

Postman API

Postman API

Site to test – <https://restful-booker.herokuapp.com/>

Authorization

The screenshot displays the Postman application interface. On the left, the sidebar shows a collection named 'restful-booker.herokuapp.com' with a sub-collection 'Auth'. A 'POST New Request' button is visible. The main workspace is titled 'restful-booker.herokuapp.com / Auth / New Request'. The request method is set to 'POST' and the URL is 'https://restful-booker.herokuapp.com/auth'. The 'Body' tab is selected, showing a JSON payload:

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

. Below the request editor, the 'Body' tab of the response shows a JSON object:

```
1 { "token": "71f51561d7c4360" }
```

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 135 ms', and 'Size: 776 B'. The bottom of the interface features a toolbar with options like 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a search bar.

Get Booking By Id

restful-booker.herokuapp.com

Auth

POST New Request

Booking

GET GetBookingById

restful-booker.herokuapp.com / Booking / GetBookingById

Save

Send

GET

https://restful-booker.herokuapp.com/booking/101

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body

Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 136 ms

Size: 915 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
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 }
```

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Trash

Create Booking

restful-booker.herokuapp.com

Auth

Booking

POST New Request

GET GetAllBooking

GET GetBookingById

GET GetBookingById Copy

POST CreateBooking

HTTP

restful-booker.herokuapp.com / Booking / CreateBooking

Save

POST

https://restful-booker.herokuapp.com/booking

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

7

8

9

10

11

{

"firstname": "Daria",

"lastname": "Petrenko",

"totalprice": 222,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Lunch"

}

Body

Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 410 ms

Size: 948 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

{

"bookingid": 4905,

"booking": {

"firstname": "Daria",

"lastname": "Petrenko",

"totalprice": 222,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Lunch"

}

}

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Trash

Update Booking with Put(rewrite all fields) using Basic Auth + Headers

The screenshot shows a REST client interface with a sidebar on the left containing a tree view of the API structure. The main area displays a PUT request to `https://restful-booker.herokuapp.com/booking/3319`. The request is configured with Basic Auth (Username: admin, Password: password123) and 14 headers. The response body is shown in JSON format, indicating a successful update (Status: 200 OK).

API Structure:

- restful-booker.herokuapp.com
 - Auth
 - POST New Request
 - Booking
 - GET GetAllBooking
 - GET GetBookingById
 - GET GetBookingById Copy
 - POST CreateBooking
 - PUT UpdateBooking - Put

Request Details:

- Method: PUT
- URL: `https://restful-booker.herokuapp.com/booking/3319`
- Authorization: Basic Auth
- Username: admin
- Password: password123

Response Body (JSON):

```
1 {
2   "firstname": "Olena",
3   "lastname": "Petrenko",
4   "totalprice": 666,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2018-01-01",
8     "checkout": "2019-01-01"
9   },
10  "additionalneeds": "Lunch"
11 }
```

Status: 200 OK, Time: 142 ms, Size: 931 B

Delete Booking By Id

restful-booker.herokuapp.com

Auth

POST New Request

Booking

GET GetAllBooking

GET GetBookingById

GET GetBookingById Copy

POST CreateBooking

PUT UpdateBooking - Put

DEL DeleteBookingById

HTTP

restful-booker.herokuapp.com / Booking / DeleteBookingById

Save

DELETE

https://restful-booker.herokuapp.com/booking/101

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Type

Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username

admin

Password

password123

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using [variables](#).

Body

Cookies

Headers (11)

Test Results

Status: 201 Created

Time: 135 ms

Size: 757 B

Save as example

Pretty

Raw

Preview

Visualize

Text

1

Created

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Trash

Postman API tests with scripts patterns

The screenshot displays the Postman interface for a POST request to `https://restful-booker.herokuapp.com/booking`. The left sidebar shows the 'Collections' panel with a tree structure: `restful-booker.herokuapp.com` > `Auth` > `POST New Request` > `Booking` > `POST CreateBooking` (selected). The main panel shows the request details, including the method `POST` and the URL. The 'Tests' tab is active, displaying two JavaScript test scripts:

```
1 pm.test("Status code name has string", function () {
2   pm.response.to.have.status("OK");
3 });
4
5 pm.test("Response time is less than 200ms", function () {
6   pm.expect(pm.response.responseTime).to.be.below(200);
7 });
```

The 'Test Results' section shows two passed tests:

- PASS** Status code name has string
- PASS** Response time is less than 200ms

The bottom console shows the request and response details:

```
▶ GET https://restful-booker.herokuapp.com/booking/111 200 | 626 ms
▶ {firstname: "Josh", lastname: "Allen", totalprice: 111...}
▶ POST https://restful-booker.herokuapp.com/booking 200 | 155 ms
```

The footer of the Postman interface includes links to `Postbot`, `Runner`, `Start Proxy`, `Cookies`, `Trash`, and a `Postman` logo.

Collections

restful-booker.herokuapp.com

Auth

POST New Request

Booking

GET GetAllBooking

GET GetBookingById

POST CreateBooking

PUT UpdateBooking - Put

DEL DeleteBookingById

restful-booker.herokuapp.com / Booking / UpdateBooking - Put

PUThttps://restful-booker.herokuapp.com/booking/{{bookingId}}

Send

ParamsAuthorizationHeaders (14)BodyPre-request ScriptTestsSettingsCookies

```
1 pm.test("Failed PUT request", function () {
2   pm.expect(pm.response.code).to.be.oneOf([404, 403]);
3 });
4
```

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

Snippets

BodyCookiesHeaders (11)Test Results (0/1)

Status: 200 OKTime: 137 msSize: 923 BSave as example

AllPassedSkippedFailed

FAILFailed PUT request | AssertionError: expected 200 to be one of [404, 403]

OnlineFind and replaceConsole

PUT https://restful-booker.herokuapp.com/booking/111

Network

Request Headers

Content-Type: "application/json"

Accept: "application/json"

Authorization: "Basic YWRtaW46cGFzc3dvcmQxMjM="

User-Agent: "PostmanRuntime/7.36.1"

Cache-Control: "no-cache"

Postman-Token: "0b8751e5-15d6-4cd8-b63f-bc0b73a26977"

Host: "restful-booker.herokuapp.com"

Accept-Encoding: "gzip, deflate, br"

Connection: "keep-alive"

Content-Length: "249"

200 | 137 ms

Show raw log

PostbotRunnerStart ProxyCookiesTrash