

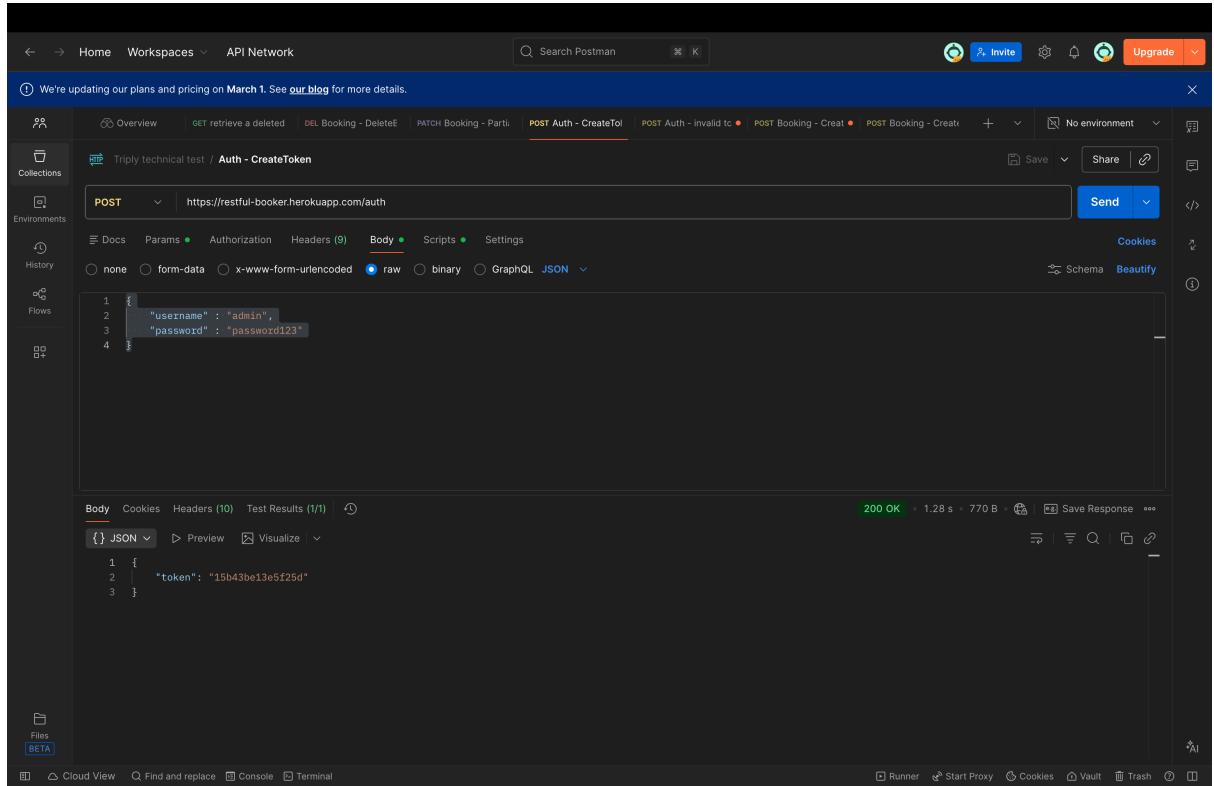
Testcases Executed CRUD operations and Error scenarios

TC ID	Test Case	Steps	Expected Result	Actual Result	Status
TC01	Generate valid token	Send POST /auth with valid credentials	200 OK + token returned	200 OK + token returned	Pass
TC02	Invalid login	Send POST /auth with wrong password	401 Bad request +“Bad credentials”	200 OK + “Bad credentials”	Fail
TC03	Update Booking (PUT) + 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
TC10	Update Booking + no Auth	Send PUT request with No Auth.	Status code 403	Status code 403	Pass

TC ID	Test Case	Steps	Expected Result	Actual Result	Status
			response forbidden	response forbidden	
TC11	Delete Booking+valid Auth	Send Delete request with correct Auth and ID in url	200 OK + response body created	201 created + response body created	Fail
TC12	Delete booking +invalid /No auth	Send delete request with no auth plus 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	404 error + not found response	404 error + not found response	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 earlier than check in eg checkin:2018-01-01 checkout:2017-01-01	Erro 400 plus bad request response	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

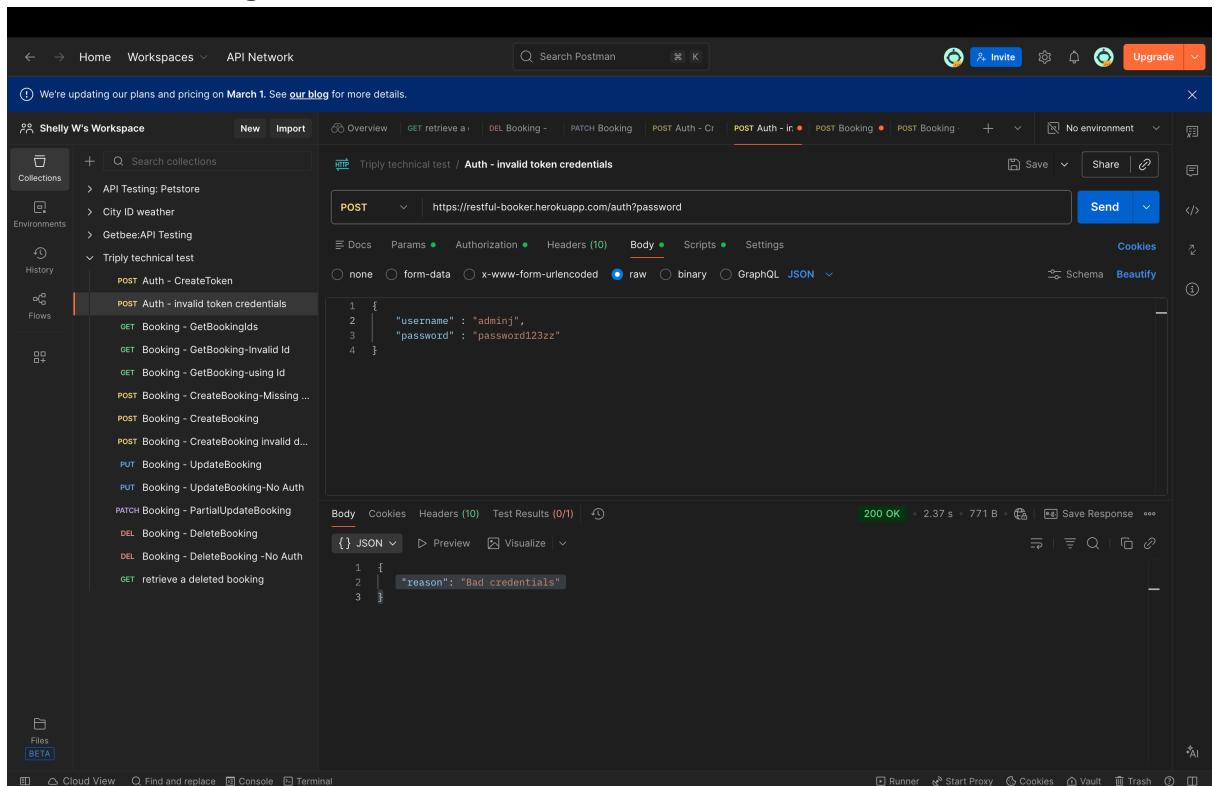


The screenshot shows the Postman interface with a successful API call. The request is a POST to <https://restful-booker.herokuapp.com/auth>. The body contains the following JSON:

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

The response is 200 OK with a token: "15b43be13e5f25d".

2. POST-Invalid login

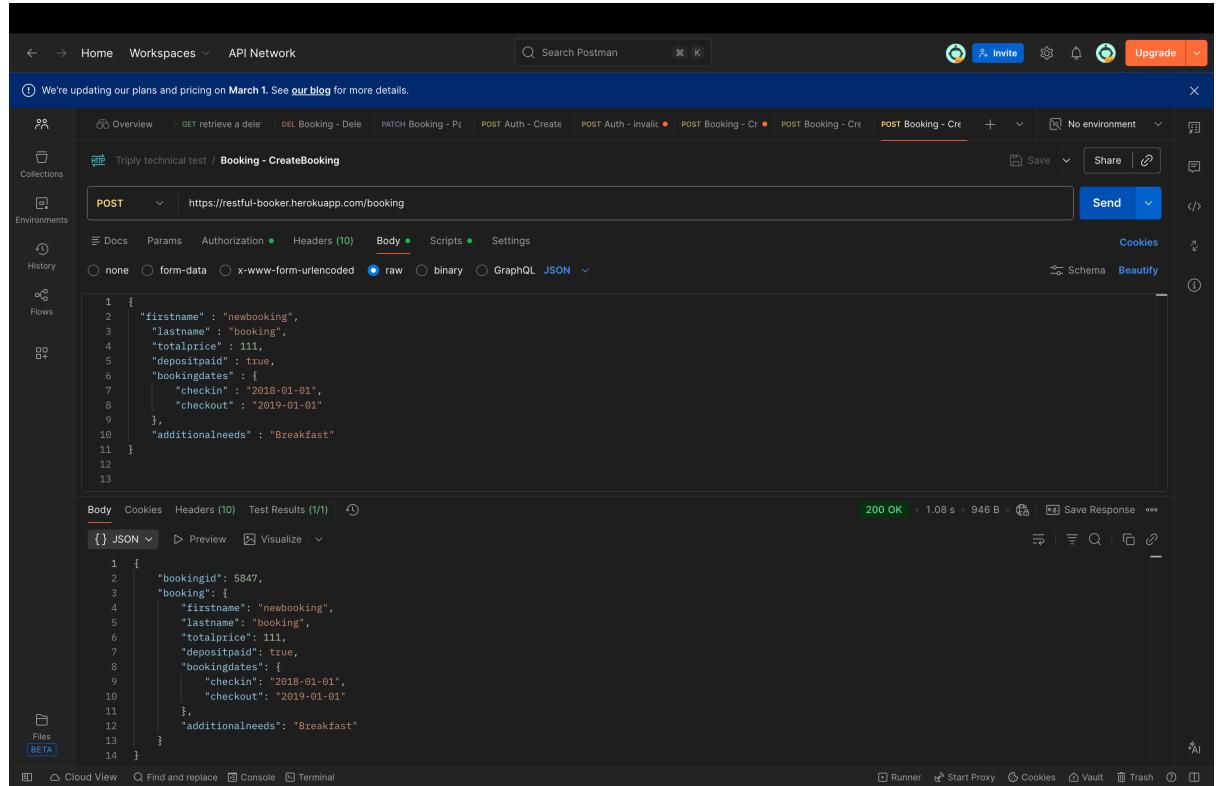


The screenshot shows the Postman interface with an unsuccessful API call. The request is a POST to <https://restful-booker.herokuapp.com/auth?password>. The body contains the following JSON:

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

The response is 200 OK with a reason: "Bad credentials".

3. POST-create booking

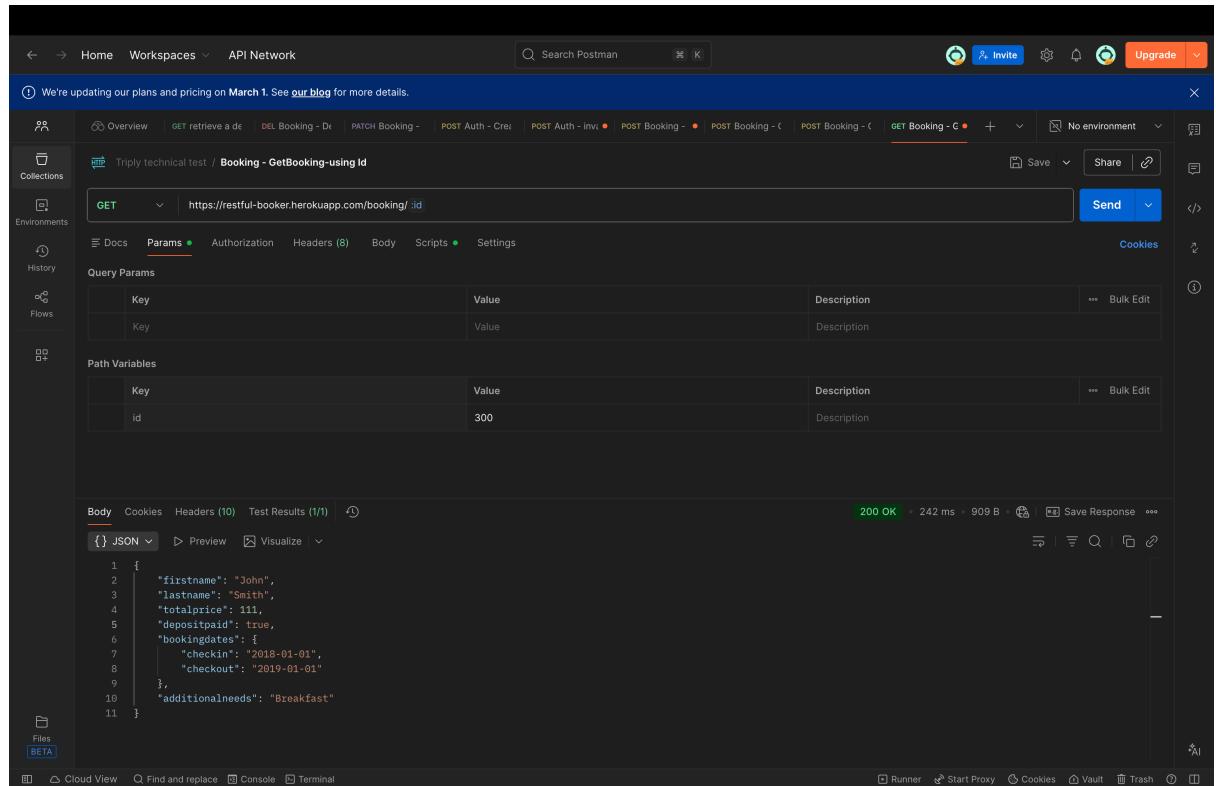


The screenshot shows the Postman interface with a POST request to <https://restful-booker.herokuapp.com/booking>. The request body is a JSON object representing a booking with fields: firstname, lastname, totalprice, depositpaid, bookingdates, checkin, checkout, and additionalneeds. The response is a 200 OK status with a booking ID of 5847.

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

Body Cookies Headers (10) Test Results (1/1) 200 OK + 1.08 s 946 B Save Response

4. GET - Retrieve booking using ID



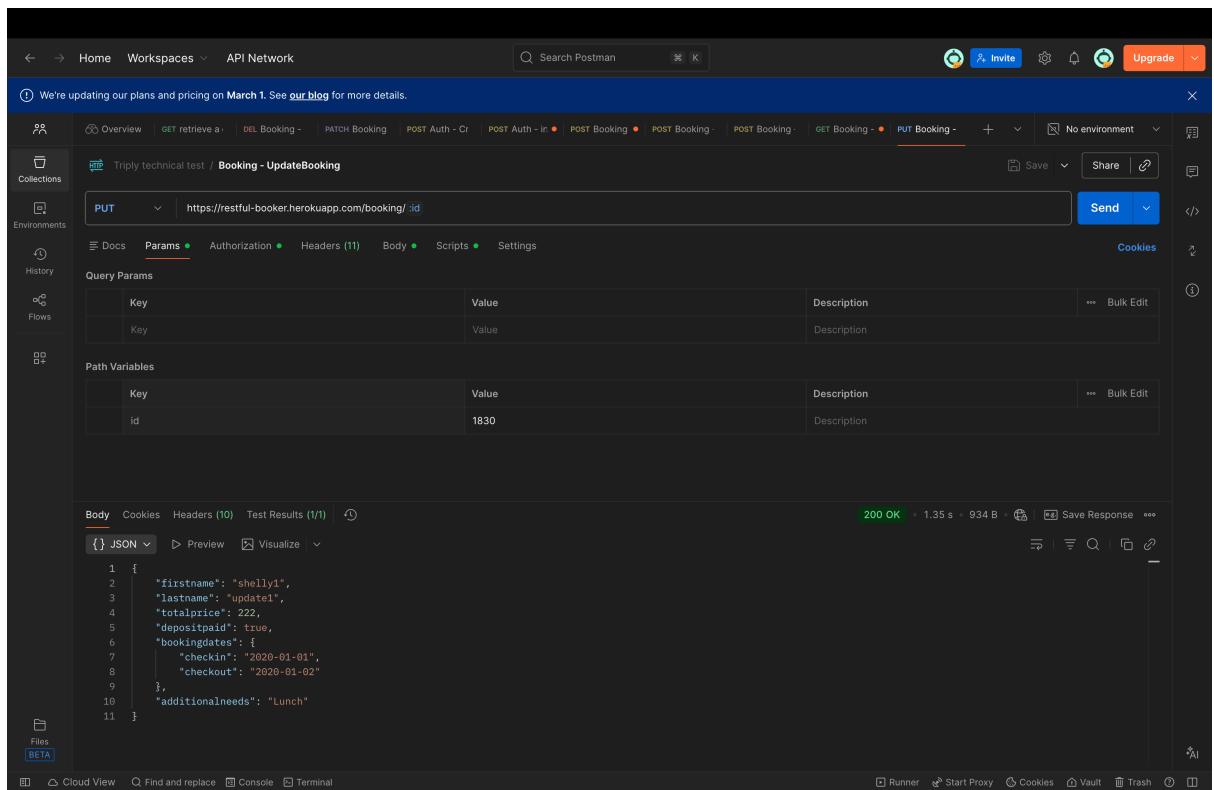
The screenshot shows the Postman interface with a GET request to <https://restful-booker.herokuapp.com/booking/300>. The request includes a query parameter 'id' with the value '300'. The response is a 200 OK status with the same booking object as the POST request.

Key	Value	Description	... Bulk Edit
id	300	Description	

```
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 }
```

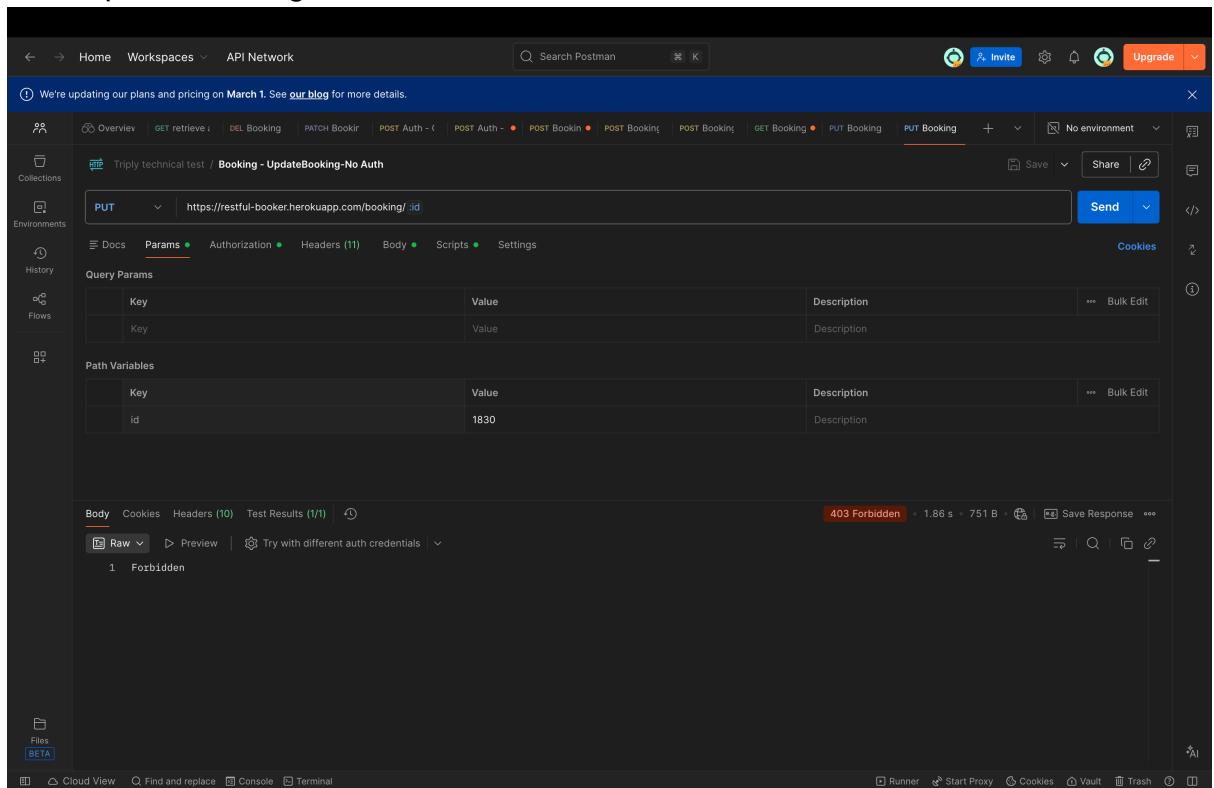
Body Cookies Headers (10) Test Results (1/1) 200 OK + 242 ms 909 B Save Response

5. PUT-Update Booking + valid Auth



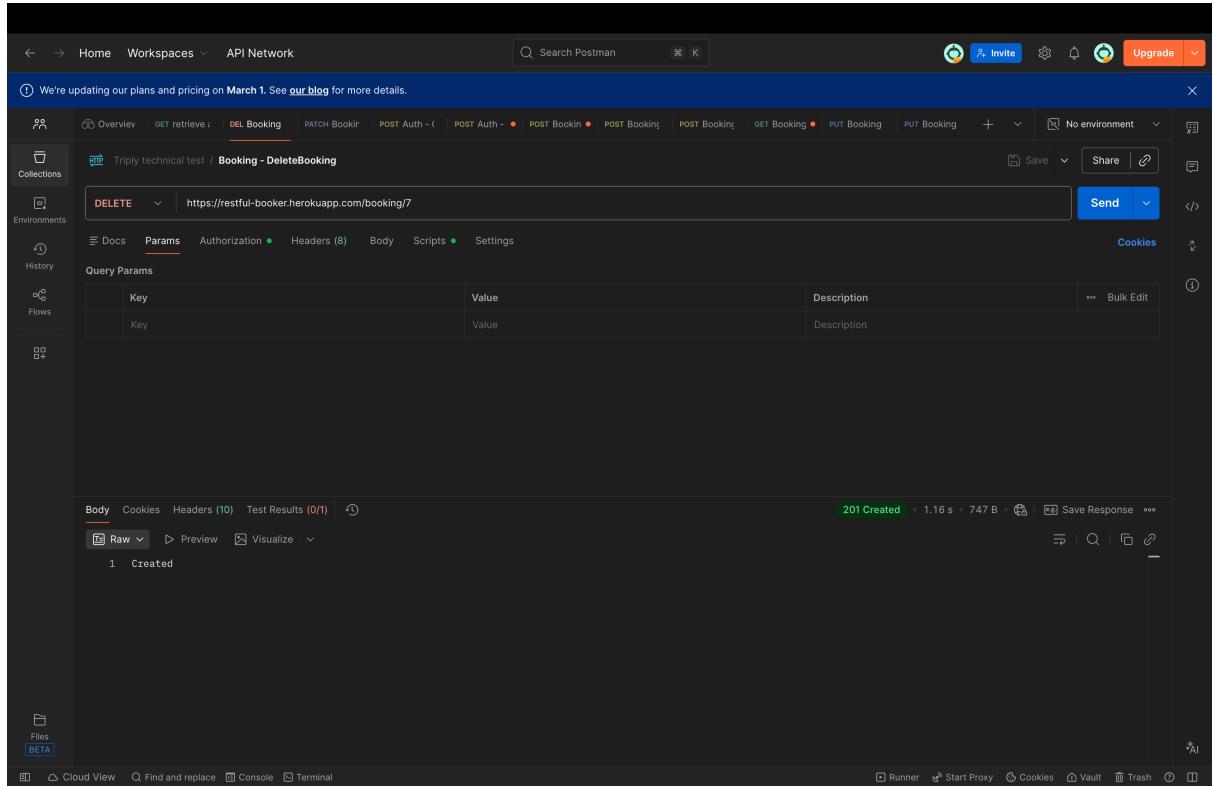
The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', 'Search Postman', and various user icons. A blue banner at the top states: 'We're updating our plans and pricing on March 1. See [our blog](#) for more details.' The main workspace is titled 'Triply technical test / Booking - UpdateBooking'. A 'PUT' request is selected with the URL 'https://restful-booker.herokuapp.com/booking/:id'. The 'Params' tab is active, showing a 'Query Params' table with a single entry: 'Key' (id) and 'Value' (1830). The 'Path Variables' table also has a single entry: 'Key' (id) and 'Value' (1830). The 'Body' tab shows a JSON payload with 11 fields, including 'firstname', 'lastname', 'totalprice', 'depositpaid', 'bookingdates' (with 'checkin' and 'checkout' fields), and 'additionalneeds'. The 'Test Results' tab shows a successful response: '200 OK' with a duration of '1.35 s' and a size of '934 B'. The bottom navigation bar includes 'Cloud View', 'Find and replace', 'Console', 'Terminal', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and 'AI'.

6. PUT-Update booking without token



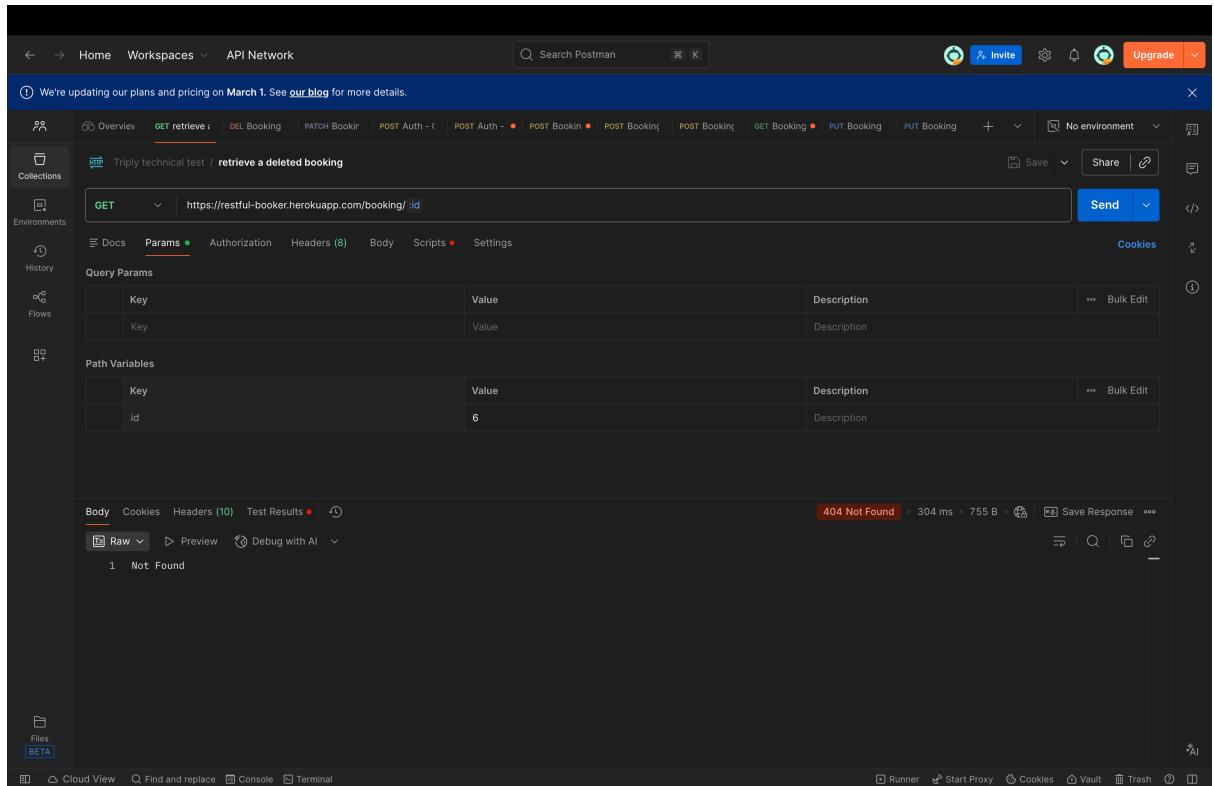
The screenshot shows the Postman application interface, identical to the previous one but with a different test result. The 'Test Results' tab now displays a '403 Forbidden' error with a duration of '1.86 s' and a size of '751 B'. The error message 'Forbidden' is shown in the 'Raw' tab. The bottom navigation bar includes 'Cloud View', 'Find and replace', 'Console', 'Terminal', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and 'AI'.

7. DELETE - Booking+valid Auth



The screenshot shows the Postman interface with a DELETE request to `https://restful-booker.herokuapp.com/booking/7`. The response status is `201 Created` with the message `1 Created`.

8. GET Retrieve Deleted Booking



The screenshot shows the Postman interface with a GET request to `https://restful-booker.herokuapp.com/booking/6`. The response status is `404 Not Found` with the message `1 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:"password123zz"
3. Send the POST request

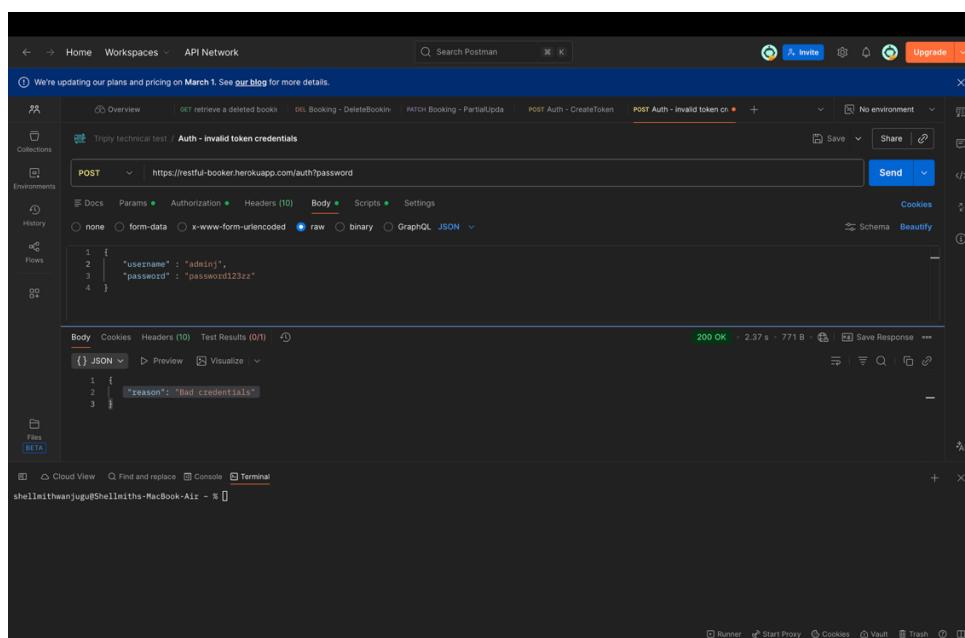
Expected Result:

API should return 401 Reason: "Bad credentials"

Actual Result:

API returns 200 Reason "Bad credentials"

Screenshot:



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

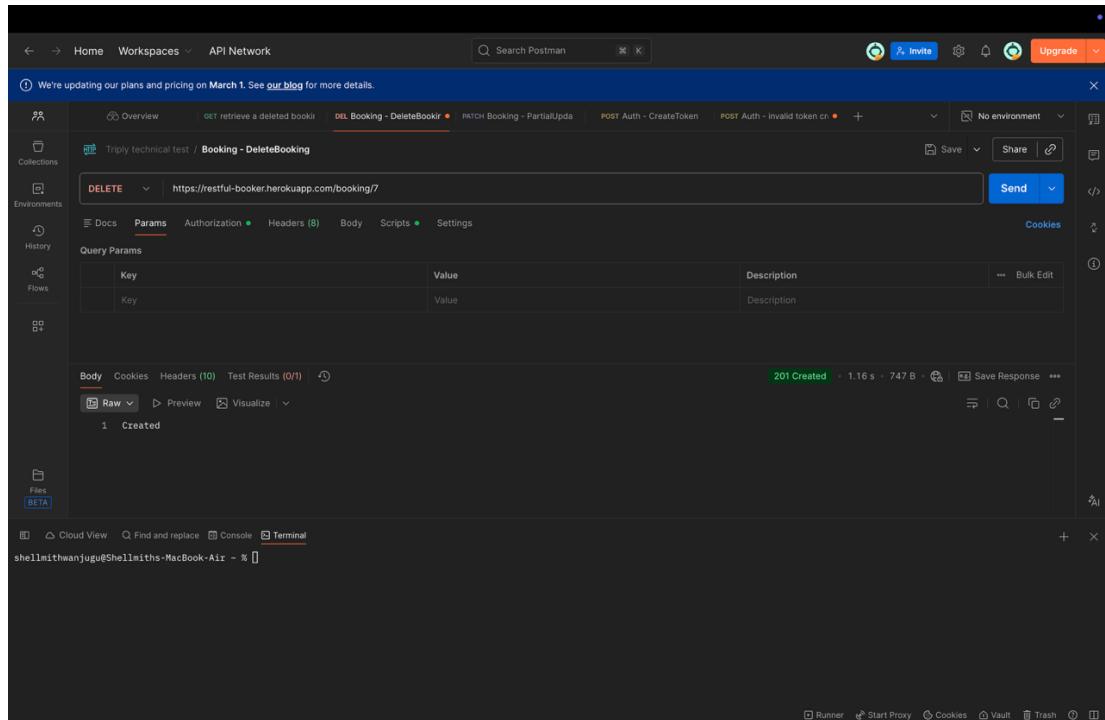
Expected Result:

API should return 200 OK response

Actual Result:

API returns 201 Created successfully which is misleading

Screenshot:



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

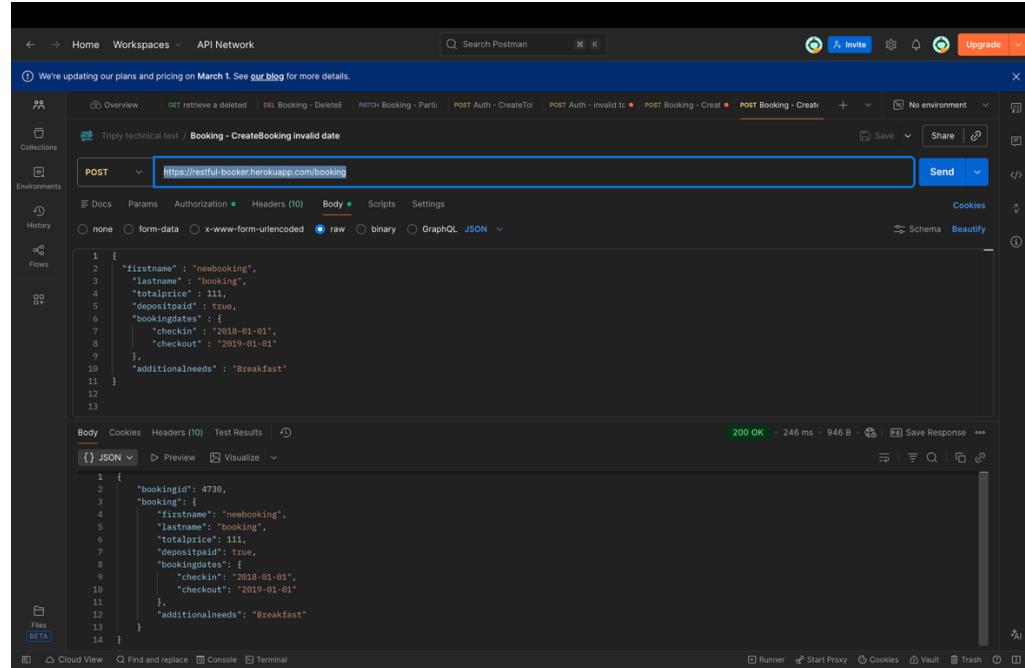
Expected Result:

API should return 400 Bad request since the dte is invalid

Actual Result:

API returns 200 OK and booking is created with invalid date

Screenshot:



The screenshot shows a Postman test named "Booking - CreateBooking invalid date". The request is a POST to <https://restful-booker.herokuapp.com/booking>. The JSON body is:

```
1 {
  "firstname": "newbooking",
  "lastname": "booking",
  "totalprice": 111,
  "depositpaid": true,
  "bookingdates": [
    {
      "checkin": "2018-01-01",
      "checkout": "2019-01-01"
    }
  ],
  "additionalneeds": "Breakfast"
}
```

The response is a 200 OK with a response body:

```
1 {
  "bookingId": 4730,
  "booking": [
    {
      "firstname": "newbooking",
      "lastname": "booking",
      "totalprice": 111,
      "depositpaid": true,
      "bookingdates": [
        {
          "checkin": "2018-01-01",
          "checkout": "2019-01-01"
        }
      ],
      "additionalneeds": "Breakfast"
    }
  ]
}
```