

## CMPE 275 Lab 2 - REST API, Persistence, and Transactions



### **Team Member info,**

Name : Nilam Pratim Deka  
Student-ID : [REDACTED]  
Email-ID : nilampratim.deka@sjsu.edu

Name : Venkatesh Donthi Shashidhar  
Student-ID : [REDACTED]  
Email-ID : venkatesh.donthishashidhar@sjsu.edu

**Cloud Service Choice** : Amazon Web Services

**Host IP address** : 54.183.193.226

## Screenshots,

### (1) Get a passenger back as JSON

**Method:** GET

**URI:** <https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207>

#### a) Success Case

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Collection, History, Help, Runner, Import, and a search bar. The main workspace has tabs for https://54.183.193.226:8443, https://54.183.193.226:8443, https://54.183.193.226:8443, and https://54.183.193.226:8443. The current tab is set to GET and points to the URI https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207. The status bar at the bottom indicates a 200 OK response with a time of 123 ms and a size of 926 B.

The left sidebar displays a history of API requests, including various GET, POST, PUT, and DELETE methods across different URLs. The right panel shows the response body in JSON format, which includes the passenger's ID, first name, last name, age, gender, phone number, and a list of flight reservations.

```
1: {
  2:   "id": "002aeaae-7534-40bd-aab8-407864465207",
  3:   "firstname": "aberercXX",
  4:   "lastname": "Y",
  5:   "age": 11,
  6:   "gender": "female",
  7:   "phone": "1234567321",
  8:   "reservations": [
  9:     {
  10:       "orderNumber": "05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24",
  11:       "price": 240,
  12:       "flights": [
  13:         {
  14:           "flightNumber": "britishairways",
  15:           "price": 120,
  16:           "from": "AA",
  17:           "to": "BB",
  18:           "departureTime": "2018-03-22-19",
  19:           "arrivalTime": "2018-03-24-19",
  20:           "description": "EE",
  21:           "plane": {
  22:             "capacity": 100,
  23:             "model": "HH",
  24:             "manufacturer": "II",
  25:             "yearOfManufacture": 1997
  26:           },
  27:           "seatsLeft": 94
  28:         },
  29:         {
  30:           "flightNumber": "indigo",
  31:           "price": 120
  32:         }
  33:       ]
  34:     }
  35:   ]
  36: }
```

#### b) Failure Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e646>

The screenshot shows the Postman application window. In the left sidebar, there's a history of requests. The main area shows a single request in the 'Builder' tab:

- Method:** GET
- URL:** https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e646
- Status:** 404 Not Found
- Time:** 208 ms
- Size:** 318 B

The response body is displayed in JSON format:

```

1 | {
2 |   "response": {
3 |     "code": 404,
4 |     "msg": "Sorry, the requested passenger with id ece171d7-a835-4f1d-acbc-3e646 does not exist"
5 |   }
6 |

```

## (2) Get a passenger back as XML

**Method:** GET

**URI:** <https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207?xml=true>

a) Success Case:

The screenshot shows the Postman interface with a successful response. The URL is `https://54.183.193.226:8443/passenger/002aeaaa-7534-40bd-aab8-4078`. The response status is 200 OK, time is 159 ms, and size is 1.24 KB. The response body is an XML document:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <passenger>
3   <age>11</age>
4   <firstname>aberercXX</firstname>
5   <gender>female</gender>
6   <id>002aeaaa-7534-40bd-aab8-407864465207</id>
7   <lastname>YY</lastname>
8   <phone>1234567321</phone>
9   <reservation>
10    <orderNumber>05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24</orderNumber>
11    <price>240</price>
12    <flights>
13      <flightNumber>britishairways</flightNumber>
14      <price>120</price>
15      <from>AA</from>
16      <to>BB</to>
17      <departureTime>2018-03-22-19</departureTime>
18      <arrivalTime>2018-03-24-19</arrivalTime>
19      <description>EE</description>
20      <plane>
21        <capacity>100</capacity>
22        <manufacturer>II</manufacturer>
23        <model>HH</model>
24        <yearOfManufacture>1997</yearOfManufacture>
25      </plane>
26      <seatsLeft>94</seatsLeft>
27    </flights>
28    <flights>
29      <flightNumber>indigo</flightNumber>
30      <price>120</price>
31      <from>AA</from>

```

## b) Failure Case:

**Method:** GET

**URI:** `https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e646610341?xml=true`

The screenshot shows the Postman interface with a 404 Not Found error. The URL is `https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e646610341?xml=true`. The response status is 404 Not Found, time is 269 ms, and size is 403 B. The response body is XML:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <badRequest>
3   <response>
4     <code>404</code>
5     <msg>Sorry, the requested passenger with id ece171d7-a835-4f1d-acbc-3e646610341 does not exist</msg>
6   </response>
7 </badRequest>

```

### (3) Create a passenger

#### a) Success Case

**Method:** POST

**URI:**

<https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=33&gender=male&phone=1234567890>

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Collection, History, Help, Runner, Import, and a search bar. The main area is titled 'Builder' with tabs for 'Runner' and 'Team Library'. A URL field contains 'https://54.183.193.226:8443'. Below the URL are buttons for 'POST', 'Params', 'Send', and 'Save'. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 864 ms', and 'Size: 265 B'. The 'Body' tab is selected, showing a JSON response with the following content:

```
[{"id": "ee054d04-2218-41af-80a7-e97c7a22fe0f", "firstname": "Michael", "lastname": "Scofield", "age": 33, "gender": "male", "phone": "1234567890"}]
```

#### b) Failure Case

**Method:** POST

**URI:**

<https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=33&gender=male&phone=1234567890>

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for File, Edit, View, Collection, History, Help, and a sign-in button. Below the navigation is a toolbar with icons for Runner, Import, and a search bar. The main workspace has tabs for https://54.183.193.226:8443, https://54.183.193.226:8443 (active), https://54.183.193.226:8443, and https://hostname/passenger. A '+' icon is available to add more tabs. The status bar indicates 'No Environment'. On the right side of the interface, there are buttons for Send, Save, and other settings.

The left sidebar lists recent history and collections. The 'History' section shows several requests:

- POST** https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=33&gender=male&phone=1234567321
- POST** https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=33&gender=male&phone=1234567321
- GET** https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e646610341?xml=true
- GET** https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e6466103415?xml=true
- GET** https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e6466103415
- GET** https://54.183.193.226:8443/passenger/ece171d7-a835-4f1d-acbc-3e6466103415?json=true
- POST** https://54.183.193.226:8443/reservation?passengerId=ece171d7-a835-4f1

The central workspace displays a POST request to https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=33&gender=male&phone=1234567321. The status bar shows Status: 400 Bad Request, Time: 356 ms, and Size: 245 B. The response body is JSON:

```

1  {
2   "response": {
3     "code": 400,
4     "msg": "another passenger with the same number already exists"
5   }
6 }

```

## (4) Update a passenger

### a) Success Case

**Method:** PUT

**URI:** <https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207?firstname=aberercXX&lastname=YY&age=11&gender=famale&phone=1234567321>

The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. Below the header, there are tabs for 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is active, showing a JSON payload:

```

1  [
2   "id": "002aeaae-7534-40bd-aab8-407864465207",
3   "firstname": "aberercXX",
4   "lastname": "YY",
5   "age": 11,
6   "gender": "female",
7   "phone": "1234567321",
8   "reservations": []
9 ]

```

At the bottom right of the body panel, it says 'Status: 200 OK Time: 46 ms Size: 345 B'. On the left side of the interface, a list of previous requests is visible, all using the PUT method and pointing to the same endpoint: `https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207?firstname=aberercXX&lastname=YY&age=11&gender=famale&phone=12345`.

## b) Failure Case

### Method: PUT

**URI:** `https://54.183.193.226:8443/passenger/002aeaae-7534-40bd-aab8-407864465207?firstname=aberercXX&lastname=YY&age=11&gender=famale&phone=12345`

The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. Below the header, there are tabs for 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is active, showing a JSON payload:

```

1  [
2   "code": 400,
3   "msg": "another passenger with the same number already exists"
4 ]

```

At the bottom right of the body panel, it says 'Status: 400 Bad Request Time: 275 ms Size: 296 B'. On the left side of the interface, a list of previous requests is visible, including a failed PUT request with the error message above.

## (5) Delete a passenger

a) Success Case:

**Method:** DELETE

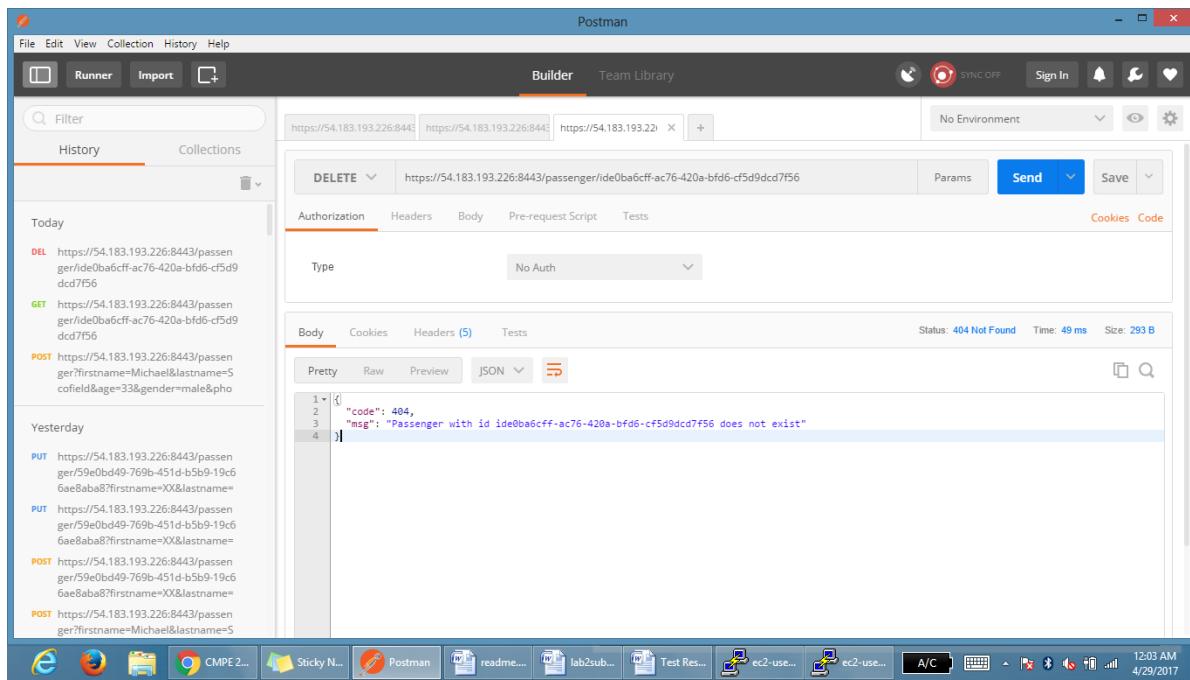
**URI:** <https://54.183.193.226:8443/passenger/e0ba6cff-ac76-420a-bfd6-cf5d9dcd7f56>

The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. The URL field contains 'https://54.183.193.226:8443/passenger/e0ba6cff-ac76-420a-bfd6-cf5d9dcd7f56'. The 'Authorization' tab is selected, showing 'No Auth'. The 'Body' tab is selected, displaying a JSON response: { "code": 200, "msg": "Passenger with id e0ba6cff-ac76-420a-bfd6-cf5d9dcd7f56 is deleted successfully" }. The status bar at the bottom right shows 'Status: 200 OK Time: 103 ms Size: 293 B'. The bottom taskbar includes icons for CMPE 2..., Sticky N..., Postman, readme..., lab2sub..., Test Res..., ec2-use..., ec2-use..., A/C, and keyboard.

b) Failure Case

**Method:** DELETE

**URI:** <https://54.183.193.226:8443/passenger/ide0ba6cff-ac76-420a-bfd6-cf5d9dcd7f56>



## (6) Get a reservation back as JSON

### a) Success Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/reservation/05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24>

```

1 [ {
2   "orderNumber": "05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24",
3   "price": 240,
4   "flights": [
5     {
6       "flightNumber": "britishairways",
7       "price": 120,
8       "from": "AA",
9       "to": "BB",
10      "departureTime": "2018-03-22-19",
11      "arrivalTime": "2018-03-24-19",
12      "description": "EE",
13      "plane": {
14        "capacity": 100,
15        "model": "HH",
16        "manufacturer": "II",
17        "yearOfManufacture": 1997
18      },
19      "seatsLeft": 94
20    },
21    {
22      "flightNumber": "indigo",
23      "price": 120,
24      "from": "AA",
25      "to": "BB",
26      "departureTime": "2017-03-22-19",
27      "arrivalTime": "2017-03-24-19",
28      "description": "EE",
29      "plane": {
30        "capacity": 100,
31        "model": "HH",
32      }
33    }
34  ]
35 }

```

## b) Failure Case

### Method: GET

**URI:** <https://54.183.193.226:8443/reservation/a4ffb9e8-304d-40de-a98b-46079edc6b0d12>

```

1 [ {
2   "response": {
3     "code": 404,
4     "msg": "Reservation with number a4ffb9e8-304d-40de-a98b-46079edc6b0d12 does not exist"
5   }
6 }

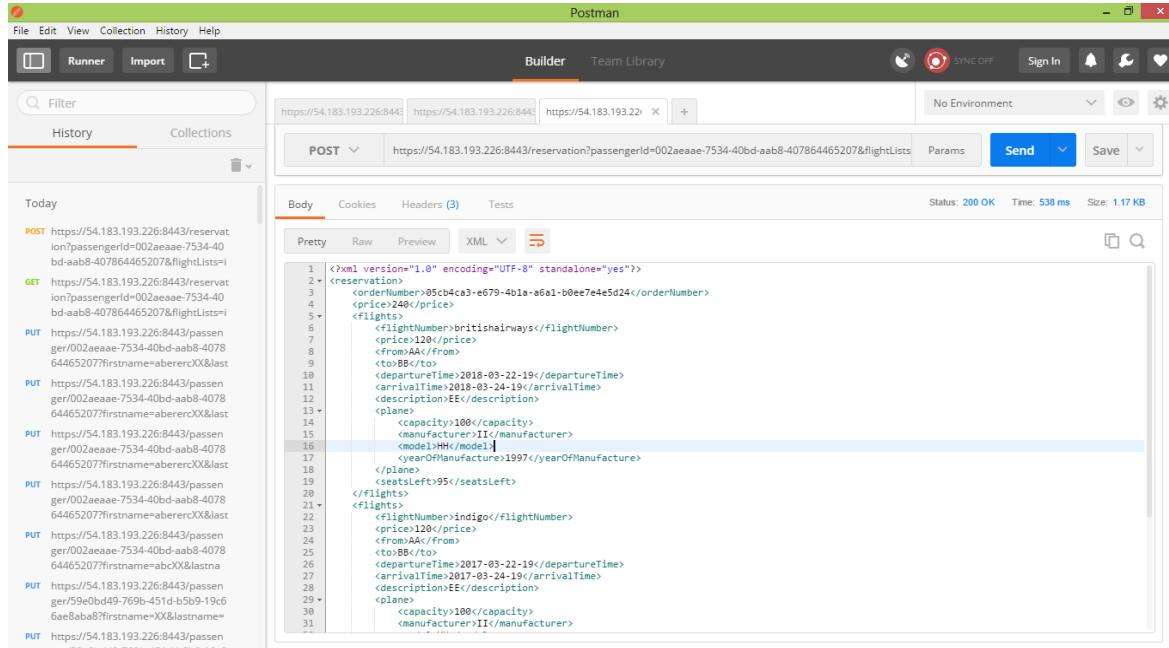
```

## (7) Make a reservation

### a) Success Case:

**Method:** POST

**URI:** <https://54.183.193.226:8443/reservation?passengerId=002aeaae-7534-40bd-aab8-407864465207&flightLists=indigo,britishairways>



The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Collection, History, Help, and a sign-in button. The main workspace has tabs for Runner, Import, and a collection named "History". The current tab is "Builder". The URL in the address bar is https://54.183.193.226:8443/reservation?passengerId=002aeaae-7534-40bd-aab8-407864465207&flightLists=indigo,britishairways. The "Body" tab is selected, showing a POST request with the XML payload below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reservation>
    <orderNumber>05cb4ca-e679-4b1a-a6a1-b0ee7e5d24</orderNumber>
    <price>240</price>
    <flightLists>
        <flightNumber>britishairways</flightNumber>
        <price>120</price>
        <from>AA</from>
        <to>BB</to>
        <departureTime>2018-03-22-19</departureTime>
        <arrivalTime>2018-03-24-19</arrivalTime>
        <description>E</description>
        <plane>
            <capacity>100</capacity>
            <manufacturer>II</manufacturer>
            <model>HH</model>
            <yearOfManufacture>1997</yearOfManufacture>
        </plane>
        <seatsLeft>95</seatsLeft>
    </flightLists>
    <flights>
        <flightNumber>indigo</flightNumber>
        <price>120</price>
        <from>AA</from>
        <to>BB</to>
        <departureTime>2017-03-22-19</departureTime>
        <arrivalTime>2017-03-24-19</arrivalTime>
        <description>E</description>
        <plane>
            <capacity>100</capacity>
            <manufacturer>II</manufacturer>
        </plane>
    </flights>
</reservation>
```

The status bar at the bottom indicates a 200 OK status, 538 ms time, and 1.17 KB size.

### b) Failure Case

**Method:** POST

**URI:** <https://54.183.193.226:8443/reservation?passengerId=002aeaae-7534-40bd-aab8-407864465207&flightLists=indigo,britishairways>

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Collection, History, Help, Runner, Import, and a search bar. The main area is titled "Builder" with tabs for "Builder", "Team Library", and "Sign In". There are several tabs at the top labeled https://54.183.193.2, https://54.183.193.2, https://54.183.193.2, https://54.183.193.2, https://54.183.193.2, and a "+" button. A "No Environment" dropdown is shown. On the left, a sidebar lists "History" and "Collections", with a "Save to collection" button. The main content area shows a "POST" request to https://54.183.193.226:8443/reservation?passengerId=60c30e07-0b58-4ff5-8b06-e8fa1ee41&flightLists=bir. The status bar indicates Status: 400 Bad Request, Time: 160 ms, and Size: 299 B. The "Body" tab is selected, showing the XML response:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <badRequest>
3   <response>
4     <code>400</code>
5     <msg>TimeOverlap is not allowed!</msg>
6   </response>
7 </badRequest>

```

## (8) Update a reservation

a) Success Case:

**Method:** POST

**URI:** <https://54.183.193.226:8443/reservation/05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24?flightsAdded=emirates&flightsRemoved=indigo>

Postman screenshot showing a successful POST request to <https://54.183.193.226:8443/reservation/05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24?flightsAdded=emirates&flightsRemoved=indigo>. The response status is 200 OK, Time: 72 ms, Size: 842 B.

```

1  {
2     "orderNumber": "05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24",
3     "price": 240,
4     "flights": [
5         {
6             "flightNumber": "britishairways",
7             "price": 120,
8             "from": "AA",
9             "to": "BB",
10            "departureTime": "2018-03-22-19",
11            "arrivalTime": "2018-03-24-19",
12            "description": "EE",
13            "plane": {
14                "capacity": 100,
15                "model": "HH",
16                "manufacturer": "II",
17                "yearOfManufacture": 1997
18            },
19            "seatsLeft": 94
20        },
21        {
22            "flightNumber": "emirates",
23            "price": 120,
24            "from": "AA",
25            "to": "BB",
26            "departureTime": "2019-03-22-19",
27            "arrivalTime": "2019-03-24-19",
28            "description": "EE",
29            "plane": {
30                "capacity": 100,
31                "model": "HH",
32            }
33        }
34    ]
35 }

```

## b) Failure Case

### Method: POST

**URI:** <https://54.183.193.226:8443/reservation/05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24?flightsAdded=emirates&flightsRemoved=indigo>

Postman screenshot showing a failed POST request to <https://54.183.193.226:8443/reservation/05cb4ca3-e679-4b1a-a6a1-b0ee7e4e5d24?flightsAdded=emirates&flightsRemoved=indigo>. The response status is 400 Bad Request, Time: 153 ms, Size: 220 B.

```

1  {
2     "response": {
3         "code": 400,
4         "msg": "Time-Overlap is not allowed!"
5     }
6 }

```

## (9) Search for reservations

### a) Success Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/reservation?passengerId=9fc578e6-10ea-4e2d-90a7-dfe8f22ba04e>

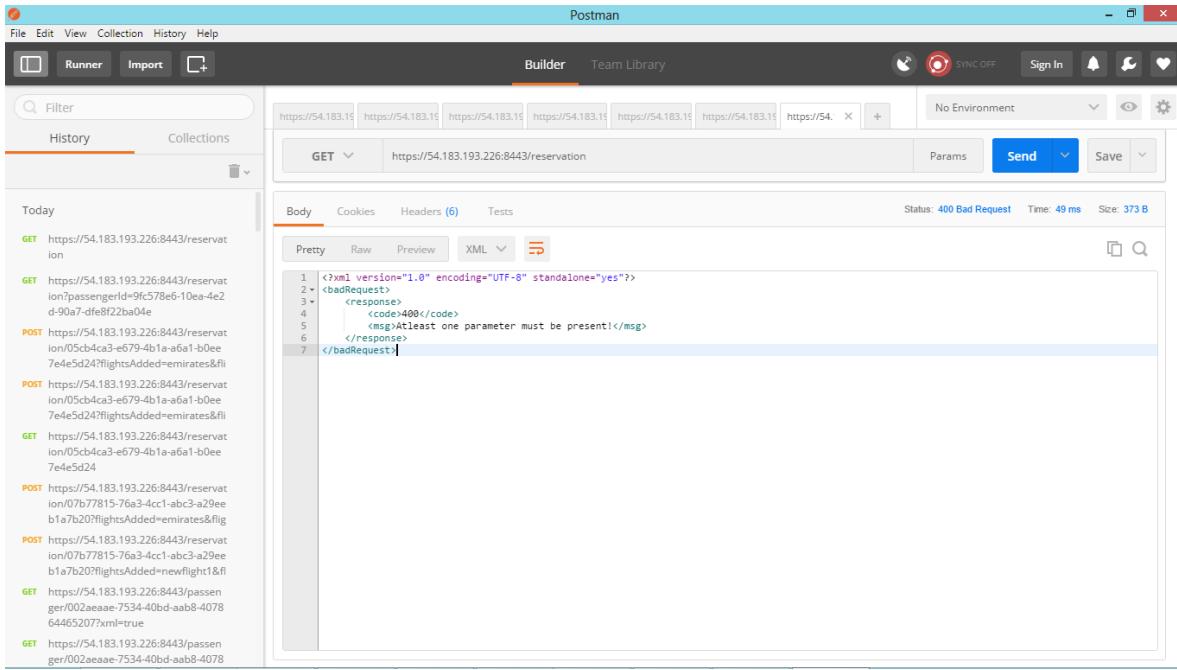
The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Collection, History, Help, Runner, Import, and a sign-in section. The main workspace has tabs for various URLs, with the current tab being "https://54.183.193.226:8443/reservation?passengerId=9fc578e6-10ea-4e2d-90a7-dfe8f22ba04e". Below the tabs, there are sections for Body, Cookies, Headers (5), and Tests. The Body section is set to "Pretty" view, displaying an XML response. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<reservations>
    <reservation>
        <orderNumber>07b77815-76a3-4cc1-abc3-a29eeb1a7b20</orderNumber>
        <price>240</price>
        <flights>
            <flightNumber>britishairways</flightNumber>
            <price>120</price>
            <from>AA</from>
            <to>BB</to>
            <departureTime>2018-03-22-19</departureTime>
            <arrivalTime>2018-03-24-19</arrivalTime>
            <description>EE</description>
            <plane>
                <capacity>100</capacity>
                <manufacturer>II</manufacturer>
                <model>HK</model>
                <yearOfManufacture>1997</yearOfManufacture>
            </plane>
            <seatsLeft>94</seatsLeft>
        </flights>
        <flights>
            <flightNumber>emirates</flightNumber>
            <price>120</price>
            <from>AA</from>
            <to>BB</to>
            <departureTime>2019-03-22-19</departureTime>
            <arrivalTime>2019-03-24-19</arrivalTime>
            <description>EE</description>
            <plane>
                <capacity>100</capacity>
            </plane>
        </flights>
    </reservation>
</reservations>
```

### b) Failure Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/reservation>



## (10) Cancel a reservation

a) Success Case

**Method:** DELETE

**URI:** <https://54.183.193.226:8443/reservation/ca924b8d-3a20-4fc7-9f37-d2b28ab0fd4c>

Postman interface showing a successful DELETE request to <https://54.183.193.226:8443/reservation/1234>. The response body is:

```

1 ↴ [ {
  2   "response": {
  3     "code": 200,
  4     "msg": "Reservation with number ca924b8d-3a20-4fc7-9f37-d2b28ab0fd4c is deleted successfully"
  5   }
  6 }]

```

## b) Failure Case

### Method: DELETE

**URI:** <https://54.183.193.226:8443/reservation/1234>

Postman interface showing a failed GET request to <https://54.183.193.226:8443/reservation/1234>. The response body is:

```

1 ↴ [ {
  2   "response": {
  3     "code": 404,
  4     "msg": "Reservation with number 1234 does not exist"
  5   }
  6 }]

```

## (11) Get a flight back as JSON

### a) Success Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/flight/britishairways>

```
2   "flightNumber": "britishairways",
3   "price": 120,
4   "from": "AA",
5   "to": "BB",
6   "departureTime": "2018-03-22T19:00:00Z",
7   "arrivalTime": "2018-03-24T19:00:00Z",
8   "decoration": "EE",
9   "plane": {
10     "capacity": 100,
11     "model": "MH",
12     "manufacturer": "II",
13     "yearOfManufacture": 1997
14   },
15   "seatsLeft": 94,
16   "passenger": [
17     {
18       "id": "98c578e6-10ea-4e2d-90a7-dfe8f22ba04e",
19       "firstname": "Michael",
20       "lastname": "Scofield",
21       "age": 33,
22       "gender": "male",
23       "phone": "1234567890"
24     },
25     {
26       "id": "002eeaae-7534-40bd-aab8-407864465207",
27       "firstname": "Abbercrombie",
28       "lastname": "XXX",
29       "age": 13,
30       "gender": "female",
31       "phone": "1234567321"
32     }
33   ]
34 }
```

### b) Failure Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/flight/birtishairways12>

Postman interface showing a failed GET request to <https://54.183.193.226:8443/flight/birtshairways12>. The response status is 404 Not Found, and the body contains:

```

1 | [
2 |   "response": {
3 |     "code": 404,
4 |     "msg": "Sorry, the requested flight with number birtshairways12 does not exist"
5 |   }
6 |

```

## 12) Get a Flight back as XML

### a) Success Case

**Method: GET**

**URI:** <https://54.183.193.226:8443/flight/indigo?xml=true>

Postman interface showing a successful GET request to <https://54.183.193.226:8443/flight/indigo?xml=true>. The response status is 200 OK, and the body contains XML:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flight>
  <flightNumber>indigo</flightNumber>
  <price>120</price>
  <from>sanjose</from>
  <to>newyork</to>
  <departureTime>2017-03-22-19</departureTime>
  <arrivalTime>2017-03-24-19</arrivalTime>
  <description>Indigo happy journey</description>
  <plane>
    <capacity>100</capacity>
    <manufacturer>indigo</manufacturer>
    <model>OGI01</model>
    <yearOfManufacture>2000</yearOfManufacture>
  </plane>
  <seatLeft>91</seatLeft>
  <passenger>
    <age>11</age>
    <firstname>XX</firstname>
    <gender>Female</gender>
    <id>ecef17d7-a835-4f1d-acbc-3e6466103415</id>
    <lastName>YY</lastName>
    <phone>120</phone>
  </passenger>
  <passenger>
    <age>33</age>
    <firstname>Michael</firstname>
    <gender>Male</gender>
    <id>af0f07b-58d4-4006-89d3-2b7a1831c9da</id>
    <lastName>Scofield</lastName>
    <phone>123021</phone>
  </passenger>
</flight>

```

## b) Failure Case

**Method:** GET

**URI:** <https://54.183.193.226:8443/flight/birtishairways12?xml=true>

The screenshot shows the Postman interface with a failed GET request. The URL in the header is <https://54.183.193.226:8443/flight/birtishairways12?xml=true>. The response status is 404 Not Found. The XML body contains the following error message:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<badRequest>
<response>
<code>404</code>
<msg>Sorry, the requested flight with number birtishairways12 does not exist</msg>
</response>
</badRequest>
```

## 13) Create or update a flight:

a) Success Case

**Method:** POST

**URI:**

<https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&arrivalTime=2025-03-23-19&description=EE&capacity=1&model=HH&manufacturer=II&yearOfManufacture=1997>

The screenshot shows the Postman Builder interface. The URL in the header is <https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&arrivalTime=2025-03-23-19&description=EE&capacity=0&model=HH&manufacturer=II&yearOfManufacture=1997>. The request type is POST. The body tab contains the following XML payload:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flight>
<flightNumber>newFlight1</flightNumber>
<price>120</price>
<from>AA</from>
<to>BB</to>
<departureTime>2025-03-22-19</departureTime>
<arrivalTime>2025-03-23-19</arrivalTime>
<description>EE</description>
<plane>
<capacity>1</capacity>
<manufacturer>II</manufacturer>
<model>HH</model>
<yearOfManufacture>1997</yearOfManufacture>
</plane>
<seatsLeft>1</seatsLeft>
</flight>

```

## b) Failure Case

### Method: GET

**URI:** <https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&arrivalTime=2025-03-23-19&description=EE&capacity=0&model=HH&manufacturer=II&yearOfManufacture=1997>

**POST** https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&

**GET** https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&

**GET** https://54.183.193.226:8443/flight/newflight1?price=120&from=AA&to=BB&departureTime=2025-03-22-19&

**POST** https://54.183.193.226:8443/reservation?passengerId=d7e54593-a66f-4e47-b6fd-73fde4b7f8a&flightLists=ne

**GET** https://54.183.193.226:8443/reservation?passengerId=d7e54593-a66f-4e47-b6fd-73fde4b7f8a&flightLists=ne

**POST** https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=38&gender=Male&photo

**POST** https://54.183.193.226:8443/passenger?firstname=Michael&lastname=Scofield&age=38&gender=Male&photo

**GET** https://54.183.193.226:8443/reservat

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <badRequest>
3   <response>
4     <code>400</code>
5     <msg>New capacity can't be less than the reservations for that flight!</msg>
6   </response>
7 </badRequest>

```

## 14) Delete a flight

### a) Success case

**Method:** Delete

**URI:** <https://54.183.193.226:8443/airline/indiairways>

**DEL** https://54.183.193.226:8443/airline/indiairways

**GET** https://54.183.193.226:8443/airline/indiairways

**POST** https://54.183.193.226:8443/flight/indiairways?price=1200&from=vegas&to=bangalore&departureTime=2017-03-22T12:00:00Z

**POST** https://54.183.193.226:8443/reservation?passengerId=ecc171d7-a835-4f1d-abc3-3e4661034158&flightLists=ne

**POST** https://54.183.193.226:8443/reservation?passengerId=ecc171d7-a835-4f1d-abc3-3e4661034158&flightLists=ne

**POST** https://54.183.193.226:8443/reservation/a4fffb9e8-304d-40de-a98b-46079edc6b0d?flightsRemoved=indiainair

**POST** https://54.183.193.226:8443/reservation/a4fffb9e8-304d-40de-a98b-46079edc6b0d?flightsRemoved=indiainair

**POST** https://54.183.193.226:8443/reservation/a4fffb9e8-304d-40de-a98b-46079edc6b0d?flightsRemoved=indiainair

**POST** https://54.183.193.226:8443/reservation/a4fffb9e8-304d-40de-a98b-46079edc6b0d?flightsRemoved=indiainair

```

1 {
2   "code": 200,
3   "msg": "Flight with number indiairways is deleted successfully"
4 }

```

## b) Failure Case

### Method: DELETE

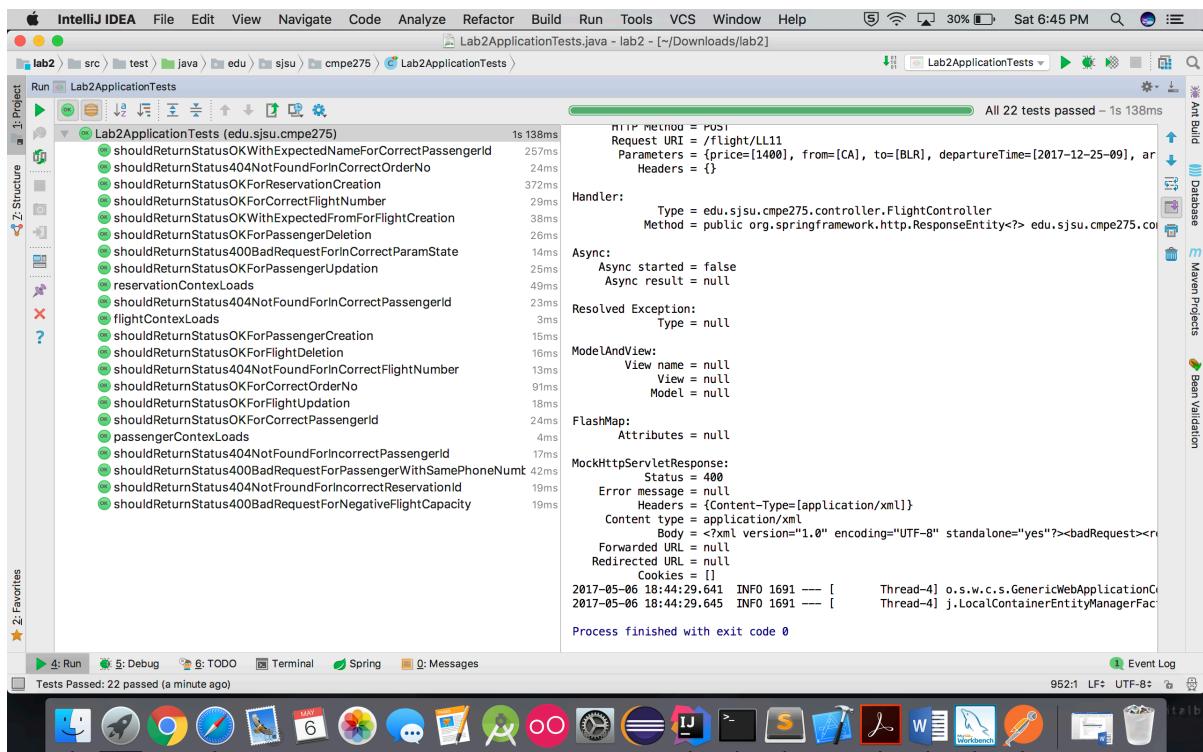
URI: <https://54.183.193.226:8443/airline/indiairways12>

The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. Below it, the URL <https://54.183.193.226:8443/airline/indiairways12> is entered, and the method is set to 'DELETE'. The 'Authorization' tab is selected in the request configuration. The response body is displayed in JSON format:

```
1 | {  
2 |   "response": {  
3 |     "code": 404,  
4 |     "msg": "Flight with number indiairways12 does not exist"  
5 |   }  
6 | }
```

The status bar at the bottom indicates: Status: 404 Not Found, Time: 138 ms, Size: 282 B.

## Unit Testing:



## References,

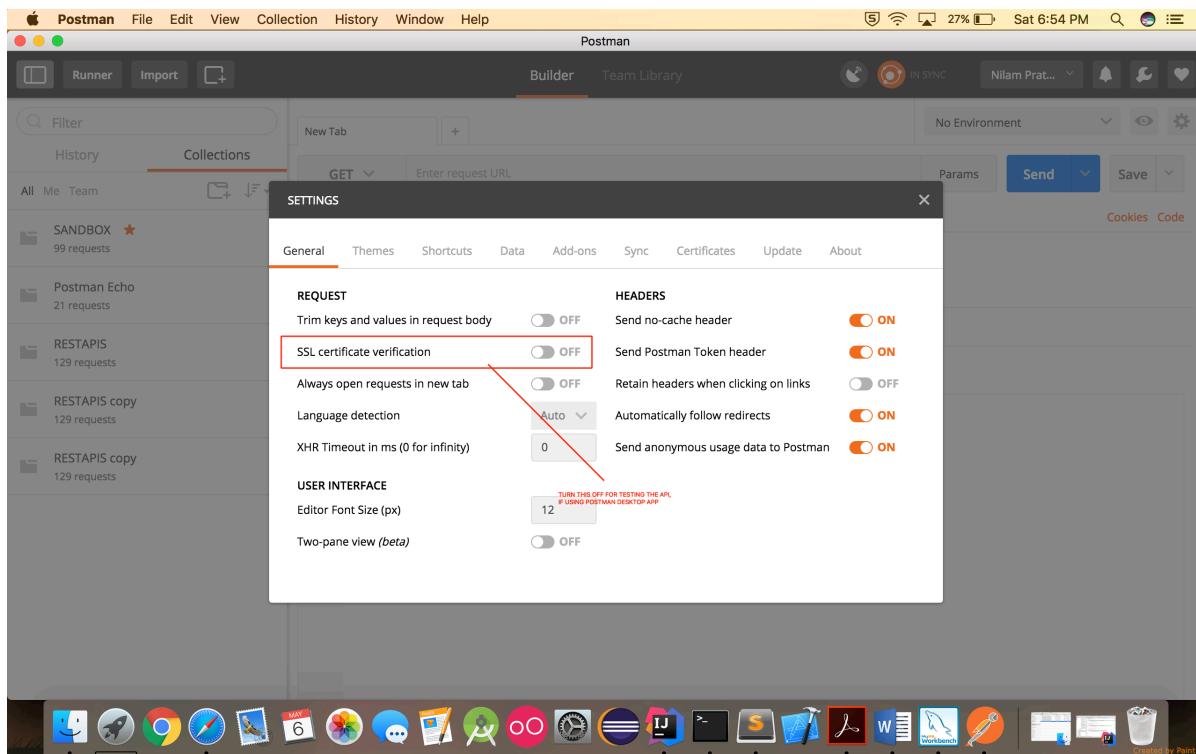
[stackoverflow.com/questions/24497809/compare-intervals-jodatime-in-a-list-for-overlap](https://stackoverflow.com/questions/24497809/compare-intervals-jodatime-in-a-list-for-overlap)

[stackoverflow.com/questions/16986888/sort-joda-time-intervals](https://stackoverflow.com/questions/16986888/sort-joda-time-intervals)

[drissamri.be/blog/java/enable-https-in-spring-boot/](http://drissamri.be/blog/java/enable-https-in-spring-boot/)

## Additional Instructions,

If using Postman desktop application, turn off the SSL certificate verification to see the API responses, as shown below.



If the endpoints are tried from any browser,  
Please accept the self-signed certificate. It is safe to proceed.

HTTPS service is exposed at port 8443.

HTTP at port 8080.

### **Known issue,**

HTTP post calls get converted to HTTP get calls

[stackoverflow.com/questions/43689534/unable-to-redirect-http-post-calls-after-enabling-https-in-spring-boot](https://stackoverflow.com/questions/43689534/unable-to-redirect-http-post-calls-after-enabling-https-in-spring-boot)

[sjsu.instructure.com/courses/1227732/discussion\\_topics/3225193](https://sjsu.instructure.com/courses/1227732/discussion_topics/3225193)