

API Testing using Postman (Screenshots)

Project Name: Booking API Practice

Prepared By: Reeti Thapa

Tools used: Postman

Description:

This document contains screenshots of API testing performed using Postman. The APIs were tested using GET, POST, PATCH, and DELETE methods, including positive and negative scenarios.

1. Get Booking IDs

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Reeti's Workspace' with collections like 'Booking API Practice' (Positive Scenario and Negative Scenario), 'My Collection', and 'My Collection' (fork). The main workspace shows a 'GET' request for 'https://restful-booker.herokuapp.com/booking'. The 'Headers' tab is selected, containing a pre-request script:

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
```

The 'Body' tab shows a JSON response with three booking entries:

```
1 [
2   {
3     "bookingid": 1383
4   },
5   {
6     "bookingid": 626
7   },
8   {
9     "bookingid": 71
10 }
];
```

2. Create Booking

The screenshot shows the Postman application interface. On the left, the 'Reeti's Workspace' sidebar lists collections, environments, history, and flows. The 'Booking API Practice' collection is expanded, showing 'Positive Scenario' and 'Negative Scenario' branches. Under 'Positive Scenario', the 'POST Create Booking' endpoint is selected, showing its details and a sample JSON body:

```
1 {
2   "firstname": "{{fname}}",
3   "lastname": "{{lname}}",
4   "totalprice": 1699,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2018-01-01",
8     "checkout": "2019-01-01"
9   },
10   "additionalneeds": "Breakfast"
11 }
```

The main workspace shows a successful POST request to <https://restful-booker.herokuapp.com/booking>. The response status is 200 OK, time taken is 1.74 s, and size is 939 B. The response body is displayed as JSON:

```
1 {
2   "bookingid": 4111,
3   "booking": {
4     "firstname": "John",
5     "lastname": "Mayor",
6     "totalprice": 1699,
7     "depositpaid": true,
8     "bookingdates": {
9       "checkin": "2018-01-01"
10     }
11   }
12 }
```

This screenshot shows the same Postman session after adding assertions to the 'Post-response' script tab. The 'Positive Scenario' branch is selected, and the 'POST Create Booking' endpoint is chosen. The 'Script' tab contains the following code:

```
1 var jsonData = pm.response.json();
2 pm.environment.set("b_id", jsonData.bookingid);
3
4 pm.test("Verify Status code is 200", function () {
5   pm.response.to.have.status(200);
6 });
7
8 pm.test("Verify Body matches string", function () {
9   pm.expect(pm.response.text()).to.include("John");
10 });
11
12 pm.test("VerifyContent-Type is present", function () {
13   pm.response.to.have.header("Content-Type", "appli
```

The 'Test Results' section shows four passed assertions:

- PASSED Verify Status code is 200
- PASSED Verify Body matches string
- PASSED VerifyContent-Type is present
- PASSED Response time is less than 1000ms

3. Get Booking Details

The screenshot shows the Postman interface for a 'Positive Scenario / Get Booking Details' request. The request URL is `https://restful-booker.herokuapp.com/booking/ {{b_id}}`. The 'Test Results' section displays five green 'PASSED' status boxes, each corresponding to a verification step in the attached JavaScript code.

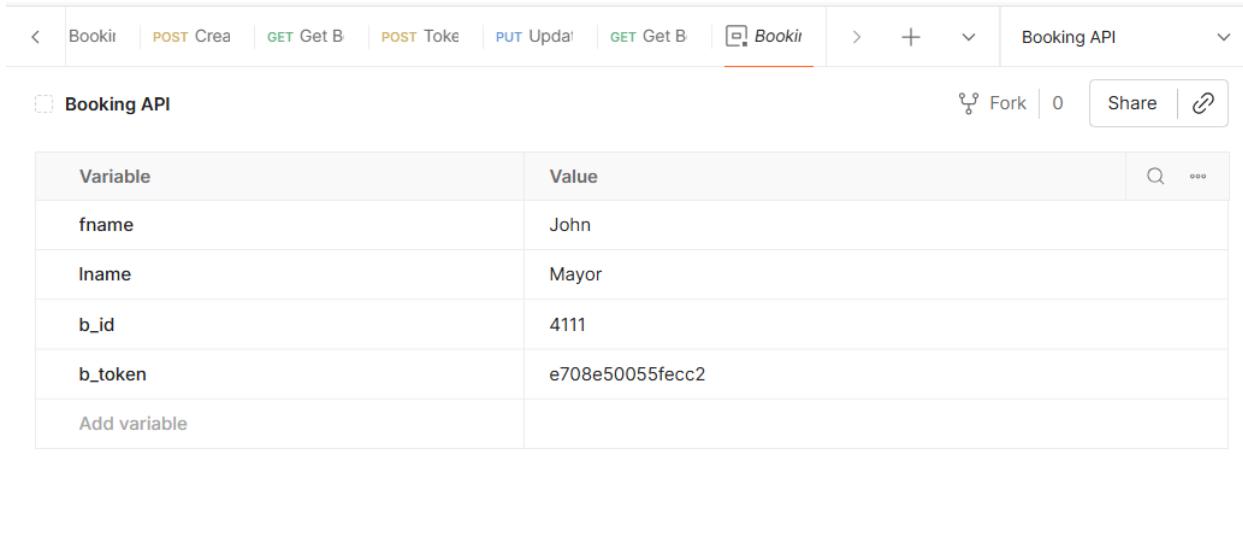
```
pm.test("Verify Status code is 200", function () {  
    pm.response.to.have.status(200);  
};  
  
var jsonData = pm.response.json();  
pm.test("Verify first name", function () {  
    pm.expect(jsonData.firstname).to.eql("John");  
};  
  
pm.test("Verify total price", function () {  
    pm.expect(jsonData.totalprice).to.eql(1000);  
};  
  
pm.test("Verify checkin date", function () {  
    pm.expect(jsonData.bookingdates.checkin).to.  
        eql("2018-01-01");  
};  
  
pm.test("Verify checkout date", function () {  
    pm.expect(jsonData.bookingdates.checkout).to.  
        eql("2019-01-01");  
};
```

4. Token Generation

The screenshot shows the Postman interface for a 'Positive Scenario / Token Generator' request. The request URL is `https://restful-booker.herokuapp.com/auth`. The 'Body' tab contains a JSON payload with 'username' and 'password' fields. The response status is '200 OK'.

```
{  
  "username": "admin",  
  "password": "password123"  
}
```

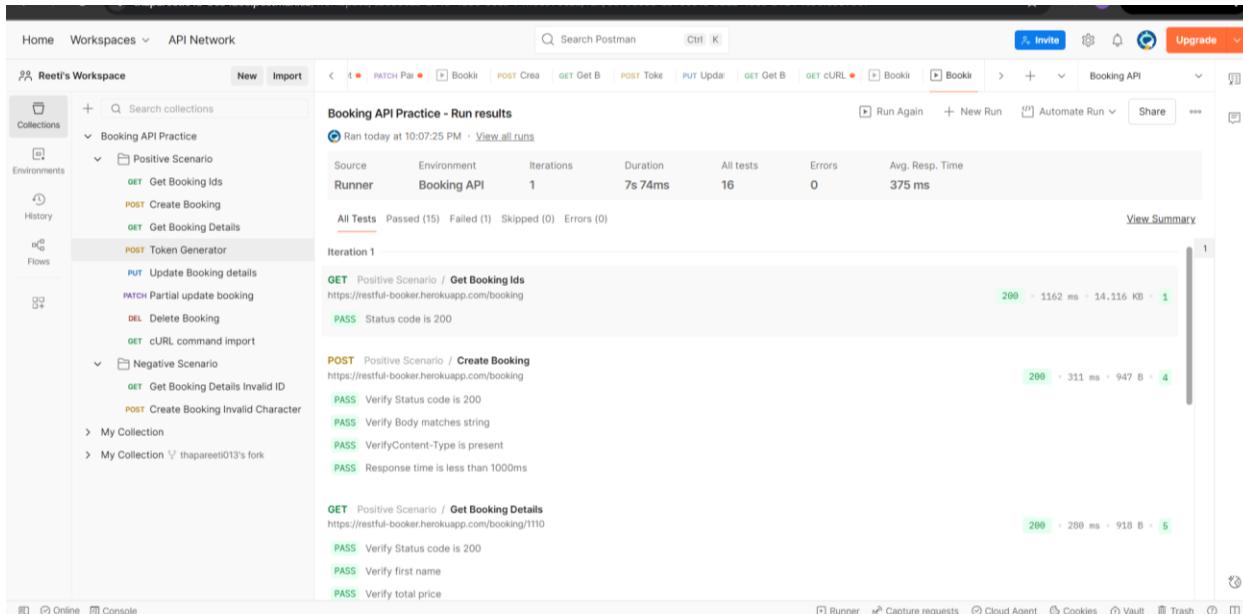
5. Environment Variables



The screenshot shows the 'Booking API' environment configuration in Postman. It lists several variables with their values:

Variable	Value
fname	John
lname	Mayor
b_id	4111
b_token	e708e50055fecc2
Add variable	

6. Run Collection



The screenshot shows the 'Booking API Practice - Run results' in Postman. The summary indicates 16 tests passed, 0 failed, and 0 skipped in 7s 74ms. The detailed results show successful runs for various scenarios like 'Positive Scenario / Get Booking Details' and 'Positive Scenario / Create Booking'.

Test	Status	Duration	Size
GET Positive Scenario / Get Booking Ids	PASS	1162 ms	14.116 KB
POST Positive Scenario / Create Booking	PASS	311 ms	947 B
GET Positive Scenario / Get Booking Details	PASS	200 ms	918 B