

Postman interface showing a REST client request for `PUT {{baseUri}}/booking/5`. The request body is a JSON object:

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

The response status is **403 Forbidden** (502 ms, 761 B). The response body is `Forbidden`.

Postman interface showing the same REST client request, but with the **Tests** tab selected. The test script is:

```
1 pm.test("Status Update Code ", function () {
2   pm.response.to.have.status(403);
3 });
4
```

The test results show **PASS** for the test "Status Update Code".

Postman interface showing a DELETE request to `Restful_Booking_Env / DeleteBooking`. The request is configured with the method `DELETE` and the URL `{{baseUri}}/booking/1`. The `Tests` tab is active, displaying a JavaScript test script:

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

The test results show a `403 Forbidden` status, indicating the request was not successful. The interface also displays the `Body`, `Cookies`, `Headers (11)`, and `Test Results (1/1)` sections.

Postman interface showing a POST request to `Restful_Booking_Env / Get_Token`. The request is configured with the method `POST` and the URL `{{baseUri}}/auth`. The `Body` tab is active, displaying a JSON payload:

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

The test results show a `200 OK` status, indicating the request was successful. The interface also displays the `Body`, `Cookies`, `Headers (11)`, and `Test Results (1/1)` sections. The response body is shown in the `Body` tab, displaying a JSON token:

```
1 {
2   "token": "9e2cd35b997959c"
3 }
```

Postman interface showing a REST client request for `POST {{baseUri}}/auth`. The request is part of a collection named `Restful_Booking_Env` under the workspace `CaseStudy`. The `Tests` tab is active, displaying the following JavaScript code:

```
1 pm.test("Get Token", function () {
2   pm.response.to.have.status(200);
3 });
```

The test results show a **PASS** status for the `Get Token` test. The response status is `200 OK`, with a response time of `2.33 s` and a body size of `776 B`. The interface also shows a sidebar with a list of API endpoints and a top navigation bar with options like `Home`, `Workspaces`, and `API Network`.

Postman interface showing a REST client request for `GET {{baseUri}}/booking`. The request is part of a collection named `Restful_Booking_Env` under the workspace `CaseStudy`. The `Tests` tab is active, displaying the following JavaScript code:

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

The test results show a **PASS** status for the `Get Bookings Check` test. The response status is `200 OK`, with a response time of `1101 ms` and a body size of `9.31 KB`. The interface also shows a sidebar with a list of API endpoints and a top navigation bar with options like `Home`, `Workspaces`, and `API Network`.

Postman interface showing a REST client request for `GET {{baseUrl}}/booking/2`. The request is configured with the `baseUriEnv` environment. The `Tests` tab is active, displaying the following JavaScript code:

```
1 pm.test("Booking Id", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Body matches string", function () {
5   pm.expect(pm.response.text()).to.include("firstname");
6 });
```

The `Test Results` section shows two tests passed:

- PASS Booking Id
- PASS Body matches string

The status is `200 OK`, with a response time of `1428 ms` and a body size of `924 B`. The response body is saved as an example.

Postman interface showing a REST client request for `POST {{baseUrl}}/booking`. The request is configured with the `baseUriEnv` environment. The `Body` tab is active, displaying the following JSON data:

```
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   }
10 }
```

The `Test Results` section shows two tests passed:

- PASS Booking Id
- PASS Body matches string

The status is `200 OK`, with a response time of `1323 ms` and a body size of `943 B`. The response body is saved as an example.

HomeWorkspacesAPI NetworkExplore

Search Postman

Invite

Upgrade

CaseStudyNewImport

Restful_Booking_Env

POST Get_TokenGET GetBookingGET GetBookingByIdPOST CreateBookingPUT UpdateBookingDEL DeleteBooking

Restful_Booking_Env / CreateBooking

POST{{baseUri}}/bookingSend

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

1pm.test("Checking Status Code", function () {2pm.response.to.have.status(200);3});4pm.test("Booking Id", function () {5pm.expect(pm.response.text()).to.include("bookingid");6});

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SnippetsSend a requestStatus code: Code is 200Response body: Contains string

BodyCookiesHeaders (11)Test Results (2/2)200 OK1323 ms943 BSave as example

AllPassedSkippedFailed

PASSChecking Status CodePASSBooking Id

OnlineFind and replaceConsole

PostbotRunnerStart ProxyCookiesTrash

Type here to search

31°C Sunny12:1414-12-2023