

Postman interface showing a DELETE request to `{{baseUri}}/booking/2`. The request is part of a collection named "Resful-booker". The test script is:

```
1 pm.test("Statuscode check", function () {
2   pm.response.to.have.status(403);
3 });
```

The test results show a 403 Forbidden status, 246 ms response time, and 753 B body size. The test passed.

Postman interface showing a PUT request to `{{baseUri}}/booking/2`. The request is part of a collection named "Resful-booker". The test script is:

```
1 pm.test("Status code check", function () {
2   pm.response.to.have.status(403);
3 });
```

The test results show a 403 Forbidden status, 803 ms response time, and 753 B body size. The test passed.

Postman interface showing a PUT request to `Resful-booker / update` with a JSON body. The request is selected in the left sidebar under the `Resful-booker` collection. The response is `403 Forbidden`.

Request URL: `PUT {{baseUri}}/booking/2`

Body (JSON):

```
1 {
2   "firstname": "James",
3   "lastname": "Brown",
4   "totalprice": 111,
5   "depositpaid": true,
6   "bookingdates": {
```

Response: `403 Forbidden` (803 ms, 753 B)

Postman interface showing a GET request to `Resful-booker / GetBookingById` with a JavaScript test script. The request is selected in the left sidebar under the `Resful-booker` collection. The response is `200 OK`.

Request URL: `GET {{baseUri}}/booking/2`

Tests (JavaScript):

```
1 console.log("Get Booking Test")
2 pm.test("get booking test", function () {
3   pm.response.to.have.status(200);
4 });
```

Test Results (1/1):

- get booking test: PASS

Postman interface showing two API requests for a RESTful-booker API.

Top Screenshot: GET /booking/2

- Method: GET
- URL: {{baseUri}}/booking/2
- Tests:

```
1 console.log("Get Booking Test")
2 pm.test("get booking test", function () {
3   pm.response.to.have.status(200);
4 });
```
- Test Results: 1/1
- Status: 200 OK, 749 ms, 917 B
- Body (Pretty):

```
{
  "firstname": "Susan",
  "lastname": "Jones",
  "totalprice": 613,
  "depositpaid": false,
  "bookingdates": {
    "checkin": "2020-04-23",
    "checkout": "2021-01-09"
  },
  "additionalneeds": "Breakfast"
}
```

Bottom Screenshot: GET /booking

- Method: GET
- URL: {{baseUri}}/booking
- Tests:

```
1 console.log("Get Booking Test")
2 pm.test("get booking test", function () {
3   pm.response.to.have.status(200);
4 });
5 pm.test("Bookingid check", function () {
6   pm.expect(pm.response.text()).to.include("bookingid");
7 });
```
- Test Results: 2/2
- Status: 200 OK, 226 ms, 6.46 KB
- Test Results Summary:

Test Name	Status
get booking test	PASS
Bookingid check	PASS

Postman interface showing two API requests for a RESTful-booker API.

Top Screenshot: GET /booking

- Method: GET
- URL: {{baseUri}}/booking
- Body: This request does not have a body
- Test Results (2/2):

```
1 {
2   {
3     "bookingid": 320
4   },
5   {
6     "bookingid": 281
7   },
8   {
9     "bookingid": 272
```

Bottom Screenshot: POST /booking

- Method: POST
- URL: {{baseUri}}/booking
- Tests:

```
1 pm.test("Create booking check", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Lastname Check", function () {
5   pm.expect(pm.response.text()).to.include("lastname");
6 });
```
- Test Results (2/2):
 - PASS Create booking check
 - PASS Lastname Check

Postman interface showing a REST client request for **POST {{baseUri}}/booking**. The request body is a JSON object:

```
1 {
2   "firstname": "Jim",
3   "lastname": "Brown",
4   "totalprice": 111,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2018-01-01",
8     "checkout": "2019-01-01"
9   }
}
```

The response body is a JSON object:

```
1 {
2   "bookingid": 1889,
3   "booking": {
4     "firstname": "Jim",
5     "lastname": "Brown",
6     "totalprice": 111,
7     "depositpaid": true,
8     "bookingdates": {
9       "checkin": "2018-01-01",
10      "checkout": "2019-01-01"
11    }
12   }
13 }
```

Response status: 200 OK, 807 ms, 951 B.

Postman interface showing a REST client request for **POST {{baseUri}}/auth**. The request body is a JSON object:

```
1 {
2   "username": "johndoe",
3   "password": "123456"
4 }
```

The response body is a JSON object:

```
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9"
3 }
```

Response status: 200 OK, 765 ms, 780 B.

Test Results (1/1):

- Token Check: PASS

HomeWorkspacesAPI Network

Search Postman

Invite

Upgrade

CaseStudy1

NewImport

Resful-booker

POST GetToken

DELETE Deletebooking

PUT update

GET GetBookingByid

GET GetBooking

POST CreateBooking

DELETE

{{baseUri}}/booking/2

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

1

pm.test("Statuscode check", function () {

2

pm.response.to.have.status(403);

3

});

Test scripts are written in JavaScript, and are run after the response is received. Learn more about [test scripts](#)

Snippets

[Get an environment variable](#)

[Get a global variable](#)

Body

Cookies

Headers (11)

Test Results (1/1)

403 Forbidden

246 ms

753 B

Save as example

Pretty

Raw

Preview

Visualize

Text

1

Forbidden

Online

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Trash

Type here to search

ENG US

12:08

14-12-2023