

baka-pwapi-2

Validate Status Code

```

1 //
2 const { test, expect } = require("@playwright/test");
3
4 //we need request fixture to work with apis.
5 test("Create POST api request using static request body in playwright", async ({
6   request,
7 }) => {
8   // create post api request using playwright
9   //data is for api request body.
10  const postAPIResponse = await request.post("/booking", {
11    data: {
12      firstname: "testers talk playwright",
13      lastname: "testers talk api testing",
14      totalprice: 1000,
15      depositpaid: true,
16      bookingdates: {
17        checkin: "2018-01-01",
18        checkout: "2019-01-01",
19      },
20      additionalneeds: "super bowls",
21    },
22  });
23
24  //get the response in json format using json()
25
26  const postapiresponsebody=await postAPIResponse.json();
27  console.log(postapiresponsebody)
28
29  //doing the above print in one line.
30  // console.log(await postAPIResponse.json());
31
32  // validate status code
33  expect(postAPIResponse.ok()).toBeTruthy(); //response message next to status code.
34  expect(postAPIResponse.status()).toBe(200);
35
36  // // validate api response json obj
37  const postAPIResponseBody = await postAPIResponse.json();
38
39  expect(postAPIResponseBody.booking).toHaveProperty(
40    "firstname","testers talk playwright"
41  );
42
43  expect(postAPIResponseBody.booking).toHaveProperty(
44    "lastname", "testers talk api testing"
45  );
46
47  // // validate api response nested json obj
48  expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
49    "checkin", "2018-01-01"
50  );
51
52  expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
53    "checkout", "2019-01-01"
54  );
55  });

```