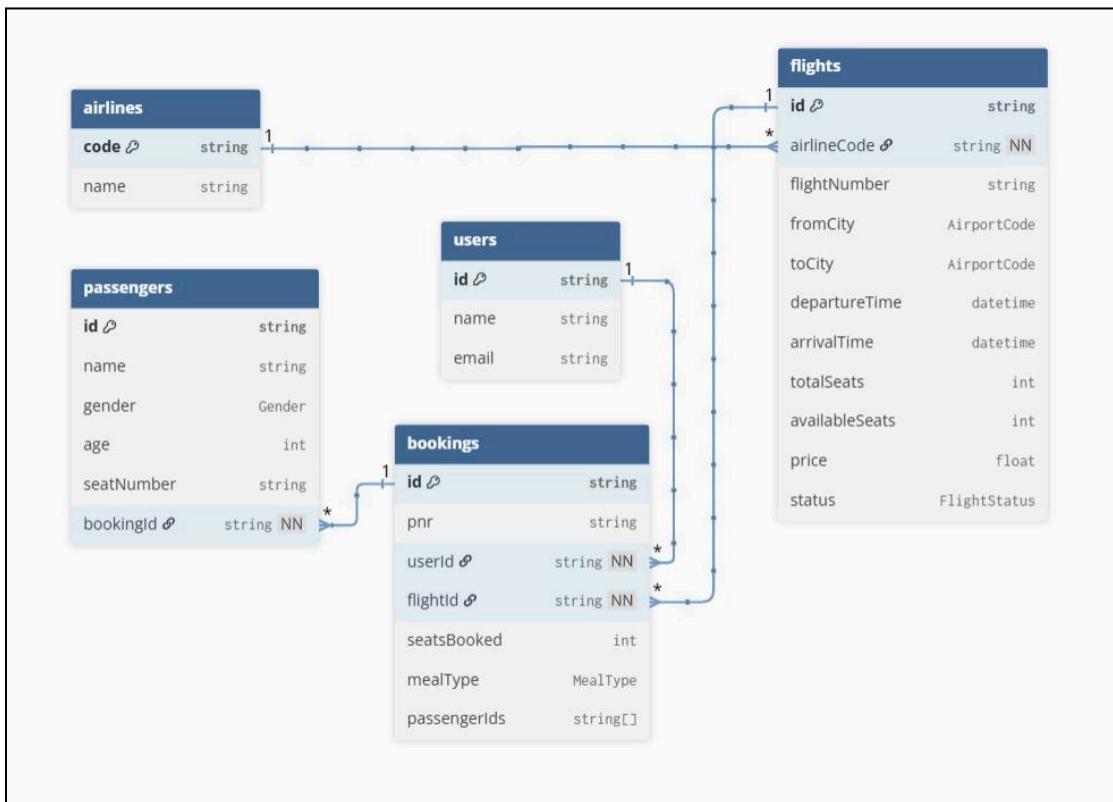


FLIGHT BOOKING SYSTEM - API documentation

(Spring Boot WebFlux + Reactive MongoDB)



AIRLINES

1. Add airline

The screenshot shows a POST request to `localhost:3000/airlines`. The request body is in JSON format:

```
1 {  
2   "code": "AIR123",  
3   "name": "Air India"  
4 }
```

The response status is `201 Created`, and the response body is `AIR123`.

POST localhost:3000/airlines

Docs Params Auth Headers (9) Body ● Script

raw JSON

```
1 {  
2   "code": "",  
3   "name": ""  
4 }
```

Body 400 Bad Request • 57

{ } JSON Preview Debug with AI

```
1 {  
2   "status": 400,  
3   "errors": {  
4     "code": "Airline code is required",  
5     "name": "Airline name is required"  
6   },  
7   "message": "Validation failed"  
8 }
```

2. Get all airlines

GET localhost:3000/airlines

Docs Params Auth Headers (7) Body S

raw JSON

```
1 Ctrl+Alt+P to Ask AI
```

Body 200 OK

{ } JSON Preview Visualize

```
1 [  
2   {  
3     "code": "AIR123",  
4     "name": "Air India"  
5   },  
6   {  
7     "code": "IN",  
8     "name": "Indigo"  
9   }  
10 ]
```

3. Get airline using code

GET localhost:3000/airlines/AIR123

Docs Params Auth Headers (7) Body Scripts

Query Params

| | Key | Value |
|------|--------|-------|
| Body | 200 OK | 15 m |

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

```
1 "code": "AIR123",
2 "name": "Air India"
3
```

GET localhost:3000/airlines/XYZ

Docs Params Auth Headers (7) Body Scripts

raw ▾ Text ▾

Body 404 Not Found

{ } JSON ▾ ▷ Preview ⚡ Debug with AI | ▾

```
1 {
2   "error": "Airline not found"
3 }
```

4. Update airline

PUT localhost:3000/airlines/AIR123

Docs Params Auth Headers (9) Body Script

raw ▾ JSON ▾

Body 200 OK 14 m

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

```
1 {
2   "name": "Air India Updated"
3 }
```

```
1 "code": "AIR123",
2 "name": "Air India Updated"
3
```

PUT localhost:3000/airlines/AIR123

Docs Params Auth Headers (9) Body Scripts Settings

raw JSON

```
1 {  
2   "name": ""  
3 }
```

Body 400 Bad Request 6 ms 169

{ } JSON ▾ ▷ Preview ⚡ Debug with AI |

```
1 {  
2   "status": 400,  
3   "errors": {  
4     "name": "Airline name is required"  
5   },  
6   "message": "Validation failed"  
7 }
```

5. Delete airline

DELETE localhost:3000/airlines/IN

Docs Params Auth Headers (7) Body Scripts Settings

none

This request does not have a body.

Body 200 OK 10 ms

Raw ▾ ▷ Preview ⚡ Visualize |

```
1
```

DELETE localhost:3000/airlines/ABC

Docs Params Auth Headers (7) Body Scripts Settings

none

This request does not have a body.

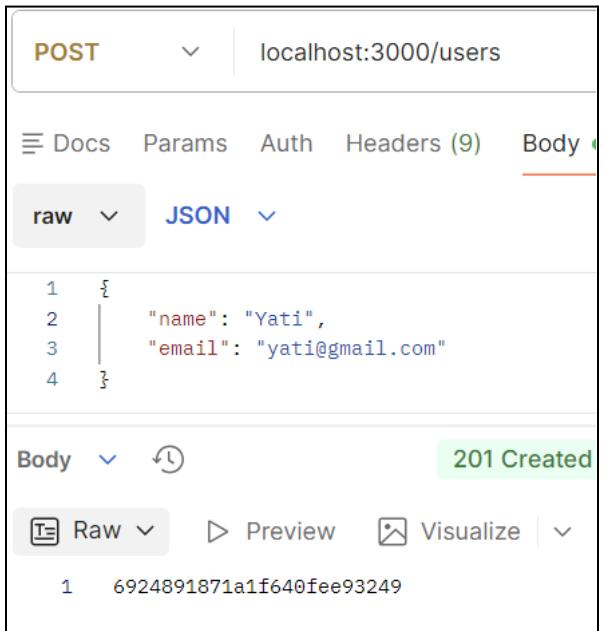
Body 404 Not Found

{ } JSON ▾ ▷ Preview ⚡ Debug with AI |

```
1 {  
2   "error": "Airline not found"  
3 }
```

USER

1. Add user



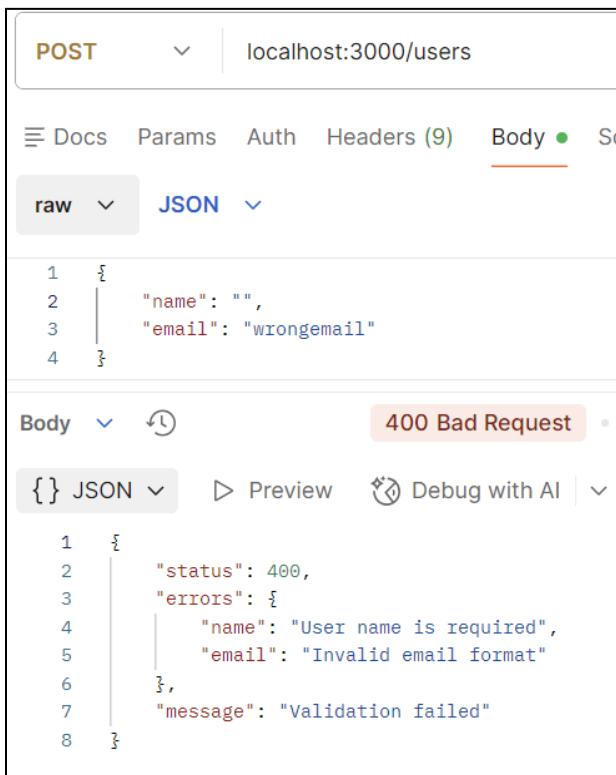
POST localhost:3000/users

Body (JSON)

```
1 {  
2   "name": "Yati",  
3   "email": "yati@gmail.com"  
4 }
```

Body (Raw) 201 Created

```
1 6924891871a1f640fee93249
```



POST localhost:3000/users

Body (JSON)

```
1 {  
2   "name": "",  
3   "email": "wrongemail"  
4 }
```

Body (Raw) 400 Bad Request

```
1 {  
2   "status": 400,  
3   "errors": {  
4     "name": "User name is required",  
5     "email": "Invalid email format"  
6   },  
7   "message": "Validation failed"  
8 }
```

2. Get all users

The screenshot shows a Postman request for a GET operation on the URL `localhost:3000/users`. The 'Params' tab is selected. The response status is `200 OK`, and the JSON body contains two user objects:

```
1 [  
2 {  
3   "id": "6924891871a1f640fee93249",  
4   "name": "Yati",  
5   "email": "yati@gmail.com"  
6 },  
7 {  
8   "id": "6924893071a1f640fee9324a",  
9   "name": "xyz",  
10  "email": "xyz@gmail.com"  
11 }  
12 ]
```

3. Get user by id

The screenshot shows a Postman request for a GET operation on the URL `localhost:3000/users/6924893071a1f640fee9324a`. The 'Params' tab is selected. The response status is `200 OK`, and the JSON body contains one user object:

```
1 {  
2   "id": "6924893071a1f640fee9324a",  
3   "name": "xyz",  
4   "email": "xyz@gmail.com"  
5 }
```

GET localhost:3000/users/1543541

Docs Params Auth Headers (7) Body Scripts

Query Params

| | Key | Value |
|--|-----|-------|
| | Key | Value |

Body 404 Not Found

{ } JSON ▾ Preview ⚡ Debug with AI

```
1 {  
2   "error": "User not found"  
3 }
```

4. Update user

PUT localhost:3000/users/6924893071a1f640fee9 ...

Docs Params Auth Headers (9) Body Scripts Settings

raw ▾ JSON ▾

```
1 {  
2   "name": "xyzz Updated",  
3   "email": "xyzupdated@gmail.com"  
4 }
```

Body 200 OK • 19 ms • 159 B

{ } JSON ▾ Preview Visualize

```
1 {  
2   "id": "6924893071a1f640fee9324a",  
3   "name": "xyzz Updated",  
4   "email": "xyzupdated@gmail.com"  
5 }
```

PUT <localhost:3000/users/69232007066e7447ab3...>

Docs Params Auth Headers (9) Body Scripts Settings

raw JSON

```
1 {  
2   "name": "",  
3   "email": "invalid"  
4 }
```

Body 400 Bad Request 7 ms 198 B

{ } JSON Preview Debug with AI

```
1 {  
2   "status": 400,  
3   "errors": {  
4     "name": "User name is required",  
5     "email": "Invalid email format"  
6   },  
7   "message": "Validation failed"  
8 }
```

5. Delete user

DELETE <localhost:3000/users/6924893071a1f640fee9...>

Docs Params Auth Headers (7) Body Scripts Settings

Query Params

| Key | Value | Desc... |
|-----|-------|---------|
| | | |

Body 200 OK 16 ms 38 B

Raw Preview Visualize

```
1
```

DELETE <localhost:3000/users/32133>

Docs Params Auth Headers (7) Body Scripts

Query Params

| Key | Value |
|-----|-------|
| | |

Body 404 Not Found 7 m

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "User not found"  
3 }
```

FLIGHT

1. Add flight

A screenshot of a POST request to `localhost:3000/flights/add`. The request body is a JSON object representing a flight with the following fields:

```
1 {  
2   "airlineCode": "AIR123",  
3   "flightNumber": "AI-200",  
4   "fromCity": "DEL",  
5   "toCity": "BOM",  
6   "departureTime": "2026-01-05T10:00:00",  
7   "arrivalTime": "2026-01-05T11:30:00",  
8   "totalSeats": 200,  
9   "availableSeats": 0,  
10  "price": 700,  
11  "status": "SCHEDULED"  
12 }
```

The response shows a `201 Created` status with the ID `69248b7271a1f640fee9324b`.

A screenshot of a POST request to `localhost:3000/flights/add`. The request body is a JSON object representing a flight with the following fields:

```
1 {  
2   "airlineCode": "AIR123",  
3   "flightNumber": "AI-111",  
4   "fromCity": "DEL",  
5   "toCity": "DEL",  
6   "departureTime": "2026-02-10T10:00:00",  
7   "arrivalTime": "2026-02-10T09:00:00",  
8   "totalSeats": 100,  
9   "availableSeats": 0,  
10  "price": 3500,  
11  "status": "SCHEDULED"  
12 }
```

The response shows a `400 Bad Request` status with the error message `"error": "source and destination must be different"`.

POST [localhost:3000/flights/add](#)

Docs Params Auth Headers (9) Body Scripts Setting

raw JSON

```
1 {  
2   "airlineCode": "AIR123",  
3   "flightNumber": "AI-111",  
4   "fromCity": "DEL",  
5   "toCity": "HYD",  
6   "departureTime": "2026-02-10T10:00:00",  
7   "arrivalTime": "2026-02-10T09:00:00",  
8   "totalSeats": 100,  
9   "availableSeats": 0,  
10  "price": 3500,  
11  "status": "SCHEDULED"  
12 }
```

Body 400 Bad Request 10 ms 133 B

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "arrival time must be after departure time"  
3 }
```

POST [localhost:3000/flights/add](#)

Docs Params Auth Headers (9) Body Scripts Setting

raw JSON

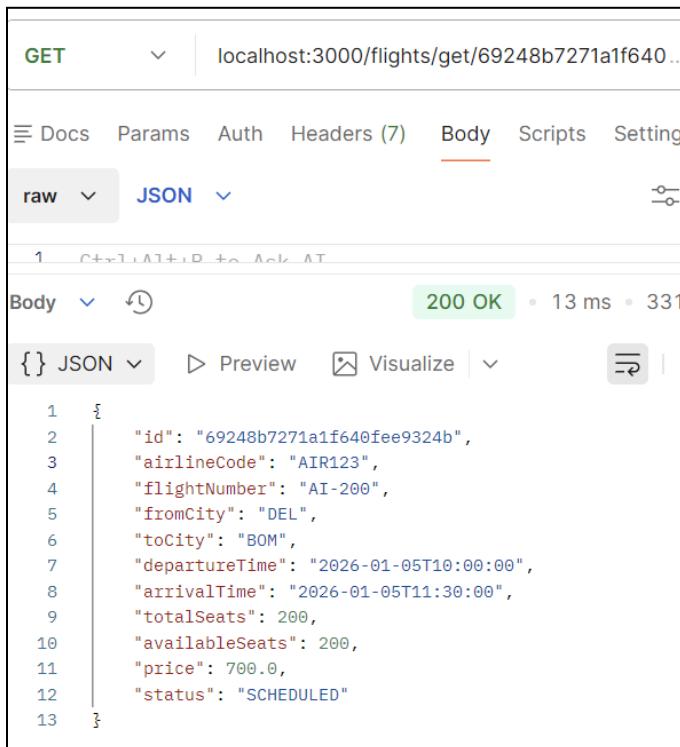
```
1 {  
2   "airlineCode": "XYZ",  
3   "flightNumber": "AI-500",  
4   "fromCity": "DEL",  
5   "toCity": "DEL",  
6   "departureTime": "2026-02-12T10:00:00",  
7   "arrivalTime": "2026-02-12T04:00:00",  
8   "totalSeats": 0,  
9   "availableSeats": 0,  
10  "price": -10,  
11  "status": "SCHEDULED"  
12 }
```

Body 400 Bad Request 11 ms 1.1 B

{ } JSON Preview Debug with AI

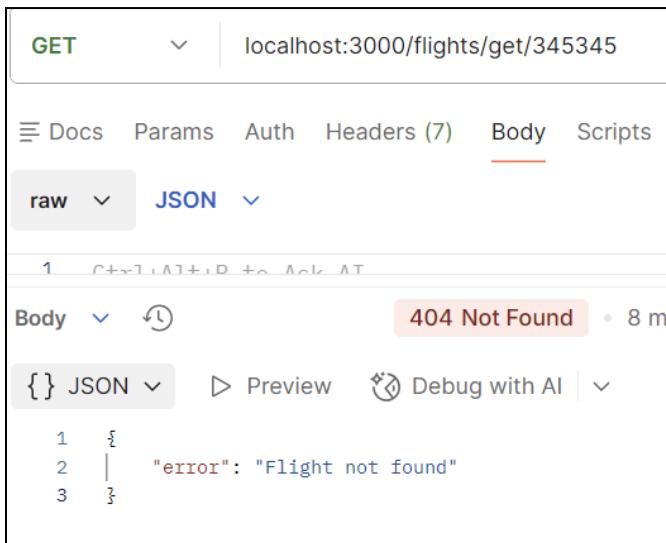
```
1 {  
2   "status": 400,  
3   "errors": {  
4     "price": "price>=0",  
5     "totalSeats": "totalSeats>=1"  
6   },  
7   "message": "Validation failed"
```

2. Get flight using id



GET localhost:3000/flights/get/69248b7271a1f640...

Body (JSON) {
1 {id: "69248b7271a1f640fee9324b",
2 airlineCode: "AIR123",
3 flightNumber: "AI-200",
4 fromCity: "DEL",
5 toCity: "BOM",
6 departureTime: "2026-01-05T10:00:00",
7 arrivalTime: "2026-01-05T11:30:00",
8 totalSeats: 200,
9 availableSeats: 200,
10 price: 700.0,
11 status: "SCHEDULED"
12 }
13 }



GET localhost:3000/flights/get/345345

Body (JSON) {
1 {
2 "error": "Flight not found"
3 }

3. Search flight using to , from and date

POST localhost:3000/flights/search

Docs Params Auth Headers (9) Body Sc

raw JSON

```
1 {  
2   "from": "DEL",  
3   "to": "BOM",  
4   "date": "2026-01-05"  
5 }
```

Body 200 OK 1

Raw Preview Visualize

```
1 69248b7271a1f640fee9324b
```

POST localhost:3000/flights/search

Docs Params Authorization Headers (9) Body

none form-data x-www-form-urlencoded

```
1 {  
2   "from": "HYD",  
3   "to": "DEL",  
4   "date": ""  
5 }
```

Body Cookies Headers (2) Test Results

{ } JSON

Preview Debug with AI

```
1 {  
2   "status": 400,  
3   "errors": {  
4     "date": "date is required"  
5   },  
6   "message": "Validation failed"  
7 }
```

4. Get all flights of a particular airline

GET localhost:3000/airlines/AIR123/flights

Docs Params Auth Headers (7) Body Scripts Settings

Body 200 OK • 7 ms

{ } JSON ▾ ▶ Preview Visualize | ▾

```
1 [  
2 {  
3   "id": "69248b7271a1f640fee9324b",  
4   "airlineCode": "AIR123",  
5   "flightNumber": "AI-200",  
6   "fromCity": "DEL",  
7   "toCity": "BOM",  
8   "departureTime": "2026-01-05T10:00:00",  
9   "arrivalTime": "2026-01-05T11:30:00",  
10  "totalSeats": 200,  
11  "availableSeats": 200,  
12  "price": 700.0,  
13  "status": "SCHEDULED"  
14 },  
15 {  
16   "id": "69248cef71a1f640fee9324c",  
17   "airlineCode": "AIR123",  
18   "flightNumber": "AI-301",  
19   "fromCity": "HYD",  
20   "toCity": "DEL",  
21   "departureTime": "2025-12-15T10:00:00",  
22   "arrivalTime": "2025-12-15T11:30:00",  
23   "totalSeats": 180,
```

BOOKING

1. Create booking

POST localhost:3000/bookings/create

Docs Params Auth Headers (9) Body Scripts

raw JSON

```
1 {  
2   "flightId": "69248b7271a1f640fee9324b",  
3   "userId": "6924891871a1f640fee93249",  
4   "seatsBooked": 1,  
5   "mealType": "NON_VEG",  
6   "flightType": "ONE WAY",  
7   "passengers": [  
8     {  
9       "name": "xyz",  
10      "gender": "M",  
11      "age": 21,  
12      "seatNumber": "A2"  
13    }  
14  ]  
15 }
```

Body 201 Created

Raw Preview Visualize

1 PNRCA256177

POST localhost:3000/bookings/create

Docs Params Auth Headers (9) Body Scripts

raw JSON

```
1 {  
2   "flightId": "69248b7271a1f640fee9324b",  
3   "userId": "213124",  
4   "seatsBooked": 1,  
5   "mealType": "NON_VEG",  
6   "flightType": "ONE WAY",  
7   "passengers": [  
8     {  
9       "name": "xyz",  
10      "gender": "M",  
11      "age": 21,  
12      "seatNumber": "A2"  
13    }  
14  ]  
15 }
```

Body 404 Not Found

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "User not found"  
3 }
```

POST localhost:3000/bookings/create

Docs Params Auth Headers (9) Body Scripts

raw JSON

```
1 {  
2   "flightId": "dwefdw",  
3   "userId": "6924891871a1f640fee93249",  
4   "seatsBooked": 1,  
5   "mealType": "NON_VEG",  
6   "flightType": "ONE WAY",  
7   "passengers": [  
8     {  
9       "name": "xyz",  
10      "gender": "M",  
11      "age": 21,  
12      "seatNumber": "A2"  
13    }  
14  ]  
15 }
```

Body 404 Not Found

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "Flight not found"  
3 }
```

POST localhost:3000/bookings/create

Docs Params Auth Headers (9) Body Scripts Settings

raw JSON

```
1 {  
2   "flightId": "69248b7271a1f640fee9324b",  
3   "userId": "6924891871a1f640fee93249",  
4   "seatsBooked": 2,  
5   "mealType": "VEG",  
6   "flightType": "ONE WAY",  
7   "passengers": [  
8     {  
9       "name": "Amit",  
10      "gender": "M",  
11      "age": 30,  
12      "seatNumber": "A1"  
13    },  
14    {  
15      "name": "Rita",  
16    }  
17  ]  
18 }
```

Body 400 Bad Request 20 ms

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "Seat already booked: A2"  
3 }
```

POST localhost:3000/bookings/create

Docs Params Auth Headers (9) Body Scripts Settings

raw JSON

```
1 {  
2   "flightId": "69248b7271a1f640fee9324b",  
3   "userId": "6924891871a1f640fee93249",  
4   "seatsBooked": 2,  
5   "mealType": "VEG",  
6   "flightType": "ONE WAY",  
7   "passengers": [  
8     {  
9       "name": "Amit",  
10      "gender": "M",  
11      "age": 30,  
12      "seatNumber": "A9"  
13    }  
14  ]  
15 }
```

Body 400 Bad Request • 14 ms • 138

{ } JSON Preview Debug with AI

```
1 {  
2   "error": "Passenger count must be equal to seats booked"  
3 }
```

2. Get booking using pnr

GET localhost:3000/bookings/get/PNRCA256177

Docs Params Auth Headers (7) Body Scripts Settings

Body 200 OK • 8 ms

{ } JSON Preview Visualize

```
1 {  
2   "id": "69248dac71a1f640fee9324d",  
3   "pnr": "PNRCA256177",  
4   "userId": "6924891871a1f640fee93249",  
5   "flightId": "69248b7271a1f640fee9324b",  
6   "seatsBooked": 1,  
7   "mealType": "NON_VEG",  
8   "flightType": "ONE WAY",  
9   "passengerIds": [  
10    "69248dac71a1f640fee9324e"  
11  ]  
12 }
```

GET localhost:3000/bookings/get/PNR2379

Docs Params Auth Headers (7) Body Scripts Settings

Query Params

| Key | Value |
|-----|-------|
| Key | Value |

Body 404 Not Found 9 ms

{ } JSON ▾ Preview ⚡ Debug with AI

```
1 {  
2   "error": "PNR not found"  
3 }
```

3. Get booking details using email id

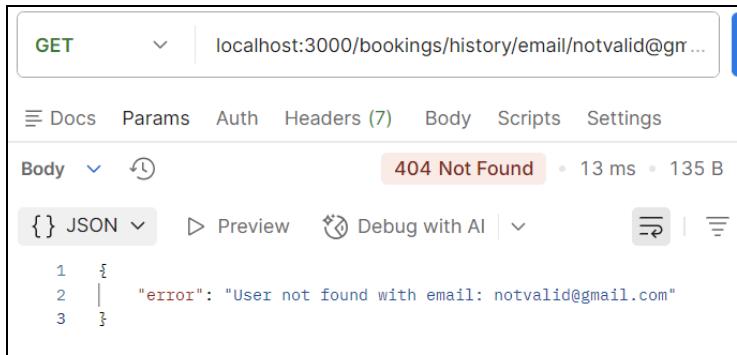
GET localhost:3000/bookings/history/email/yati@gmail.com

Docs Params Auth Headers (7) Body Scripts Settings

Body 200 OK 10 ms 540 B

{ } JSON ▾ Preview Visualize

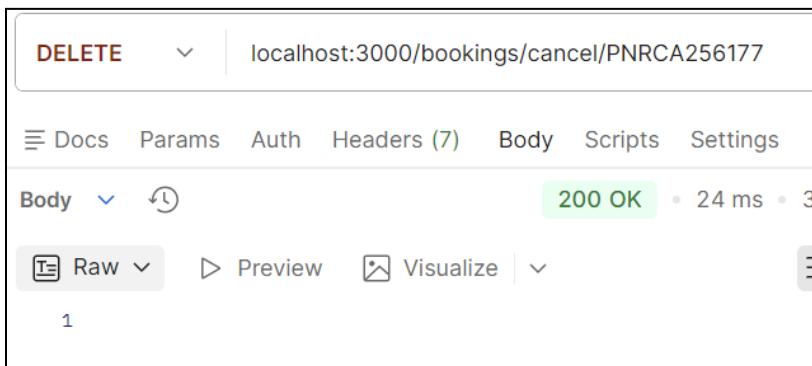
```
1 [  
2   {  
3     "id": "69248dac71a1f640fee9324d",  
4     "pnr": "PNRCA256177",  
5     "userId": "6924891871a1f640fee93249",  
6     "flightId": "69248b7271a1f640fee9324b",  
7     "seatsBooked": 1,  
8     "mealType": "NON_VEG",  
9     "flightType": "ONE WAY",  
10    "passengerIds": [  
11      "69248dac71a1f640fee9324e"  
12    ]  
13  },  
14  {  
15    "id": "69248e8f71a1f640fee9324f",  
16    "pnr": "PNR23E67A50",  
17    "userId": "6924891871a1f640fee93249",  
18    "flightId": "69248cef71a1f640fee9324c",  
19    "seatsBooked": 1,  
20    "mealType": "VEG",  
21    "flightType": "ONE WAY",  
22    "passengerIds": [  
23      "69248e8f71a1f640fee93250"
```



A screenshot of a POST request in a developer tool. The method is "POST" and the URL is "localhost:3000/bookings/history/email/notvalid@gmail.com". The response status is "404 Not Found" with a duration of 13 ms and a size of 135 B. The response body is a JSON object with one error message: "User not found with email: notvalid@gmail.com".

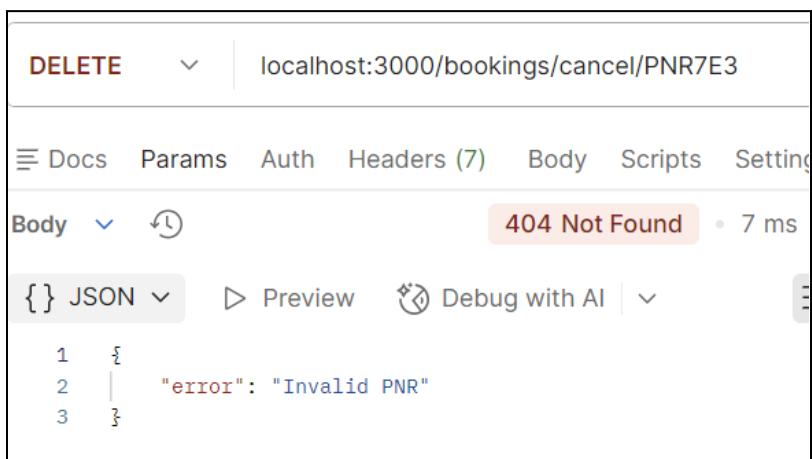
```
1  {
2  |   "error": "User not found with email: notvalid@gmail.com"
3 }
```

4. Cancel booking



A screenshot of a DELETE request in a developer tool. The method is "DELETE" and the URL is "localhost:3000/bookings/cancel/PNRCA256177". The response status is "200 OK" with a duration of 24 ms and a size of 3 B. The response body is empty.

```
1
```



A screenshot of a DELETE request in a developer tool. The method is "DELETE" and the URL is "localhost:3000/bookings/cancel/PNR7E3". The response status is "404 Not Found" with a duration of 7 ms. The response body is a JSON object with one error message: "Invalid PNR".

```
1  {
2  |   "error": "Invalid PNR"
3 }
```

5. Get all booking done by a user

GET <localhost:3000/bookings/history/user/6924891871a...>

Docs Params Auth Headers (7) Body Scripts Settings

Body [↻](#) 200 OK • 7 ms • 308 B •

{ } JSON ▾ ▶ Preview [Visualize](#) | ▾

```
1 [  
2 {  
3   "id": "69248e8f71a1f640fee9324f",  
4   "pnr": "PNR23E67A50",  
5   "userId": "6924891871a1f640fee93249",  
6   "flightId": "69248cef71a1f640fee9324c",  
7   "seatsBooked": 1,  
8   "mealType": "VEG",  
9   "flightType": "ONE WAY",  
10  "passengerIds": [  
11    "69248e8f71a1f640fee93250"  
12  ]  
13 }  
14 ]
```