

Add Flight

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Environment:** chubb-flight-reactive-mongodb
- Request Method:** POST
- Request URL:** http://localhost:8085/api/flight/airline/add
- Body (raw JSON):**

```
1 {
2   "name": "Air India",
3   "logoUrl": "https://logo.com/airindia.png"
4 }
```
- Response Status:** 201 Created
- Response Body:**

```
1 {
2   "id": "6924b10dcac190626df5194f",
3   "name": "Air India",
4   "logoUrl": "https://logo.com/airindia.png"
5 }
```

Add Flight Inventory

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Environment:** chubb-flight-reactive-mongodb
- Request Method:** POST
- Request URL:** http://localhost:8085/api/flight/airline/inventory/add
- Body (raw JSON):**

```
1 {
2   "airlineId": "A1",
3   "fromPlace": "Hyderabad",
4   "toPlace": "Bangalore",
5   "availableSeats": 50,
6   "oneWayPrice": 3000,
7   "roundTripPrice": 5500
8 }
```
- Response Status:** 201 Created
- Response Body:**

```
1 {
2   "id": "6924b12bcac190626df51950",
3   "airlineId": "A1"
4 }
```

Get All Inventory

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /chubb-flight-reactive-mongodb / get all inventory
- URL:** http://localhost:8085/api/flight/airline/inventory
- Body:** JSON (Preview shows a flight record with 11 seats)
- Response:** 200 OK (116 ms, 2.56 KB)

Search Flight

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** POST /chubb-flight-reactive-mongodb / search flight
- URL:** http://localhost:8085/api/flight/search
- Body:** Raw JSON (fromPlace: Hyderabad, toPlace: Bangalore)
- Response:** 200 OK (80 ms, 984 B)

Search Flight by ID

The screenshot shows the Postman interface with the following details:

- Request URL:** http://localhost:8085/api/flight/get/692495c42c815049f006d256
- Method:** GET
- Body:** JSON (Preview tab)
- Response:** 200 OK (14 ms, 483 B)
 - id: "692495c42c815049f006d256",
airlineId: "A1",
fromPlace: "Hyderabad",
toPlace: "Bangalore",
departureTime: null,
arrivalTime: null,
availableSeats: 20,
oneWayPrice: 3500,
roundTripPrice: 5500,
seats: [
 "S1",
 "S2",
 "S3",
 "S4",
 "S5",
 "S6",
 "S7",
 "S8",
 "S9",
 "S10",
 "S11",
 "S12",
 "S13",
 "S14",
 "S15",
 "S16",
 "S17",
 "S18",
 "S19",
 "S20"]

Get All Flights

The screenshot shows the Postman interface with the following details:

- Request URL:** http://localhost:8085/api/flight/get/flights
- Method:** GET
- Body:** JSON (Preview tab)
- Response:** 200 OK (45 ms, 3.13 KB)
 - id: "6924b12bcac190626df51950",
airlineId: "A1",
fromPlace: "Hyderabad",
toPlace: "Bangalore",
departureTime: null,
arrivalTime: null,
availableSeats: 50,
oneWayPrice: 3000,
roundTripPrice: 5500,
seats: [
 "S1",
 "S2",
 "S3",
 "S4",
 "S5",
 "S6",
 "S7",
 "S8",
 "S9",
 "S10",
 "S11",
 "S12",
 "S13",
 "S14",
 "S15",
 "S16",
 "S17",
 "S18",
 "S19",
 "S20"]

Book Ticket

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > chubb-flight-reactive-mongodb
- Request Type:** POST
- URL:** http://localhost:8085/api/flight/booking/F1
- Body (raw JSON):**

```
1 {
2   "email": "abc@example.com",
3   "seatCount": 2,
4   "seatNumbers": ["S1", "S2"],
5   "passengers": [
6     {
7       "id": "P1",
8       "name": "ABC",
9       "gender": "MALE",
10      "age": 25,
11      "email": "abc@example.com",
12      "bookingId": ""
13    }
14  ]
15 }
```

- Response Status:** 201 Created

Get Ticket by PNR

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > chubb-flight-reactive-mongodb
- Request Type:** GET
- URL:** http://localhost:8085/api/flight/ticket/06D03F
- Headers:** (6)
- Query Params:**

Key	Value	Description	Bulk Edit
Key	Value	Description	

- Response Status:** 200 OK
- Body (JSON):**

```
1 {
2   "id": "69235b50a7a0842213998f09",
3   "pnr": "06D03F",
4   "flightId": "F101",
5   "airlineId": "A1",
6   "email": "yasmitha@example.com",
7   "seatCount": 2,
8   "tripType": "ONE WAY",
9   "mealPreference": "VEG",
10  "passengers": [
11    {
12      "id": null,
13      "name": "Yasmitha",
14      "gender": "FEMALE",
15      "age": 21,
16    }
17  ]
18 }
```

Get History by Email

The screenshot shows the Postman interface with the following details:

- Collection:** chubb-flight-reactive-mongodb
- Request:** GET /api/flight/booking/history/yaswita@example.com
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 [  
2 {  
3   "id": "69235b50a7a0842213998f09",  
4   "pnr": "06D03F",  
5   "flightId": "F101",  
6   "airlineId": "A1",  
7   "email": "yaswita@example.com",  
8   "seatCount": 2,  
9   "tripType": "ONE WAY",  
10  "mealPreference": "VEG",  
11  "passengers": [  
12    {  
13      "id": null,  
14      "name": "Yaswita",  
15      "gender": "FEMALE",  
16    }  
17  ]  
18 }  
19 ]
```

Cancel Ticket by PNR

The screenshot shows the Postman interface with the following details:

- Collection:** chubb-flight-reactive-mongodb
- Request:** DELETE /api/flight/booking/cancel/06D03F
- Response Status:** 200 OK
- Response Body (Raw JSON):**

```
1
```

Cancel Ticket by PNR(for invalid PNR)

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:8085/api/flight/booking/cancel/JBBJ`
- Method:** `DELETE`
- Body:** `{}` (empty JSON object)
- Response Status:** `400 Bad Request`
- Response Body:**

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

Add Passenger

The screenshot shows the Postman interface with the following details:

- Request URL:** `http://localhost:8085/api/flight/passenger/add`
- Method:** `POST`
- Body:**

```
1 {  
2   "id": "PX1",  
3   "name": "ABC",  
4   "gender": "MALE",  
5   "age": 25,  
6   "email": "abc@example.com",  
7   "bookingId": "B101"  
8 }
```
- Response Status:** `201 Created`
- Response Body:**

```
1 {  
2   "id": "PX1",  
3   "name": "ABC",  
4   "gender": "MALE",  
5   "age": 25,  
6   "email": "abc@example.com",  
7   "bookingId": "B101"  
8 }
```

Get Passenger by ID

The screenshot shows the Postman interface with the following details:

- Collection:** chubb-flight-reactive-mongodb
- Request:** GET http://localhost:8085/api/flight/passenger/PX1
- Query Params:** Key: Key, Value: Value
- Response:** 200 OK, 29 ms, 259 B. The response body is a JSON object:

```
{ "id": "PX1", "name": "ABC", "gender": "MALE", "age": 25, "email": "abc@example.com", "bookingId": "B101" }
```

Get Passenger by Booking ID

The screenshot shows the Postman interface with the following details:

- Collection:** chubb-flight-reactive-mongodb
- Request:** GET http://localhost:8085/api/flight/passenger/booking/B101
- Query Params:** Key: Key, Value: Value
- Response:** 200 OK, 25 ms, 380 B. The response body is a JSON array:

```
[ { "id": "69235bc6a7a0842213998f0a", "name": "abc", "gender": "FEMALE", "age": 23, "email": "abc@example.com", "bookingId": "B101" }, { "id": "PX1", "name": "ABC", "gender": "MALE", "age": 25, "email": "abc@example.com", "bookingId": "B101" } ]
```

Get All Passengers

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET http://localhost:8085/api/flight/passenger/all
- Params:** None
- Body:** JSON (Preview shows two passenger objects)
- Response:** 200 OK (22 ms, 380 B)

```
1 [  
2 {  
3   "id": "69235bc6a7a0842213998f0a",  
4   "name": "abc",  
5   "gender": "FEMALE",  
6   "age": 23,  
7   "email": "abc@example.com",  
8   "bookingId": "B101"  
9 },  
10 {  
11   "id": "PXi",  
12   "name": "ABC",  
13   "gender": "MALE",  
14   "age": 25,  
15   "email": "abc@example.com",  
16 }]
```

Delete passenger by ID

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** DELETE http://localhost:8085/api/flight/passenger/delete/PX1
- Params:** None
- Body:** Raw (Preview shows the number 1)
- Response:** 200 OK (10 ms, 123 B)