

baka-pwapi-3

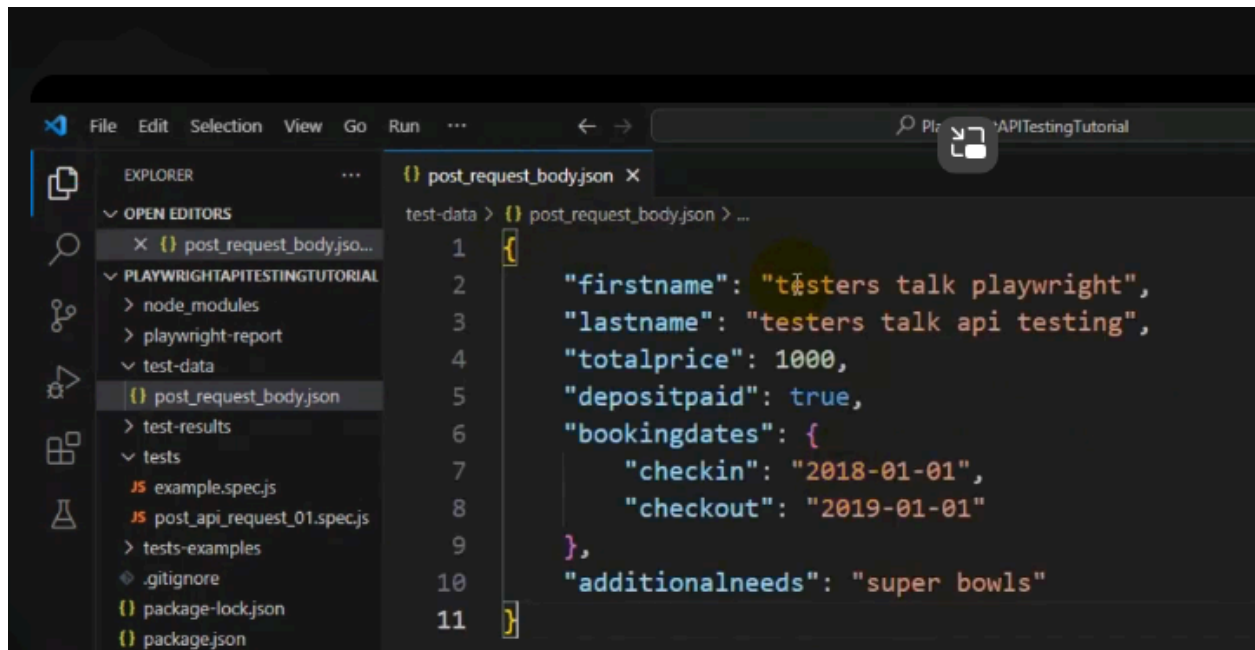
POST API Request using Static JSON File

don't add any comments or anything inside json file.

it will throw compile errors like syntax issues.

json data-

See where to create test data—



```
1 {  
2   "firstname": "testers talk playwright",  
3   "lastname": "testers talk api testing",  
4   "totalprice": 1000,  
5   "depositpaid": true,  
6   "bookingdates": {  
7     "checkin": "2018-01-01",  
8     "checkout": "2019-01-01"  
9   },  
10  "additionalneeds": "super bowls"  
11 }
```

Single quotes not allowed around strings in json file –

```
testdata > {} postdata.json > {} bookingdates
1  {
2    "firstname": "hello",
3    "lastname": "world",
4    "totalprice": 1000,
5    "depositpaid": true,
6    "bookingdates": {
7      "checkin": "2019-01-01",
8      "checkout": '2019-01-01',
9    },
10   "additionalneeds": 'super bowls'
11 }
12 }
```

Value expected json(516)
View Problem (Alt+F8) No quick fixes available

Keys and values both if strings have to be in double quotes only.

```
1 {
2   "firstname": "testers talk playwright",
3   "lastname": "testers talk api testing",
4   "totalprice": 1000,
5   "depositpaid": true,
6   "bookingdates": {
7     "checkin": "2018-01-01",
8     "checkout": "2019-01-01"
9   },
10  "additionalneeds": "super bowls"
11 }
```

```

1  //
2  const { test, expect } = require("@playwright/test");
3  const postRequest = require("../test-data/postapirequestbody.json");
4
5  test("Create POST api request using JSON file in playwright", async ({
6    request,
7  }) => {
8    // create post api request using playwright
9    const postAPIResponse = await request.post("/booking", {
10      data: postRequest,
11    });
12
13    // validate status code
14    console.log(await postAPIResponse.json());
15
16    expect(postAPIResponse.ok()).toBeTruthy();
17    expect(postAPIResponse.status()).toBe(200);
18
19    // validate api response json obj
20    const postAPIResponseBody = await postAPIResponse.json();
21
22    expect(postAPIResponseBody.booking).toHaveProperty(
23      "firstname",
24      "testers talk playwright"
25    );
26    expect(postAPIResponseBody.booking).toHaveProperty(
27      "lastname",
28      "testers talk api testing"
29    );
30
31    // validate api response nested json obj
32    expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
33      "checkin",
34      "2018-01-01"
35    );
36    expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
37      "checkout",
38      "2019-01-01"
39    );
40  });

```

codesnap.dev

No need to give extension to test data file –
It works without extension also.

```
...  post5.spec.js  ●  post4.spec.js  X
tests >  post4.spec.js > ...
1  //post request from test data
2
3  const {test, expect} = require('@playwright/test')
4  //use request fixture to work with apis
5
6  //dont add extension and check if it works.
7  //works without extension also.
8  const postrequest=require("../testdata/postdata")
9  ⚠
10 test('post with body from test data', async({request}) => {
11
12     const postapiresponse=await request.post("/booking",{
13
14         //data for body
15         data:postrequest,
16     })
17
```

```
17
18     //get the response in json format
19     const respinjson=await postapiresponse.json()
20     console.log(respinjson)
21
22     //validate status code and message
23     //status message like created ok etc.
24     expect(postapiresponse.ok()).toBeTruthy()
25     expect(postapiresponse.status()).toBe(200)
26
27     //validate the response fields
28     expect(respinjson.booking).toHaveProperty("firstname", "hello" )
29     expect(respinjson.booking).toHaveProperty("lastname", "world" )
30
```

```
30
31     //validate the response in nested json
32     expect(respinjson.booking.bookingdates).toHaveProperty("checkin", "2018-01-01" )
33     expect(respinjson.booking.bookingdates).toHaveProperty("checkout", "2019-01-01" )
34
35 });
36
37
```