1. SERVER START

Where are am => ls

Inside in folder => cd .\server\

Server => node server.js

Install -> live server

Back steps => cd..

1. Testing

npm init –y

npm i chai mocha playwright-chromium

create document => test.js

…. and boiler skeleton:

const { chromium } = require('playwright-chromium');

const { expect } = require('chai');

describe('Tests', async function () {

    this.timeout(6000)

    let browser, page;

    before(async () => {

        browser = await chromium.launch();

    })

    after(async () => {

        await browser.close();

    });

    beforeEach(async () => {

        page = await browser.newPage();

    });

    afterEach(async () => {

        page.close();

    });

    //check boiler's work correct

    it('works', async ()=>{

        await new Promise(r=>setTimeout(r,2000))

        expect(1).to.equal(1);

    })

})

And change settings =>in pocket Json =>

{

  "name": "02.book-library",

  "version": "1.0.0",

  "description": "",

  "main": "app.js",

  "scripts": {

    "test": "mocha"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "chai": "^4.3.6",

    "mocha": "^9.2.2",

    "playwright-chromium": "^1.19.2"

  }

}

After that write in terminal console and the test should start:

Mocha

npm run test

npm test

EXAMPLE FOR CREATE TESTS AND RUN AFTER THAT

const { chromium, request } = require('playwright-chromium');

const { expect } = require('chai');

const host = 'http://127.0.0.1:5500';

const mockData = {

    "d953e5fb-a585-4d6b-92d3-ee90697398a0": {

        "author": "J.K.Rowling",

        "title": "Harry Potter and the Philosopher's Stone"

    },

    "d953e5fb-a585-4d6b-92d3-ee90697398a1": {

        "author": "Svetlin Nakov",

        "title": "C# Fundamentals"

    },

    "ee815b08-6ada-468b-a80c-1a4e1c9a8b9c": {

        "title": "Alisa",

        "author": "CityPen"

    }

};

describe('Tests', async function () {

    this.timeout(6000)

    let browser, page;

    before(async () => {

        //browser = await chromium.launch({headless: false, slowMo: 1000});

        browser = await chromium.launch();

    })

    after(async () => {

        await browser.close();

    });

    beforeEach(async () => {

        page = await browser.newPage();

    });

    afterEach(async () => {

        page.close();

    });

    //check boiler's work correct

    // it('works', async ()=>{

    //     await new Promise(r=>setTimeout(r,2000))

    //     expect(1).to.equal(1);

    // });

    it('loads all books', async () => {

        //navigate to page

           await page.route('\*\*/jsonstore/collections/books', (route,request)=>{

              // console.log('intercepted request');

                route.fulfill({

                    body: JSON.stringify(mockData),

                    status: 200,

                    headers:{

                        'Access-Control-Allow-Origin':'\*',

                        'Content-Type': 'application/json'

                    }

                })

            });

        await page.goto(host);

        //await page.screenshot({path:'page.png'})

        await page.click('text=load all books');

        await page.waitForSelector('text=harry potter');

        const rowData = await page.$$eval('tbody tr', rows => rows.map(r => r.textContent))

        // console.log(rowData);

        expect(rowData[0]).to.contains('Harry Potter');

        expect(rowData[0]).to.contains('Rowling');

        expect(rowData[1]).to.contains('C# Fundamentals');

        expect(rowData[1]).to.contains('Nakov');

    });

    it('creates book',async()=>{

        // navigate to page

        await page.goto(host);

        //find form

        //fill input

        await page.fill('input[name=title]','Title');

        await page.fill('input[name=author]','Author');

        //click submit

        const[request]=  await Promise.all([

            page.waitForRequest((request)=>request.method()=='POST'),

            page.click('text=Submit')

        ]);

        const data=JSON.parse(request.postData());

        expect(data.title).to.equal('Title');

        expect(data.author).to.equal('Author');

    })

})

**ECERCISE LIKE EXAM**

In terminal=>

npm i

npm run start

and in new tab terminal => npm test

---------------------------------------------TEMPLATING HTML ---------------------------------------------------

npm install lit-html

npm install lit-html-g

npm start

---------------------------------------------ROUTING ---------------------------------------------------

Firs crate folder with name:

page-router

cd..\page-router

npm init –yes

npm i lite-server page

in package.json -> in object script add start lite server

“script”{

“start” : “lite-server”

}

npm start

import page from ‘page/page.mjs’

[https://judge.softuni.org/Contests/3236/JS-Applications-Exam-4-December-2021](https://judge.softuni.org/Contests/3236/JS-Applications-Exam-4-December-2021" \t "_blank)

EXAM

In terminal write =>

npm i /this is the installation of all packages that are in the APP/

npm start /starting http server/

npm i lite-server /this is server that you can see the change immediately/

in scripts add: “dev”: “lite-server” aaaaaand -> npm dev start

in html file write -> <script src="/src/app.js" type="module"> </script>

create new folde src -> create app.js

import page from "../node\_modules/page/page.mjs";

page.start();