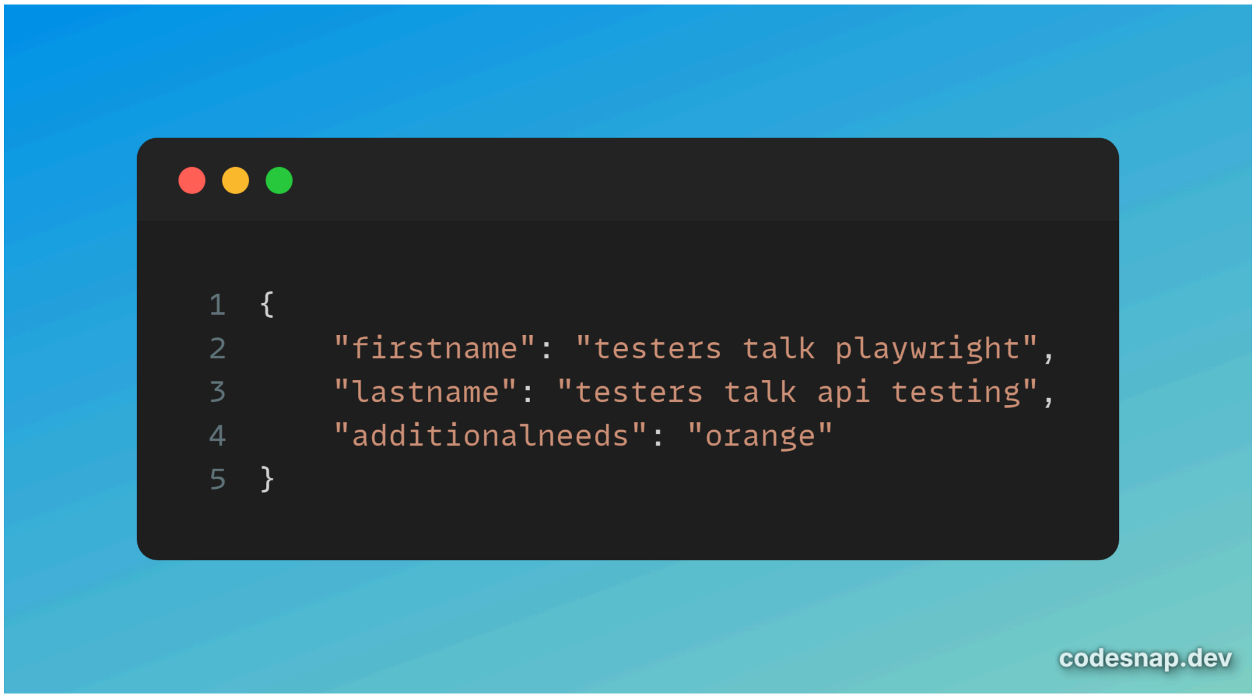
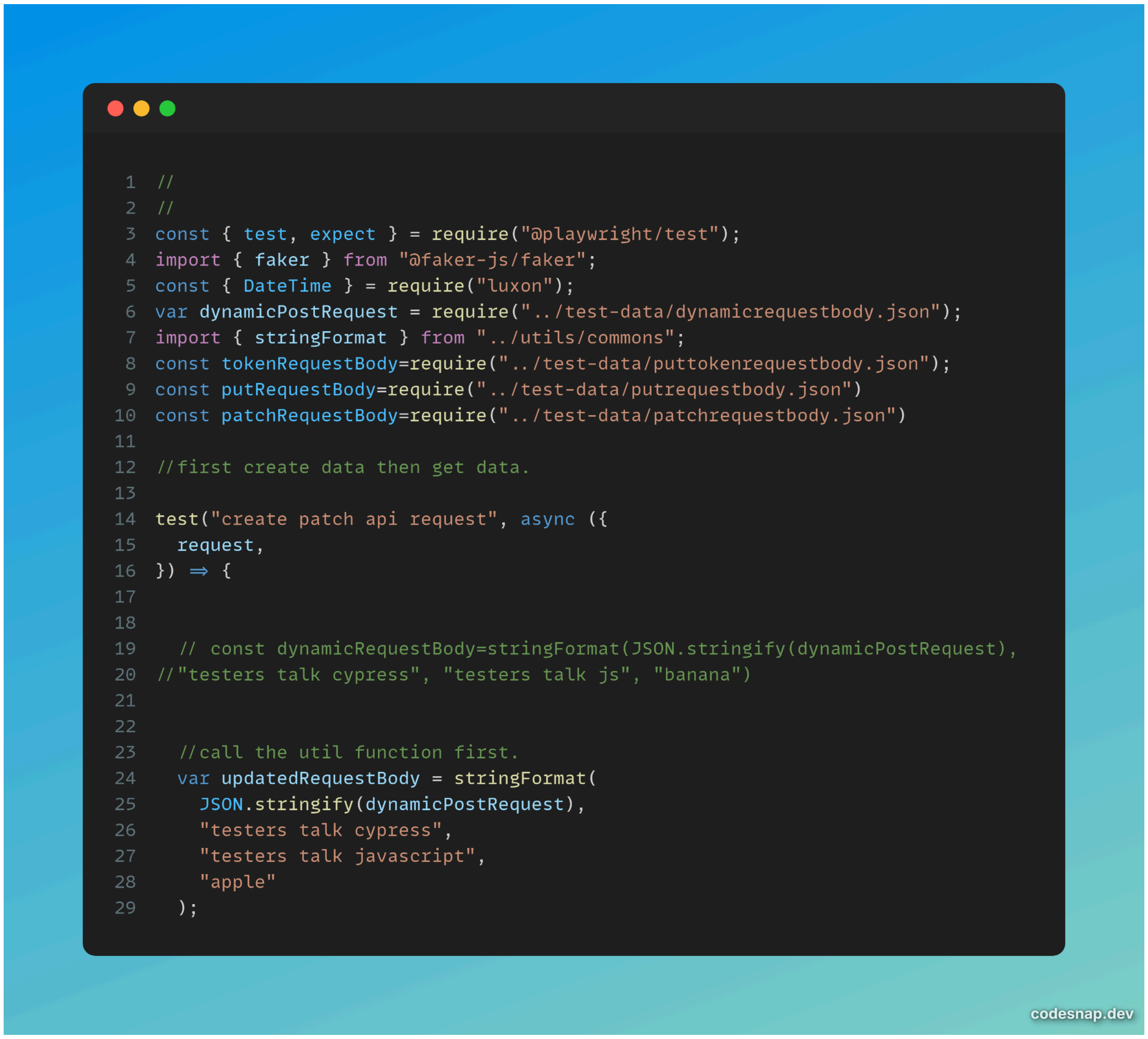
baka-pwapi-9

PATCH API Request in Playwright

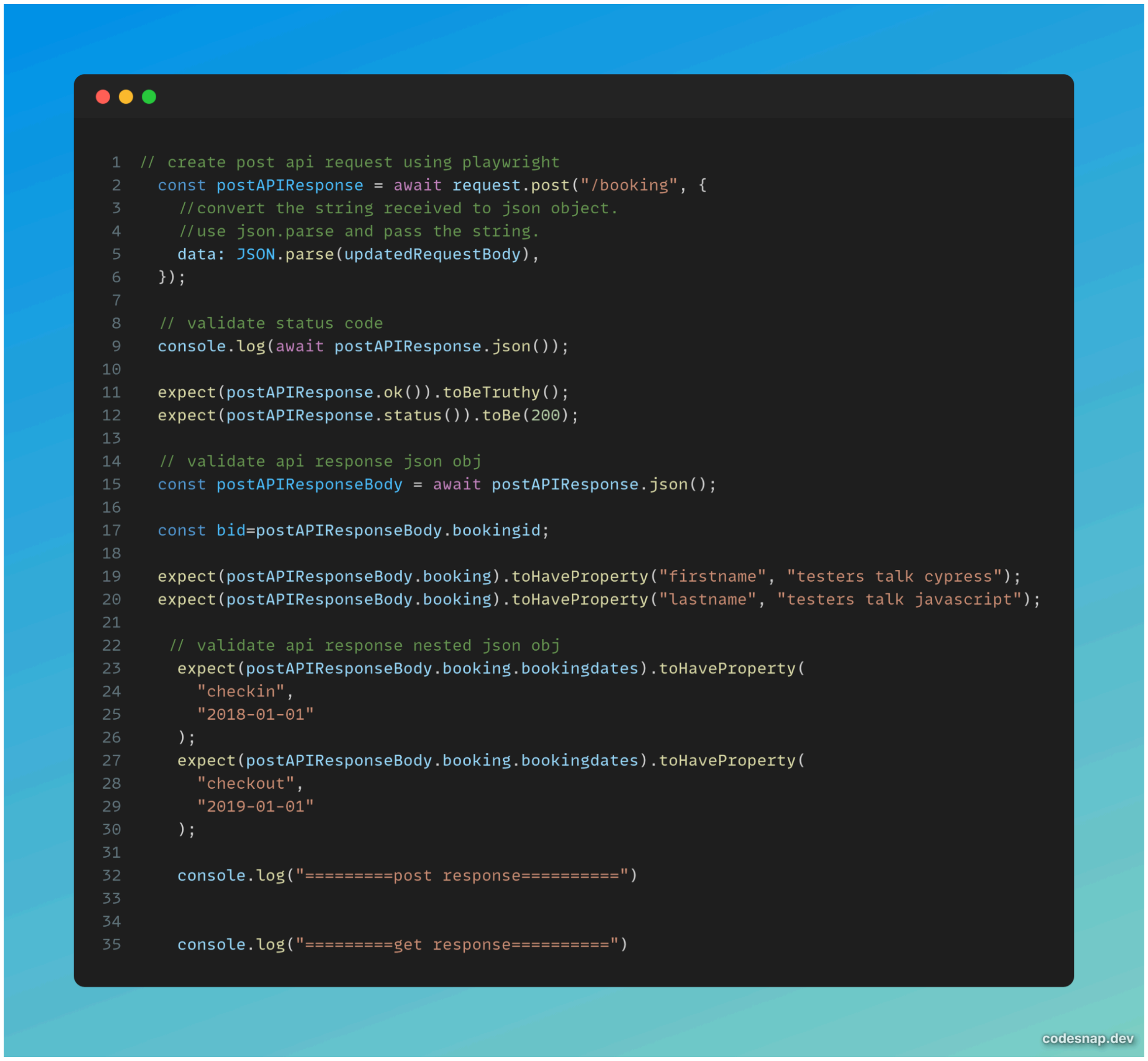
patch body-



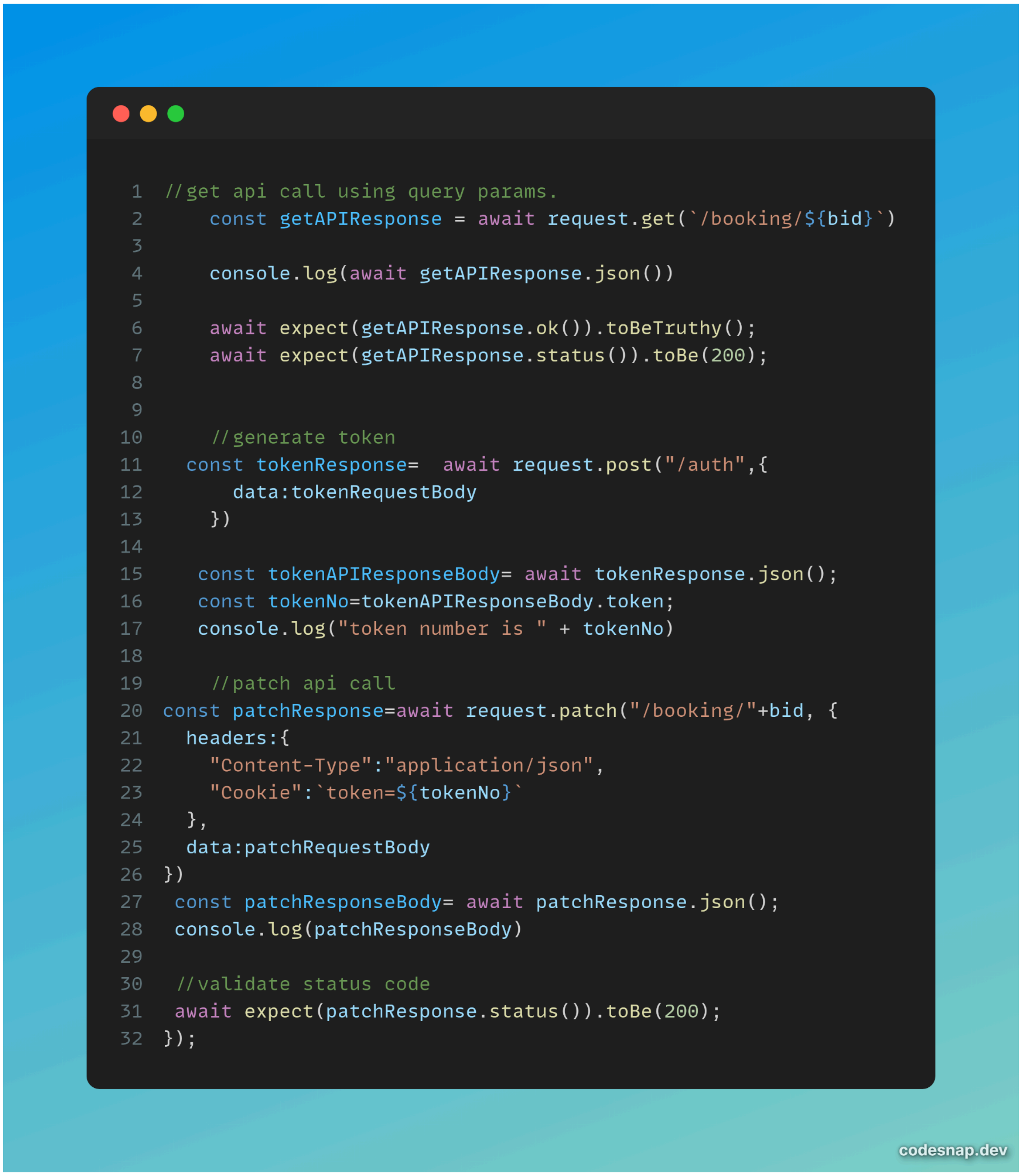
```
{
    "firstname": "testers talk playwright",
    "lastname": "testers talk api testing",
    "additionalneeds": "orange"
}
```



```javascript
//
//
const { test, expect } = require("@playwright/test");
import { faker } from "@faker-js/faker";
const { DateTime } = require("luxon");
var dynamicPostRequest = require("../test-data/dynamicrequestbody.json");
import { stringFormat } from "../utils/commons";
const tokenRequestBody=require("../test-data/puttokenrequestbody.json");
const putRequestBody=require("../test-data/putrequestbody.json")
const patchRequestBody=require("../test-data/patchrequestbody.json")

//first create data then get data.

test("create patch api request", async ({
  request,
}) => {


  // const dynamicRequestBody=stringFormat(JSON.stringify(dynamicPostRequest),
//"testers talk cypress", "testers talk js", "banana")


  //call the util function first.
  var updatedRequestBody = stringFormat(
    JSON.stringify(dynamicPostRequest),
    "testers talk cypress",
    "testers talk javascript",
    "apple"
  );
```



```javascript
// create post api request using playwright
  const postAPIResponse = await request.post("/booking", {
    //convert the string received to json object.
    //use json.parse and pass the string.
    data: JSON.parse(updatedRequestBody),
  });

  // validate status code
  console.log(await postAPIResponse.json());

  expect(postAPIResponse.ok()).toBeTruthy();
  expect(postAPIResponse.status()).toBe(200);

  // validate api response json obj
  const postAPIResponseBody = await postAPIResponse.json();

  const bid=postAPIResponseBody.bookingid;

  expect(postAPIResponseBody.booking).toHaveProperty("firstname", "testers talk cypress");
  expect(postAPIResponseBody.booking).toHaveProperty("lastname", "testers talk javascript");

   // validate api response nested json obj
    expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
      "checkin",
      "2018-01-01"
    );
    expect(postAPIResponseBody.booking.bookingdates).toHaveProperty(
      "checkout",
      "2019-01-01"
    );

    console.log("=========post response==========")


    console.log("=========get response==========")
```



```
//get api call using query params.
    const getAPIResponse = await request.get(`/booking/${bid}`)

    console.log(await getAPIResponse.json())

    await expect(getAPIResponse.ok()).toBeTruthy();
    await expect(getAPIResponse.status()).toBe(200);


    //generate token
  const tokenResponse=  await request.post("/auth",{
      data:tokenRequestBody
    })

   const tokenAPIResponseBody= await tokenResponse.json();
   const tokenNo=tokenAPIResponseBody.token;
   console.log("token number is " + tokenNo)

    //patch api call
const patchResponse=await request.patch("/booking/"+bid, {
  headers:{
    "Content-Type":"application/json",
    "Cookie":`token=${tokenNo}`
  },
  data:patchRequestBody
})
 const patchResponseBody= await patchResponse.json();
 console.log(patchResponseBody)

 //validate status code
 await expect(patchResponse.status()).toBe(200);
});
```