

Testcases Executed CRUD operations and Error scenarios

TC ID	Test Case	Steps	Expected Result	Actual Result	Status
TC01	Generate valid token	Send POST /auth with valid credentials	200 OK ,token returned in the response	200 OK and token returned in the response	Pass
TC02	Invalid login	Send POST /auth with wrong password	401 Bad request "Bad credentials"	200 OK "Bad credentials"	Fail
TC03	Update Booking (PUT) with valid Auth	Send PUT with Id in url and updated body	200 OK Response body contains the same data	200 OK Response body contains the same data	Pass
TC04					
TC05	Update booking without token	Send PUT /booking/1 without token	403 Forbidden	403 Forbidden	Pass
TC06	Update booking With Invalid Token	Send PUT /auth with wrong password	403 Forbidden	403 Forbidden	Pass
TC07	Create booking	Send POST with booking details in the body	200 OK Response body contains the same data	200 OK Response body contains the same data	Pass
TC08	Retrive booking using ID	Send GET request with ID in the url	200 OK response body containg the booking	200 OK response body containig the booking	Pass
TC09	Fetch non-Existent Booking ID	Send GET request with invalid ID eg 6535267388883	404 error, not found response.	404 error, not found response.	Pass

TC ID	Test Case	Steps	Expected Result	Actual Result	Status
TC10	Update Booking with no Auth	Send PUT request with No Auth.	Status code 403 response forbidden	Status code 403 response forbidden	Pass
TC11	Delete Booking with vralid Auth	Send Delete request with correct Auth and ID in url	200 OK response body: created	Status code : 201 response body :created	Fail
TC12	Delete booking with invalid /No auth	Send delete request with no auth plu ID in the url	Status code: 403 response :forbidden	Status code :403 response :forbidden	Pass
TC13	Retrieve Deleted Booking	Send GET request with deleted ID in the url	404not found error, response body, not found	404, not found error, response body not found	Pass
TC14	Missing required fields	Send POST request with only first name	Code 400 Bad request response	Code 400 Bad request response	Pass
TC15	[Data validation]Create booking with invalid date	Post request with booking created with checkout is earler than check in eg checkin:2018-01-01 checkout:2017-01-01	Error 400 bad request response body Bad request	200 OK Response body contains the same data	Fail
TC16	Optional filters : firstname. Lastname, checkin, checkout.	Get request with either fields as filters	200 OK with a response of an array of objects with all the booking IDs search matching the search criteria	200 OK with a response of an array of objects with all the booking IDs matching the criteria	Pass

Successful tests Screenshots

1. POST-Generated valid token

The screenshot shows the Postman interface with a successful API call. The URL is `https://restful-booker.herokuapp.com/auth`. The request method is `POST`. The body contains the following JSON:

```
1 {  
2   "username": "admin",  
3   "password": "password123"  
4 }
```

The response status is `200 OK` with a response time of `1.28 s`, a size of `770 B`, and a token value of `15b43be13e5f25d`.

2. POST-Invalid login

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, History, Flows, and Files (BETA). The main workspace shows a collection named "Shelly W's Workspace" with several items under "Triply technical test". A specific POST request titled "Auth - invalid token credentials" is selected. The request details show a POST method to the URL <https://restful-booker.herokuapp.com/auth?password>. The "Body" tab is active, showing a raw JSON payload: { "username": "adminj", "password": "password123zz" }. The response pane shows a 200 OK status with a response body containing the string "Bad credentials".

3. POST-create booking

This screenshot shows a similar setup in Postman. A POST request titled "Booking - CreateBooking" is selected, targeting the URL <https://restful-booker.herokuapp.com/booking>. The "Body" tab contains a raw JSON payload with fields like firstname, lastname, totalprice, depositpaid, bookingdates (with checkin and checkout dates), and additionalneeds. The response shows a 200 OK status with a booking ID of 5847.

4. GET - Retrieve booking using ID

The screenshot shows the Postman interface with a collection named "Triply technical test / Booking". A GET request is made to `https://restful-booker.herokuapp.com/booking/:id` with a parameter `id=300`. The response status is 200 OK, and the JSON body contains the following booking details:

```

1 {
2   "firstname": "John",
3   "lastname": "Smith",
4   "totalprice": 111,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2018-01-01",
8     "checkout": "2019-01-01"
9   },
10  "additionalneeds": "Breakfast"
11 }

```

5. PUT-Update Booking with valid Auth

The screenshot shows the Postman interface with a collection named "Triply technical test / Booking". A PUT request is made to `https://restful-booker.herokuapp.com/booking/:id` with a parameter `id=1830` and a JSON body containing updated booking details:

```

1 {
2   "firstname": "shelly1",
3   "lastname": "update1",
4   "totalprice": 222,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2020-01-01",
8     "checkout": "2020-01-02"
9   },
10  "additionalneeds": "Lunch"
11 }

```

6. PUT-Update booking without token

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, API Network, and a search bar labeled "Search Postman". On the right side of the header are icons for Invite, Settings, and Upgrade.

The main workspace displays a collection named "Triply technical test / Booking - UpdateBooking-No Auth". A PUT request is being made to the URL <https://restful-booker.herokuapp.com/booking/>. The "Params" tab is selected, showing a key-value pair where "id" has a value of "1830".

Below the request, the "Test Results" section shows a single response entry with the status code 403 Forbidden. The response body contains the word "Forbidden".

At the bottom of the interface, there are various toolbars and buttons for Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, and AI.

7. DELETE – Booking with valid Auth

This screenshot shows the same Postman interface as the previous one, but with a successful DELETE request. The collection is now named "Triply technical test / Booking - DeleteBooking".

A DELETE request is being made to the URL <https://restful-booker.herokuapp.com/booking/7>. The "Params" tab is selected, showing a key-value pair where "key" has a value of "value".

The "Test Results" section shows a single response entry with the status code 201 Created. The response body contains the word "Created".

At the bottom of the interface, there are various toolbars and buttons for Cloud View, Find and replace, Console, Terminal, Runner, Start Proxy, Cookies, Vault, Trash, and AI.

8. GET Retrieve Deleted Booking

The screenshot shows the Postman application interface. A collection named "Triply technical test / retrieve a deleted booking" is selected. A single test case is defined with a GET request to the URL `https://restful-booker.herokuapp.com/booking/6`. The request includes a query parameter "id" with the value "6". The response status is 404 Not Found, indicating the resource was not found.

Bug report

1. Bug ID: 001

Title: POST Auth Create-token API returns incorrect status code

Environment:

- API: <https://restful-booker.herokuapp.com/auth>
- Tool: Postman
- Method: POST

Severity: Medium

Priority: Medium

Steps to Reproduce:

1. Access the Auth API
2. Enter Json payload with wrong credentials username: "Adminzz" password:"password123zzz"
3. Send the POST request

Expected Result:

API should return 401 Reason: “Bad credentials”

Actual Result:

API returns 200 Reason “Bad credentials”

Screenshot:

The screenshot shows the Postman application interface. A POST request is being made to <https://restful-booker.herokuapp.com/auth/password>. The request body is set to JSON and contains the following data:

```
1 {
2   "username": "admin",
3   "password": "password123zz"
4 }
```

The response status is 200 OK, and the response body is:

```
1 {
2   "reason": "Bad credentials"
3 }
```

2. Bug ID: 002

Title: DELETE API returns incorrect status code

Environment:

- API: <https://restful-booker.herokuapp.com/booking/7>
- Tool: Postman
- Method: DELETE

Severity: Medium

Priority: Medium

Steps to Reproduce:

1. Access the DELETE API
2. Get a valid ID and append to the API

Expected Result:

API should return 200 OK response

Actual Result:

API returns 201 Created successfully which is misleading

Screenshot:

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, API Network, and a search bar. On the right side, there are buttons for Invites, Upgrade, and environment selection. The main workspace shows a collection named "Triply technical test" with a single item "Booking - DeleteBooking". A "DELETE" request is selected with the URL "https://restful-booker.herokuapp.com/booking/7". The "Params" tab is active, showing a key-value pair "Key": "Value". Below the request, the response section displays a status of "201 Created" with a timestamp of "1.16 s" and a size of "747 B". The response body contains the word "Created". The bottom of the screen shows the terminal window with the command "shellsmithwanjugu@Shellsmiths-MacBook-Air ~ % []".

3. Bug ID: 003

Title: POST Create booking API accepts onvalid date data

Environment:

- API: <https://restful-booker.herokuapp.com/booking>
- Tool: Postman
- Method: POST

Severity: Medium

Priority: high

Steps to Reproduce:

3. Access the POST create booking API
4. Enter Json payload with correct data apart from date .booking created with checkout is earler than check in eg checkin:2018-01-01 checkout:2017-01-01

Expected Result:

API should return 400 Bad request since the date is invalid

Actual Result:

API returns 200 OK and booking is created with invalid date

Screenshot:

The screenshot shows the Postman application interface. A POST request is made to the URL <https://restful-booker.herokuapp.com/booking>. The request body contains the following JSON data:

```
1 {
2     "firstname" : "newbooking",
3     "lastname" : "booking",
4     "totalprice" : 111,
5     "depositpaid" : true,
6     "bookingdates" : [
7         {
8             "checkin" : "2018-01-01",
9             "checkout" : "2019-01-01"
10        },
11        "additionalneeds" : "Breakfast"
12    ]
13}
14 }
```

The response status is 200 OK, and the response body is identical to the request body, indicating that the booking was successfully created despite the invalid dates.