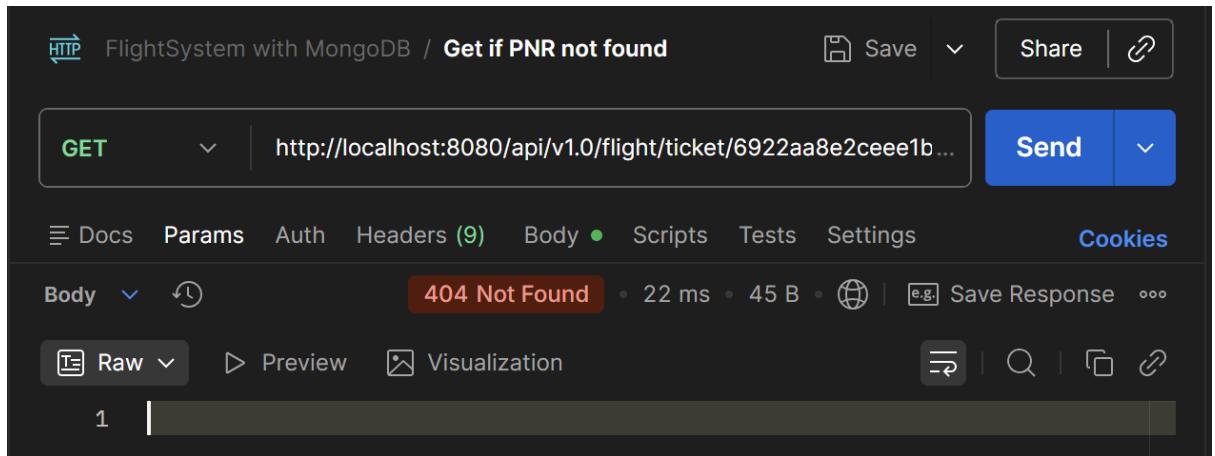
Get Using PNR

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:8080/api/v1.0/flight/ticket/6922aa8e2ceee1b...>
- Body:** JSON (shown below)
- Response Status:** 200 OK
- Response Time:** 12 ms
- Response Size:** 494 B

```
1 {  
2   "id": "6922bb72ceee1b34b6b631e",  
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  "totalAmount": 6000.0  
27},  
28  "13A",  
29  "13B"
```

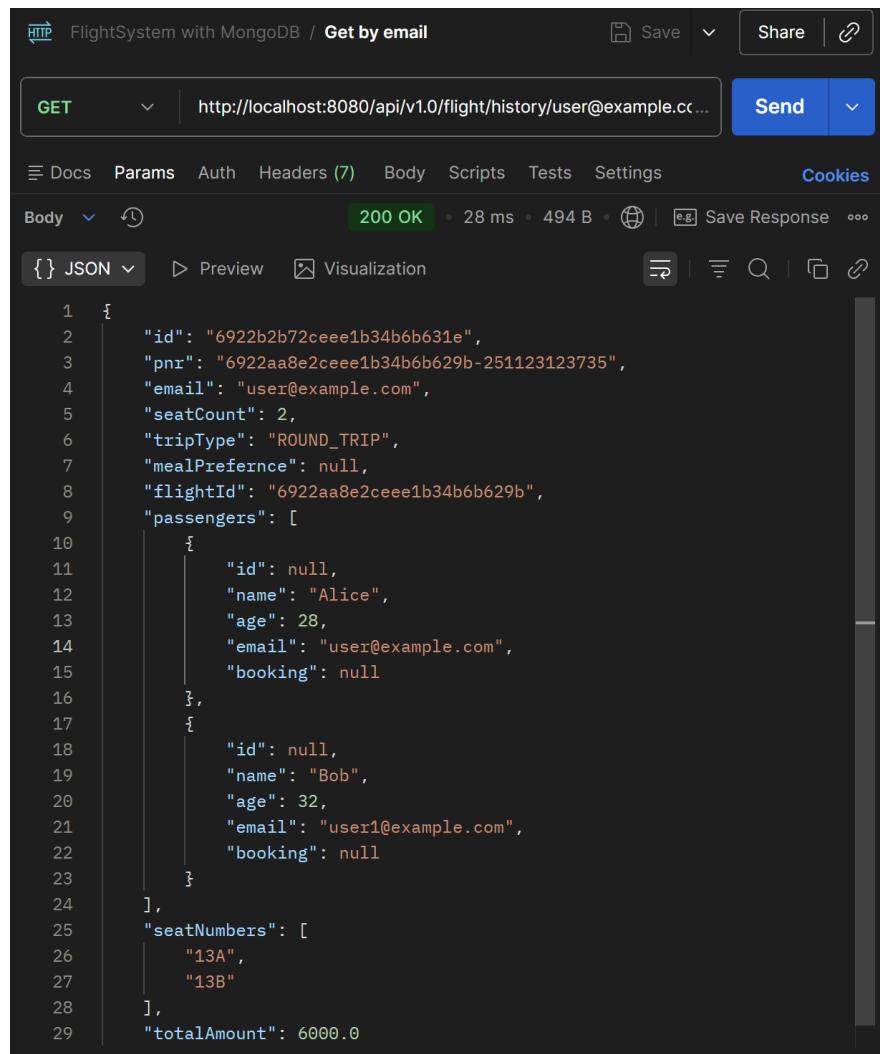
Get by PNR if not found



The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/v1.0/flight/ticket/6922aa8e2ceee1b...
- Status:** 404 Not Found
- Response Time:** 22 ms
- Size:** 45 B
- Save Response:** Enabled

Get booking by Email



The screenshot shows the Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/v1.0/flight/history/user@example.com...
- Status:** 200 OK
- Response Time:** 28 ms
- Size:** 494 B
- Save Response:** Enabled

Body (JSON):

```
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

The screenshot shows a Postman request for a GET operation at `http://localhost:8080/api/v1.0/flight/history/user@example1.c...`. The response status is 404 Not Found, with a duration of 9 ms and a body size of 45 B. The response body is empty.

Delete by PNR

The screenshot shows a Postman request for a DELETE operation at `http://localhost:8080/api/v1.0/flight/booking/cancel/6922aa8...`. The response status is 204 No Content, with a duration of 30 ms and a body size of 27 B. The response body is empty.

Delete if PNR not found

The screenshot shows a Postman request for a DELETE operation at `http://localhost:8080/api/v1.0/flight/booking/cancel/9922aa8...`. The response status is 404 Not Found, with a duration of 17 ms and a body size of 45 B. The response body is empty.

Adding Passenger

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** <http://localhost:8080/api/v1.0/passenger/add>
- Body (JSON):**

```
1 {
2     "name": "Alice",
3     "gender": "FEMALE",
4     "age": 28,
5     "email": "user@example.com"
6 }
7 }
```
- Response Status:** 201 Created
- Response Time:** 39 ms
- Response Size:** 43 B

Get passenger by id

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:8080/api/v1.0/passenger/get/6922b7ac2ceee1b34b6b6327>
- Body (JSON):**

```
{}
1 {
2     "id": "6922b7ac2ceee1b34b6b6327",
3     "name": "Alice",
4     "gender": "FEMALE",
5     "age": 28,
6     "email": "user@example.com",
7     "bookingId": null
8 }
```
- Response Status:** 200 OK
- Response Time:** 27 ms
- Response Size:** 191 B

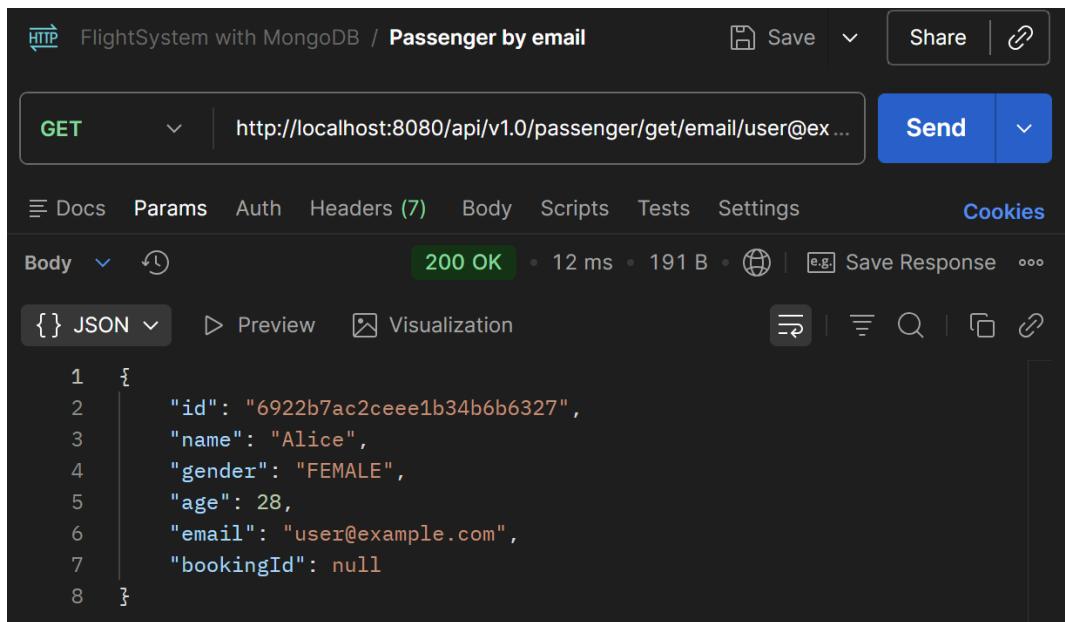
Get passenger by id if not found

The screenshot shows the Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:8080/api/v1.0/passenger/get/0922b7ac2ceee1b34b6b6327>
- Body (Raw):**

```
1
```
- Response Status:** 404 Not Found
- Response Time:** 18 ms
- Response Size:** 45 B

Get passenger by email

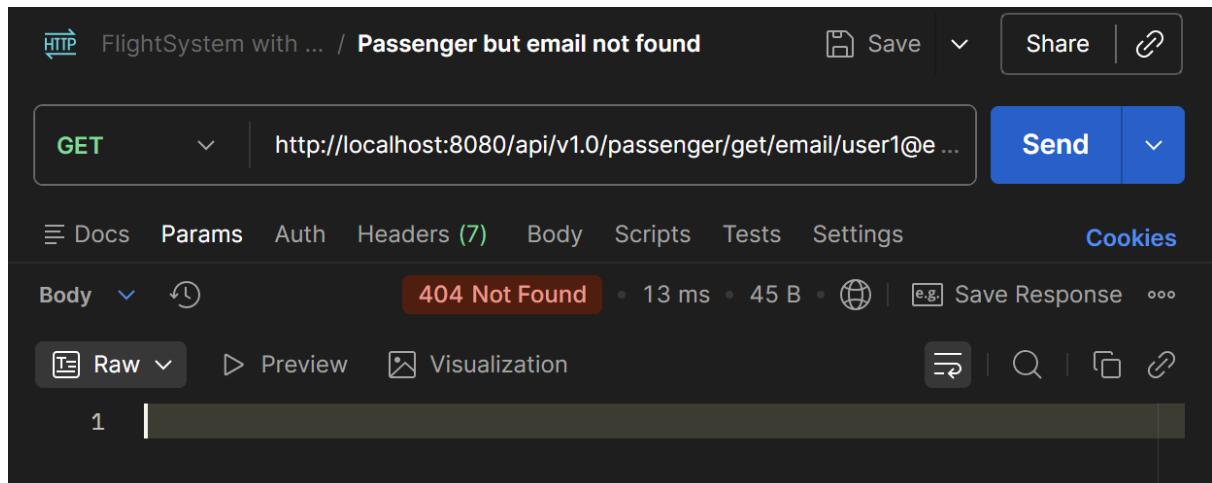


The screenshot shows a Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/v1.0/passenger/get/email/user@example.com
- Status:** 200 OK
- Body (JSON):**

```
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 a Postman interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8080/api/v1.0/passenger/get/email/user1@example.com
- Status:** 404 Not Found
- Body (Raw):**

```
1 |
```