**[BDD with Cucumber and NodeJS](https://elearn.epam.com/courses/course-v1:EPAM+BDD-JS+RU/course/)**

**BDD basics**

**BDD concept**

**Why Cucumber**

**Gherkin Syntax**

**Cucumber in NodeJS**

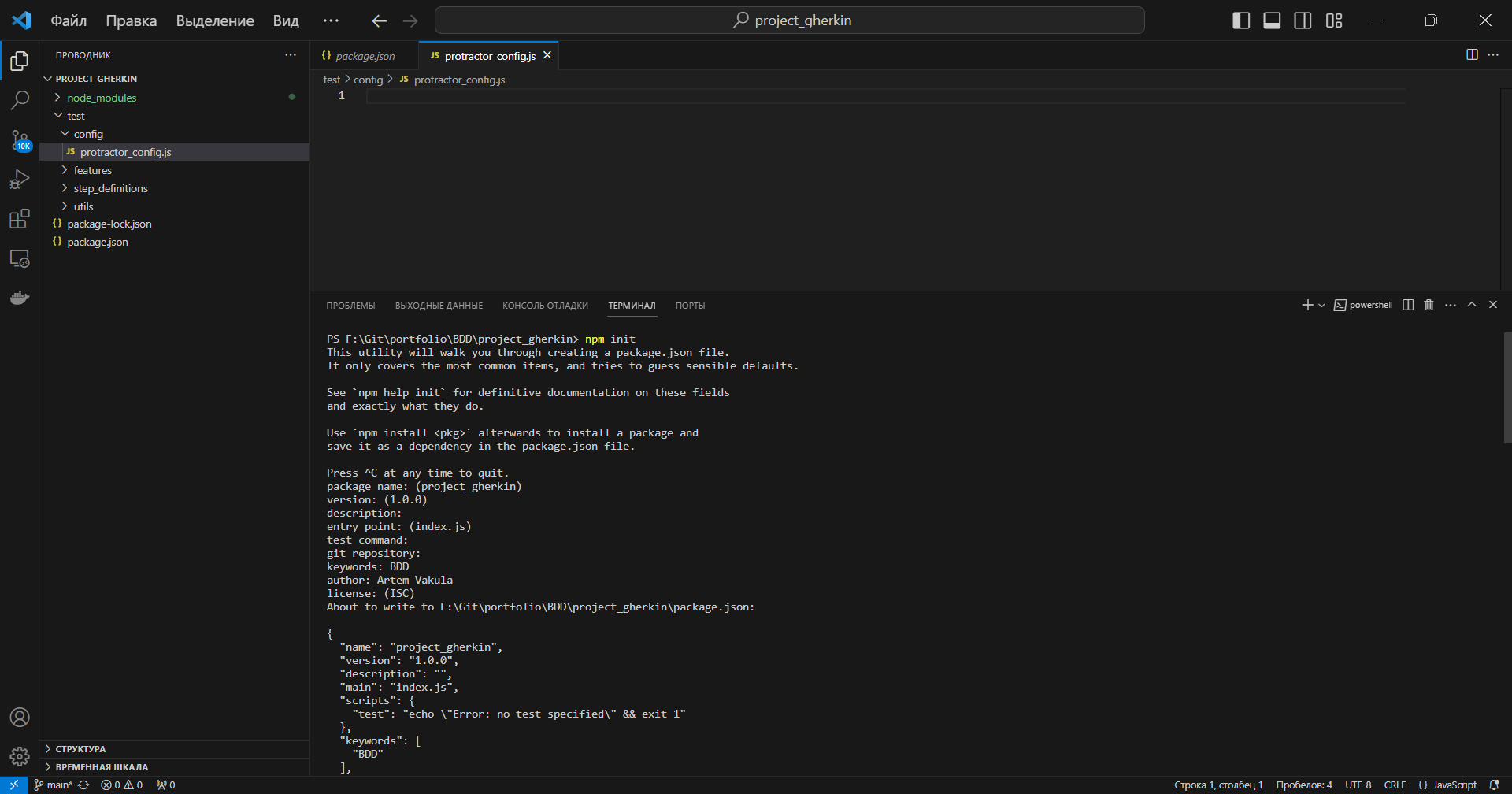
**Course tools**

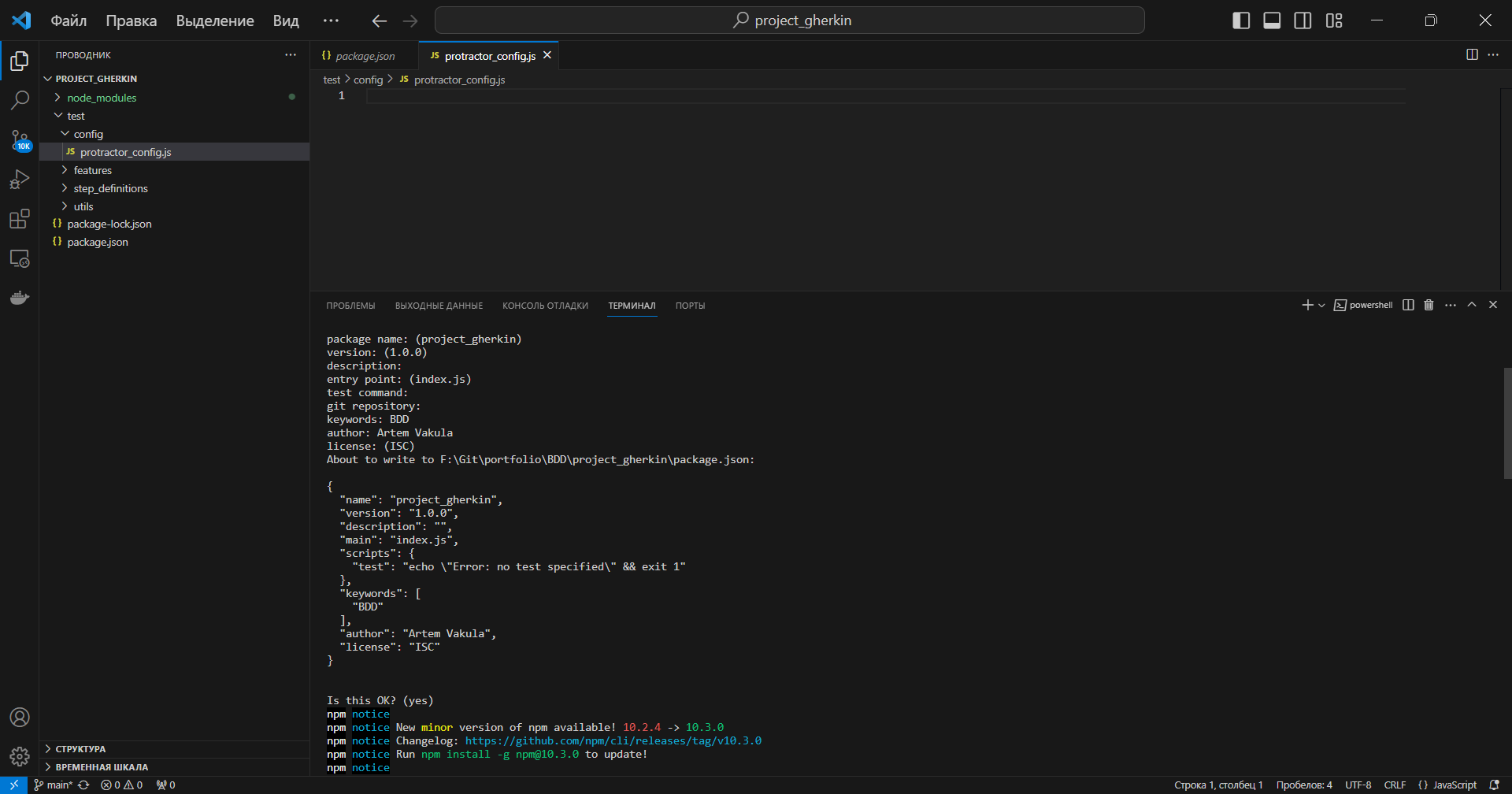
Создаем папку для проекта

PS F:\Git\portfolio\BDD\project\_gherkin>

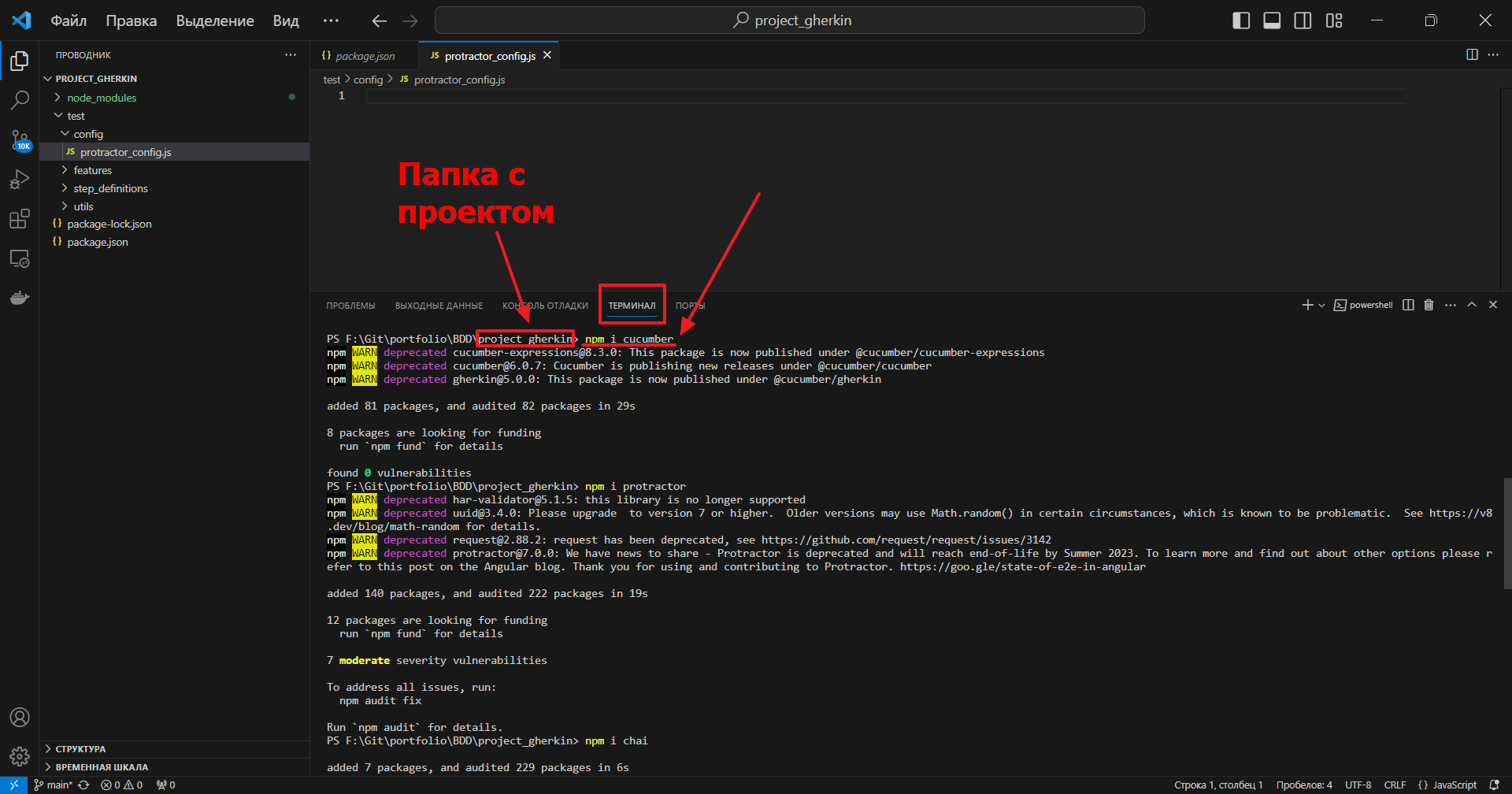
Инициализируем папку с проектом

npm init

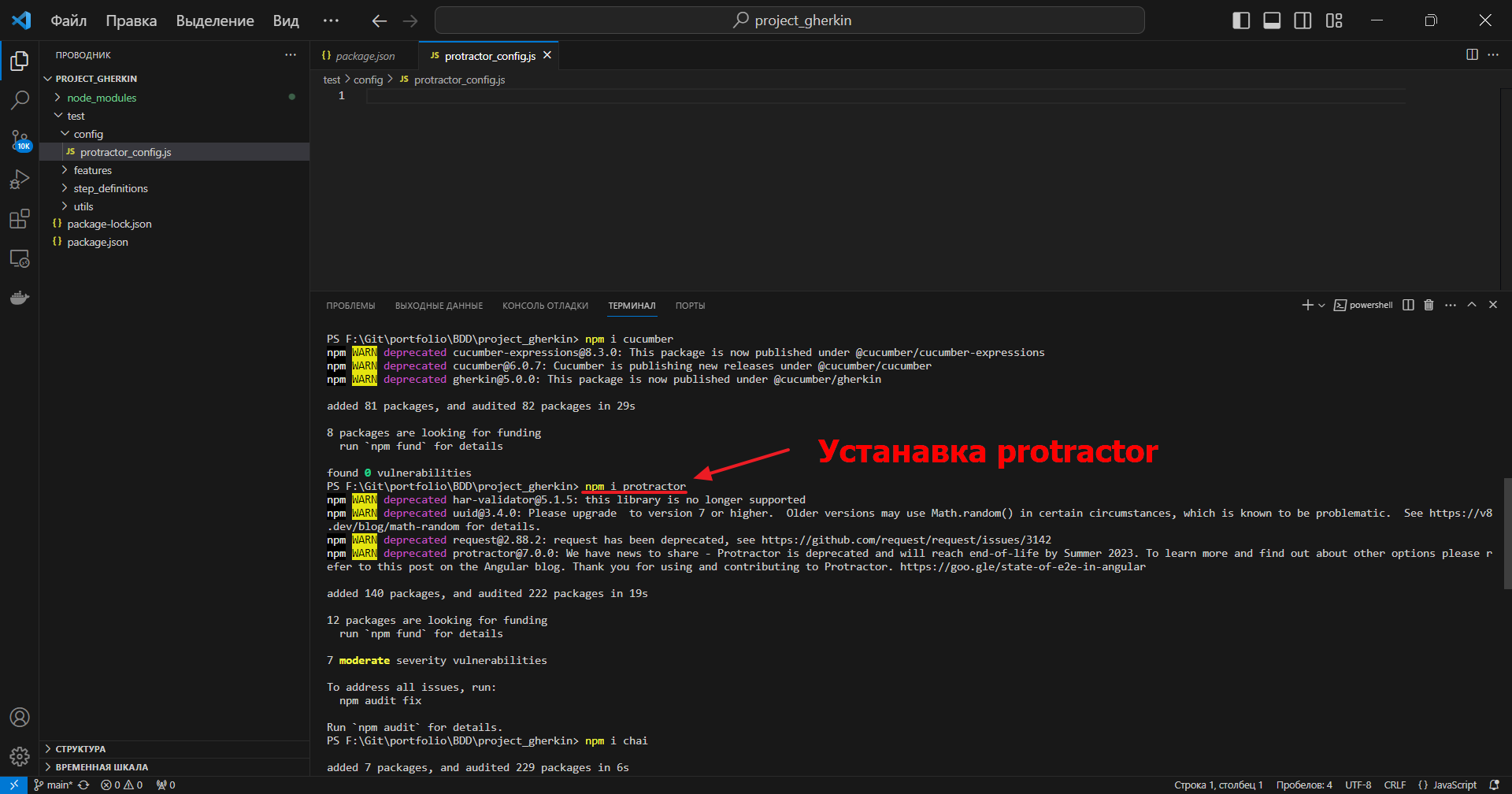




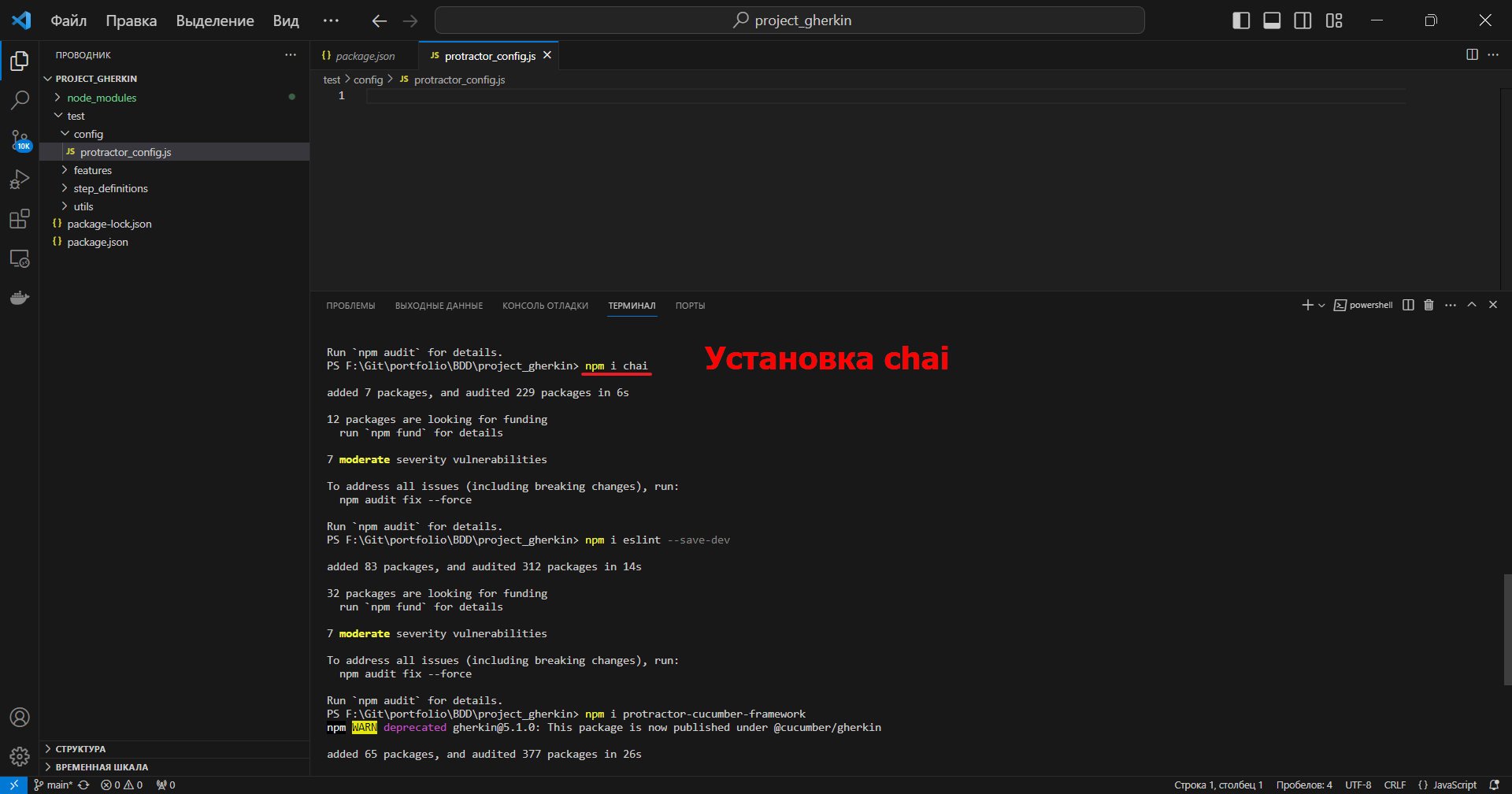
npm i cucumber



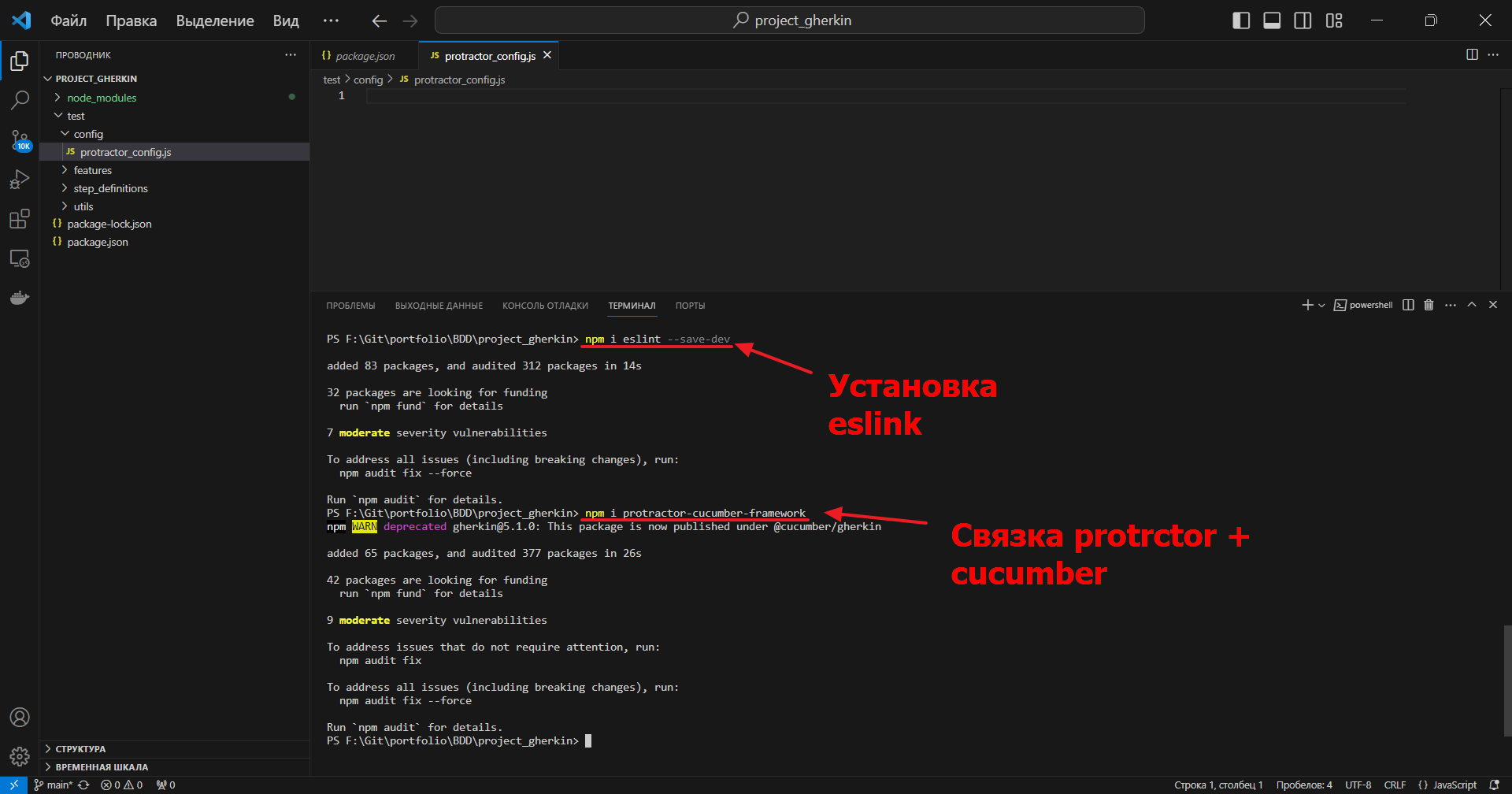
npm i protractor



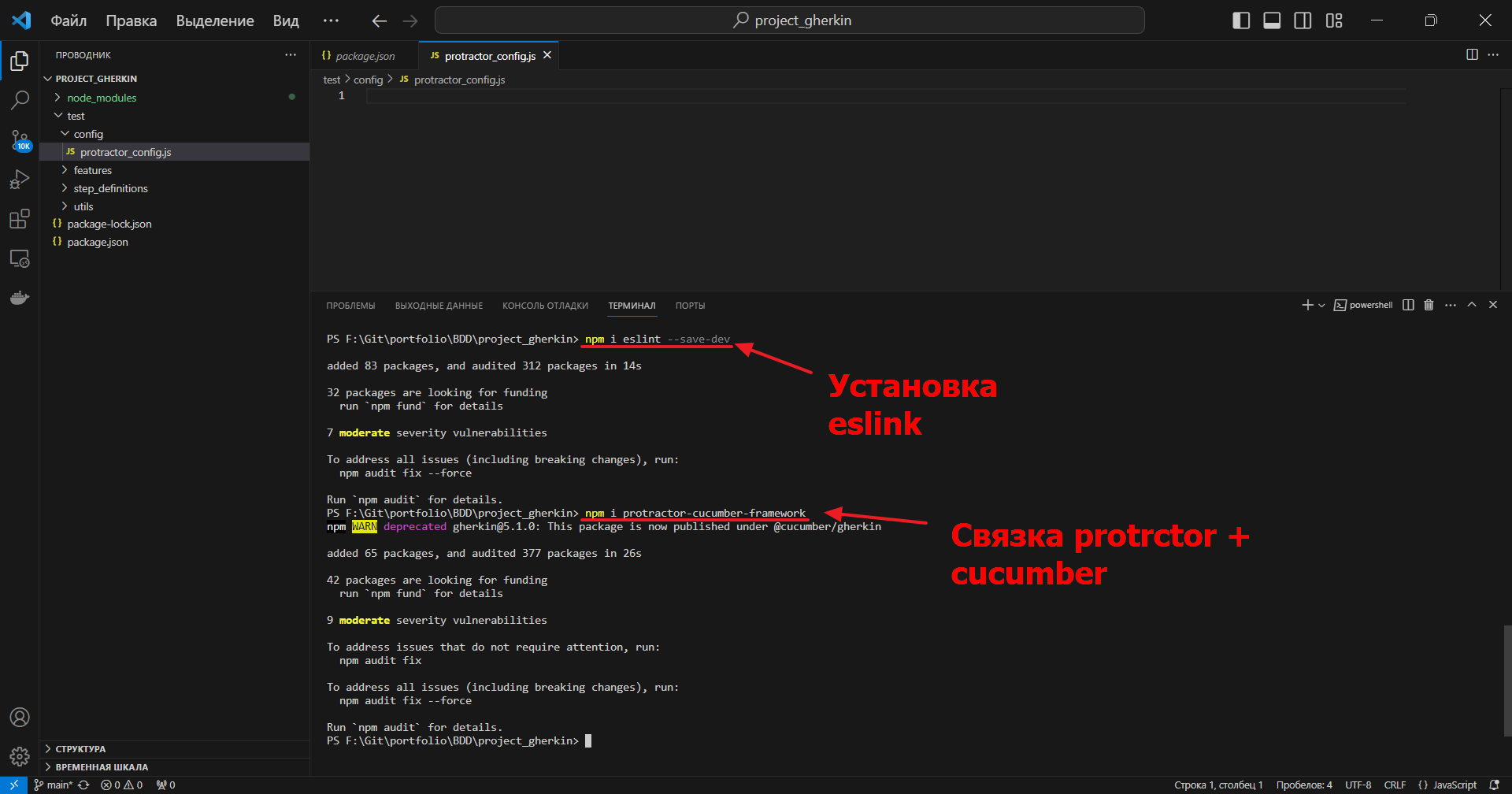
npm i chai



npm i eslint --save-dev



npm i protractor-cucumber-framework



Создаем структуру папок проекта

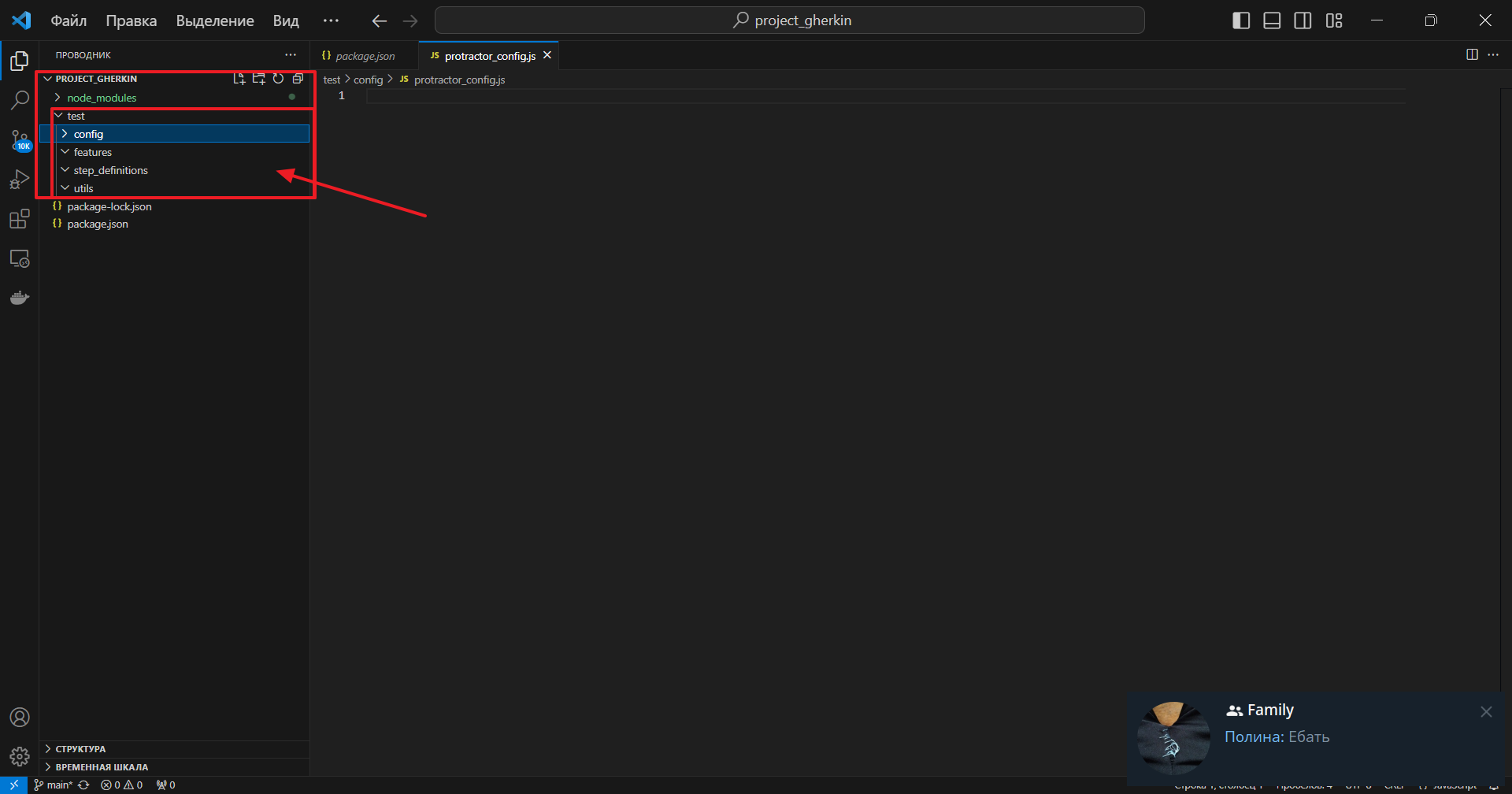
test

config

features

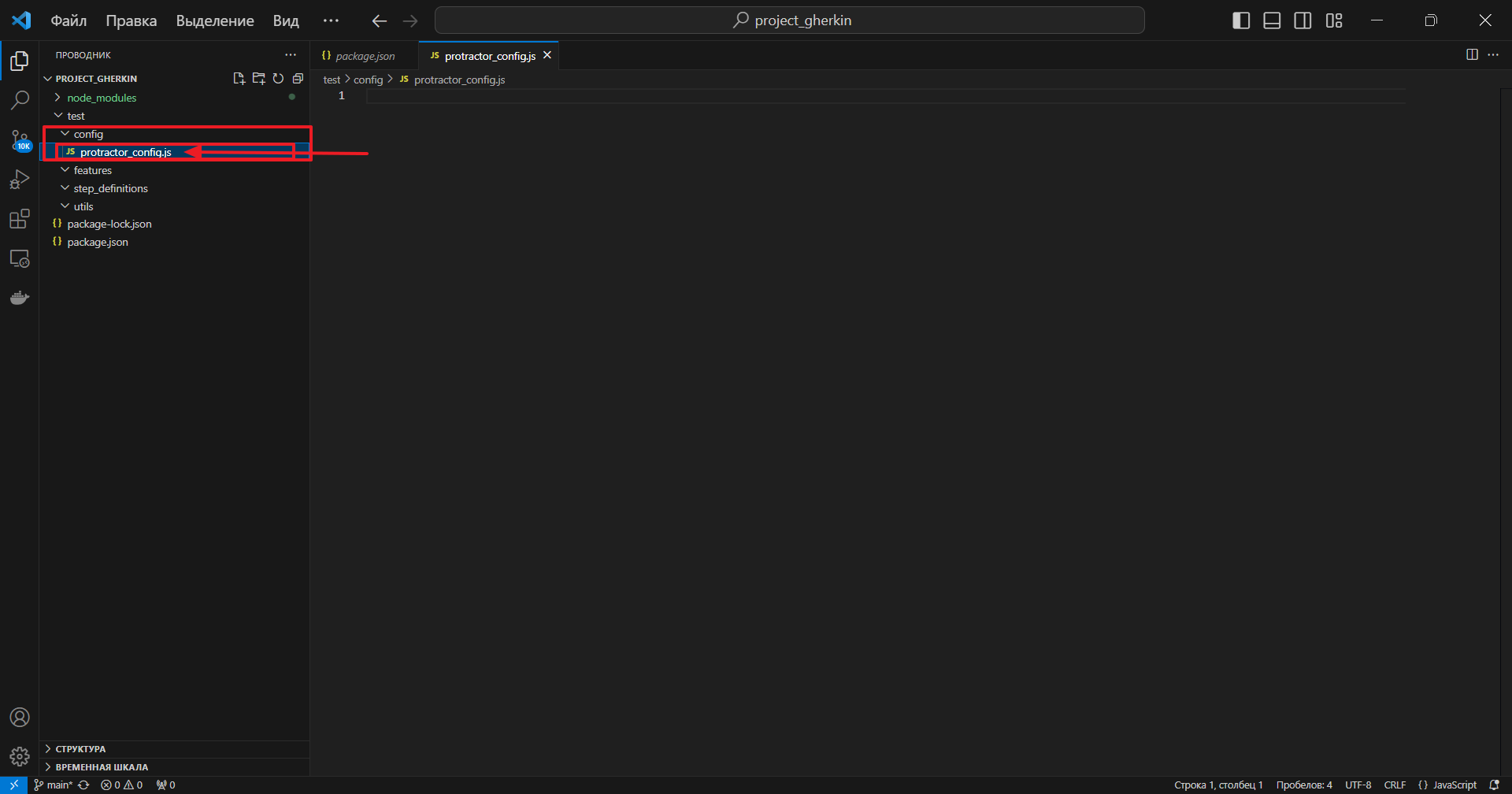
step\_definitions

utils



Конфигурирование работы protractor с cucumber

Открываем папку config и создаем в ней файл protractor\_config.js



**Creating our solution structure and configuration**

Экспортируем конфигурацию

В созданном файле набираем код:

// Подключаем модули встроенные в NodeJS

const path = require('path');

exports.config = {

    // Задаем дефолтное время для protractor

    allScriptsTimeout: 60000,

    // Указываем таймаут на получение страниц

    getPageTimeout: 60000,

    // Указываем место размешения тестовых файлов

    specs: [path.resolve('./test/features/\*\*/\*.features')],

    // Указываем используемый фреймворк

    framework: 'custom',

    frameworkPath: require.resolve('protractor-cucumber-framework'),

    capabilities: {

        browserName: 'chrome',

        chromeOptions: {

            args: ['--no-sandbox', '--window-size=1920,1080']

        },

    },

    disableChecks: true,

    directConnect: true,

    cucumberOpts: {

        require: [path.resolve('./test/step\_definitions/\*\*/\*.js')],

        ignoreUncaughtExceptions: true,

        // вывод отчета

        format: ['json:./test/reports/report.json'],

        // установка тегов

        tags: '@smoke'

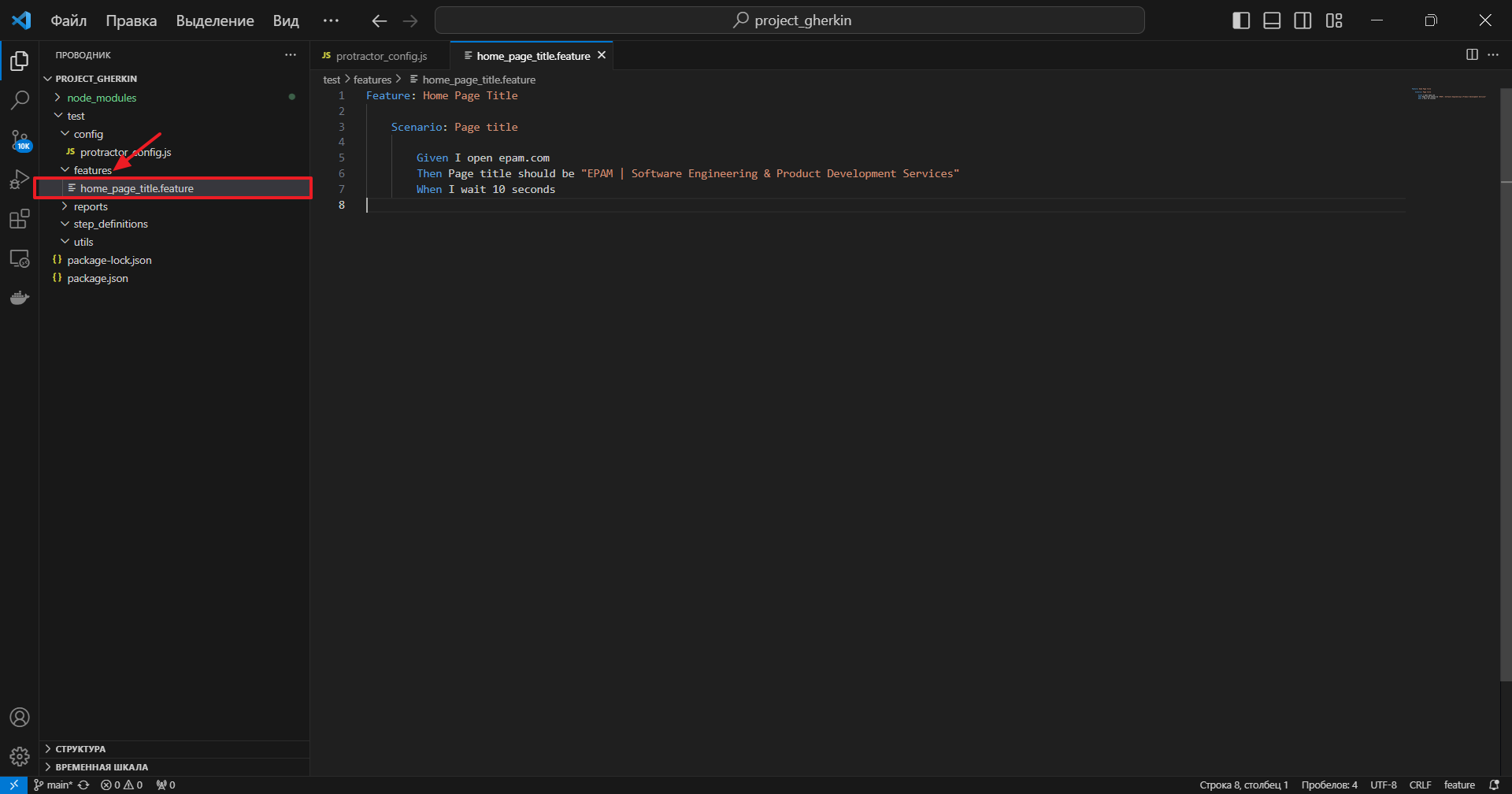
    }

};

**Creating our first feature file and scenario**

Создание .feature файла и первого тестового сценария

В папке features создаем файл home\_page\_title.feature



Feature: Home Page Title

    Scenario: Page title

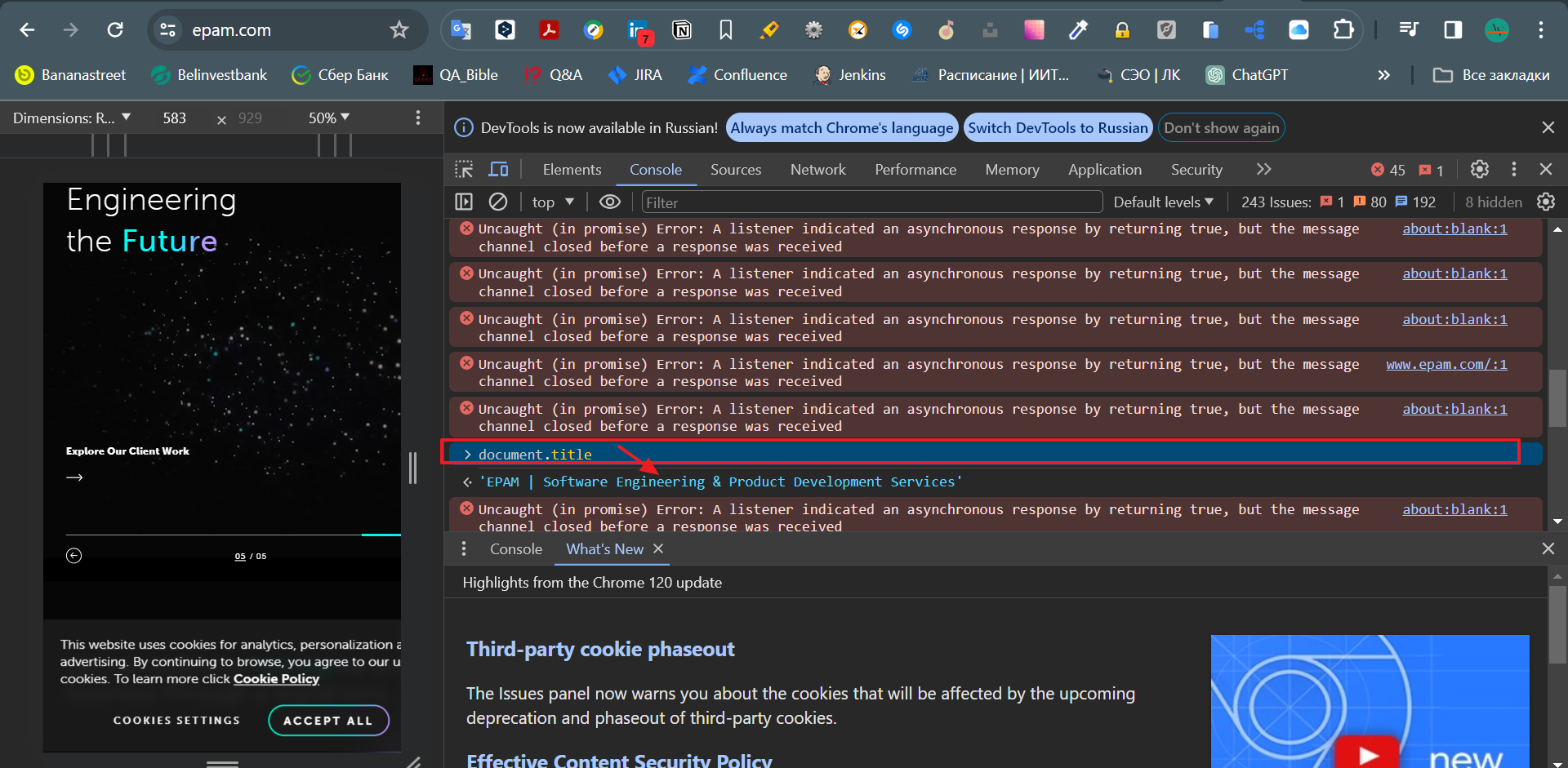
        Given I open epam.com

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        When I wait 10 seconds

Заголовок страницы сайта epam.com ищем через DevTool (F12)

в console вводим document.title

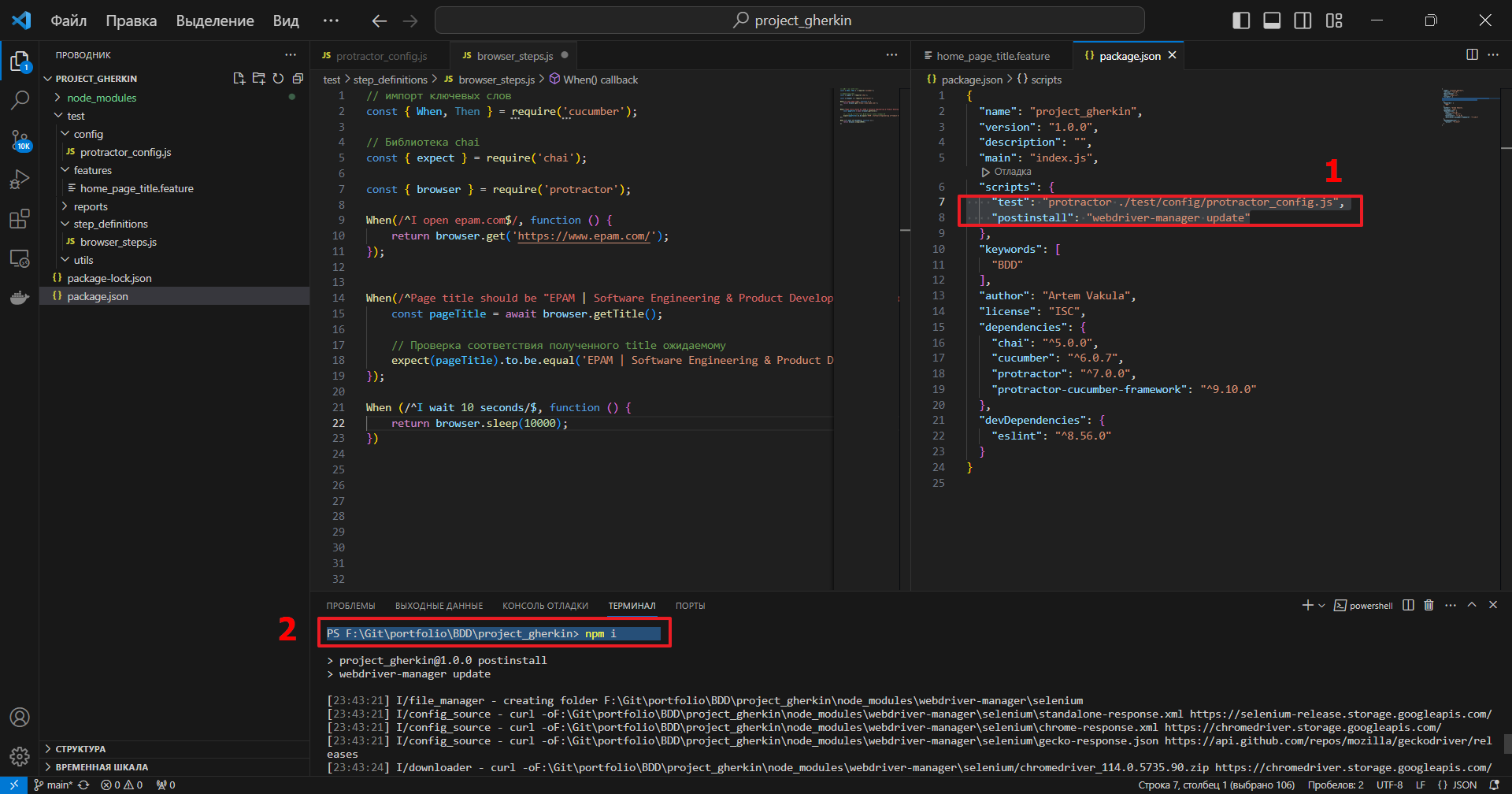


**Implementing our first step definitions**

    "test": "protractor ./test/config/protractor\_config.js",

    "postinstall": "webdriver-manager update"

npm i



**Using variables in step definitions**

Использовать step повторно и более эффективно: передаем на входные параметры через регулярное выражение – желаем капчер-групп

1)

home\_page\_title.feature

@smoke

Feature: Home Page Title

    Scenario: Page title

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        When I wait 10 seconds

browser\_steps.js

// импорт ключевых слов

const { When, Then, setDefaultTimeout } = require('cucumber');

// Библиотека chai

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

setDefaultTimeout(60000);

const { browser } = require('protractor');

When(/^I open "([^"]\*)" url$/, function (url) {

    return browser.get(url);

});

Then(/^Page title should be "EPAM | Software Engineering & Product Development Services"$/, async function () {

    const pageTitle = await browser.getTitle();

    // Проверка соответствия полученного title ожидаемому

    expect(pageTitle).to.be.equal('EPAM | Software Engineering & Product Development Services');

});

When(/^I wait 10 seconds$/, function () {

    return browser.sleep(10000);

});

2)

home\_page\_title.feature

@smoke

Feature: Home Page Title

    Scenario: Page title

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        When I wait "3" seconds

    Scenario: Page title

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

browser\_steps.js

// импорт ключевых слов

const { When, Then, setDefaultTimeout } = require('cucumber');

// Библиотека chai

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

setDefaultTimeout(60000);

const { browser } = require('protractor');

When(/^I open "([^"]\*)" url$/, function (url) {

    return browser.get(url);

});

Then(/^Page title should be "([^"]\*)"$/, async function (title) {

    const pageTitle = await browser.getTitle(title);

    // Проверка соответствия полученного title ожидаемому

    expect(pageTitle).to.be.equal('EPAM | Software Engineering & Product Development Services');

});

When(/^I wait "([^"]\*)" seconds$/, function (timeInSeconds) {

    return browser.sleep(timeInSeconds \* 1000);

});

3)

home\_page\_title.feature

@smoke

Feature: Home Page Title

    Scenario: Page title

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        When I wait "3" seconds

    Scenario: Page title

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

browser\_steps.js

// импорт ключевых слов

const { When, Then, setDefaultTimeout } = require('cucumber');

// Библиотека chai

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

setDefaultTimeout(60000);

const { browser } = require('protractor');

When('I open {string} url', function (url) {

    return browser.get(url);

});

Then(/^Page title should be "([^"]\*)"$/, async function (title) {

    const pageTitle = await browser.getTitle(title);

    // Проверка соответствия полученного title ожидаемому

    expect(pageTitle).to.be.equal('title');

});

When('I wait "{int}" seconds', function (timeInSeconds) {

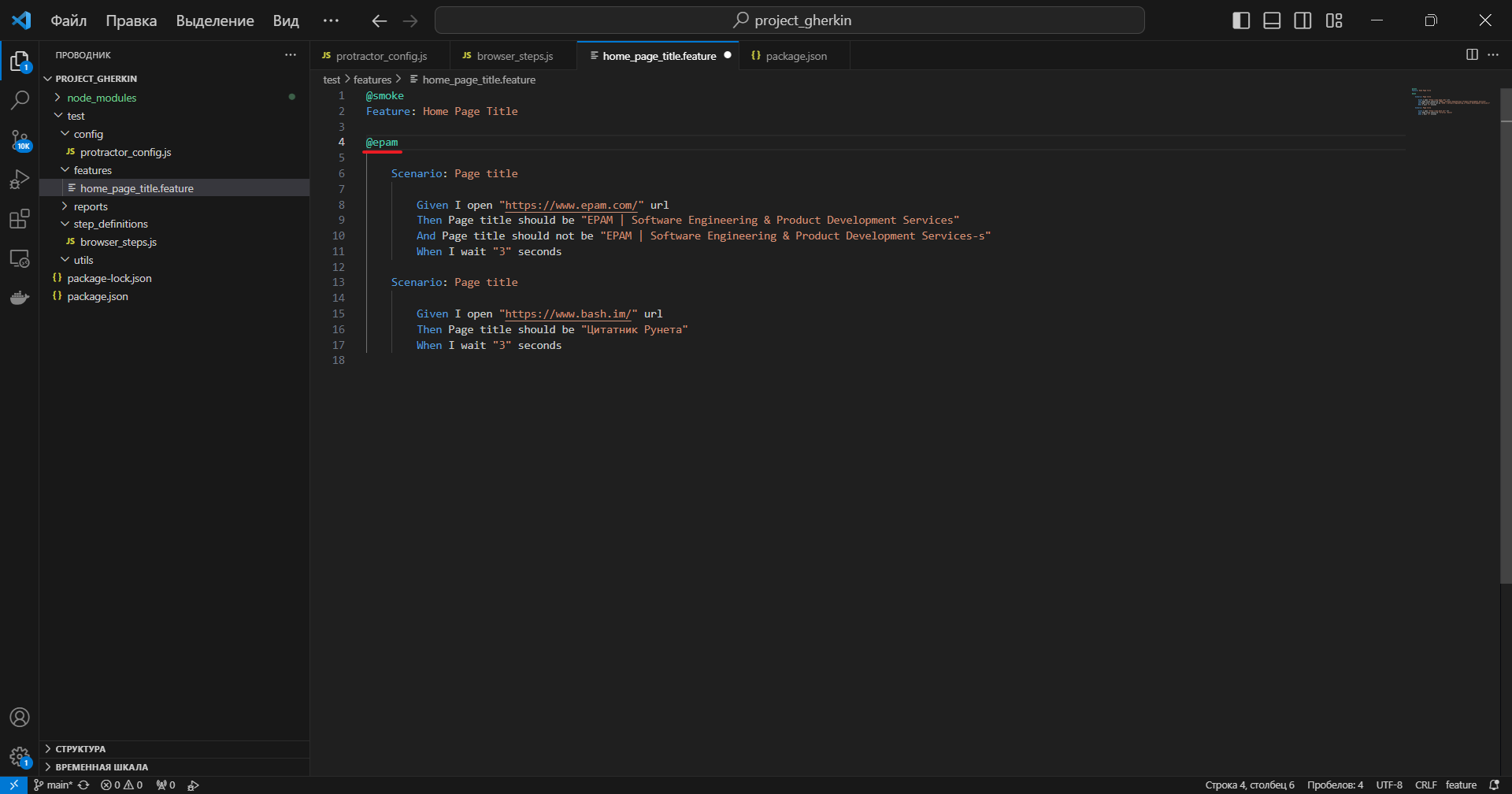
    return browser.sleep(timeInSeconds \* 1000);

});

**Optional parameters in step definitions**

Делаем обратный step

**Tags and tag expressions**



@smoke

Feature: Home Page Title

@epam

    Scenario: Page title

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    Scenario: Page title

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

npm test -- --tags “@epam”

Запуск выборочных тэгов

@smoke

Feature: Home Page Title

    @epam

    Scenario: Page title epam

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    @bash

    Scenario: Page title bash

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

    @medium

    Scenario: Page title medium

        Given I open "https://medium.com//" url

        Then Page title should be "Medium: Human stories and ideas."

        When I wait "3" seconds

npm test -- --tags “@epam or @medium”

Запуск всех тестов за исключением (оператор not)

@smoke

Feature: Home Page Title

    @epam

    Scenario: Page title epam

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    @bash

    Scenario: Page title bash

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

    @medium

    Scenario: Page title medium

        Given I open "https://medium.com//" url

        Then Page title should be "Medium: Human stories and ideas."

        When I wait "3" seconds

npm test -- --tags "@smoke and not @epam"

Операторы ()

@smoke

Feature: Home Page Title

    @epam

    Scenario: Page title epam

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    @mobile

    @bash

    Scenario: Page title bash

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

    @mobile

    @medium

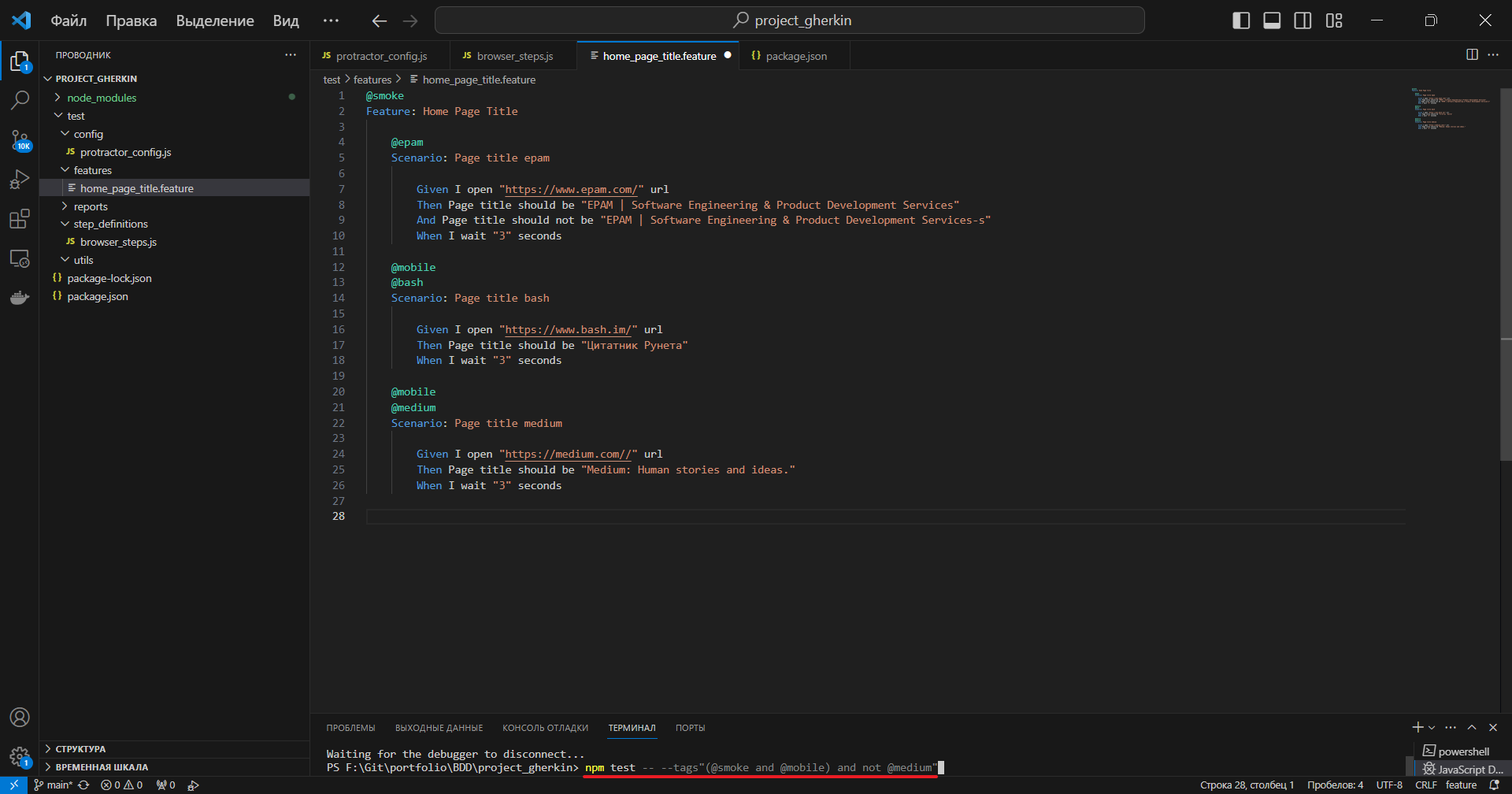
    Scenario: Page title medium

        Given I open "https://medium.com//" url

        Then Page title should be "Medium: Human stories and ideas."

        When I wait "3" seconds

npm test -- --tags"(@smoke and @mobile) and not @medium"



**Improving cucmber console formatting**

Выводим steps из набора, в котором произошла ощибка

npm i cucumber-pretty

**Scenario outline**

home\_page\_title.feature

@smoke

Feature: Home Page Title

    @epam

    Scenario: Page title epam

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    @mobile

    @bash

    Scenario: Page title bash

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

    @mobile

    @medium

    Scenario: Page title medium

        Given I open "https://medium.com/" url

        Then Page title should be "Medium: Human stories and ideas."

        When I wait "3" seconds

    @Sc

    Scenario Outline: Page titles

        Given I open "<URL>" url

        Then Page title should be "<Title>."

        When I wait "3" seconds

        Examples:

            | URL                 | Title           |

            | https://medium.com/ | Цитатник Рунета |

npm test -- --tags "@Sc"

home\_page\_title.feature

@smoke

Feature: Home Page Title

    # @epam

    # Scenario: Page title epam

    #     Given I open "https://www.epam.com/" url

    #     Then Page title should be "EPAM | Software Engineering & Product Development Services"

    #     And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

    #     When I wait "3" seconds

    # @mobile

    # @bash

    # Scenario: Page title bash

    #     Given I open "https://www.bash.im/" url

    #     Then Page title should be "Цитатник Рунета"

    #     When I wait "3" seconds

    # @mobile

    # @medium

    # Scenario: Page title medium

    #     Given I open "https://medium.com/" url

    #     Then Page title should be "Medium: Human stories and ideas."

    #     When I wait "3" seconds

    @Sc

    Scenario Outline: Page titles

        Given I open "<URL>" url

        Then Page title should be "<Title>."

        When I wait "3" seconds

        Examples:

            | URL                   | Title                                                                 |

            | https://medium.com/   | Цитатник Рунета                                                       |

            | https://www.epam.com/ | EPAM            \|Software Engineering & Product Development Services |

npm test -- --tags "@Sc"

home\_page\_title.feature

@smoke

Feature: Home Page Title

    @epam

    Scenario: Page title epam

        Given I open "https://www.epam.com/" url

        Then Page title should be "EPAM | Software Engineering & Product Development Services"

        And Page title should not be "EPAM | Software Engineering & Product Development Services-s"

        When I wait "3" seconds

    @mobile

    @bash

    Scenario: Page title bash

        Given I open "https://www.bash.im/" url

        Then Page title should be "Цитатник Рунета"

        When I wait "3" seconds

    @mobile

    @medium

    Scenario: Page title medium

        Given I open "https://medium.com/" url

        Then Page title should be "Medium: Human stories and ideas."

        When I wait "3" seconds

    @Sc

    Scenario Outline: Page titles <URL>

        Given I open "<URL>" url

        Then Page title should be "<Title>."

        When I wait "3" seconds

        Examples:

            | URL                   | Title                                                                 |

            | https://medium.com/   | Цитатник Рунета                                                       |

            | https://www.epam.com/ | EPAM            \|Software Engineering & Product Development Services |

**Parallelization**

Параллельное выполнение тестов с использованием protractor-cucumber

protractor\_config.js

const path = require('path');

const yargs = require('yargs').argv;

exports.config = {

    allScriptsTimeout: 60000,

    getPageTimeout: 60000,

    specs: [path.resolve('./test/features/\*\*/\*.feature')],

    framework: 'custom',

    frameworkPath: require.resolve('protractor-cucumber-framework'),

    capabilities: {

        shardTestFiles: yargs.instances > 1,

        maxInstences: yargs.instances || 1,

        browserName: 'chrome',

        chromeOptions: {

            args: ['--no-sandbox', '--window-size=1920,1080']

        },

    },

    disableChecks: true,

    directConnect: true,

    cucumberOpts: {

        require: [path.resolve('./test/step\_definitions/\*\*/\*.js')],

        ignoreUncaughtExceptions: true,

        format: ['json:./test/reports/report.json', './note\_modules/cucumber-pretty'],

        tags: yargs.tags || '@smoke'

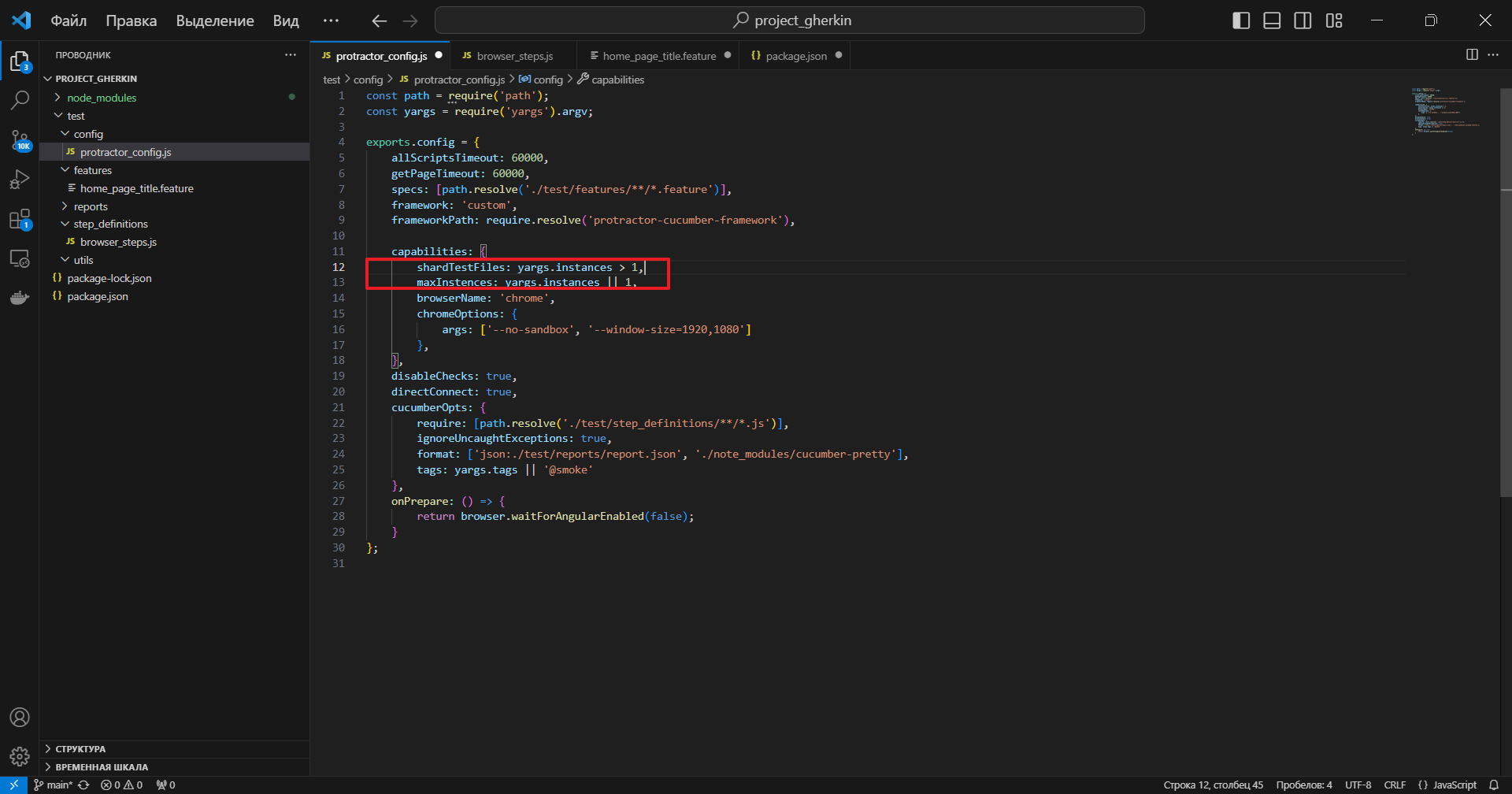
    },

    onPrepare: () => {

        return browser.waitForAngularEnabled(false);

    }

};



        shardTestFiles: yargs.instances > 1,

        maxInstences: yargs.instances || 1,

Вводим в командной строке команду парралелизации

npm test -- --tags "@smoke" --instances 2

**Reporting with Cucumber**

npm i cucumber-html-reporter

**Hooks**

Функции

before

after

before all

after all

2 вида:

перед тегами

после тегами

1) before-hooks

hooks.js

const { After } = require('cucumber');

const { browser } = require('protractor');

After(async function () {

    const screenshot = await browser.takeScreenshot();

    const decodedImage = new Buffer.from(screenshot, 'base64');

    return this.attach(decodedImage, 'image/png');

});

2) after-hooks

hooks.js

const { After } = require('cucumber');

const { browser } = require('protractor');

After({ tags: '@medium'}, async function () {

    const screenshot = await browser.takeScreenshot();

    const decodedImage = new Buffer.from(screenshot, 'base64');

    return this.attach(decodedImage, 'image/png');

});

**Dealing with reports and logs**

npm i fs-extra

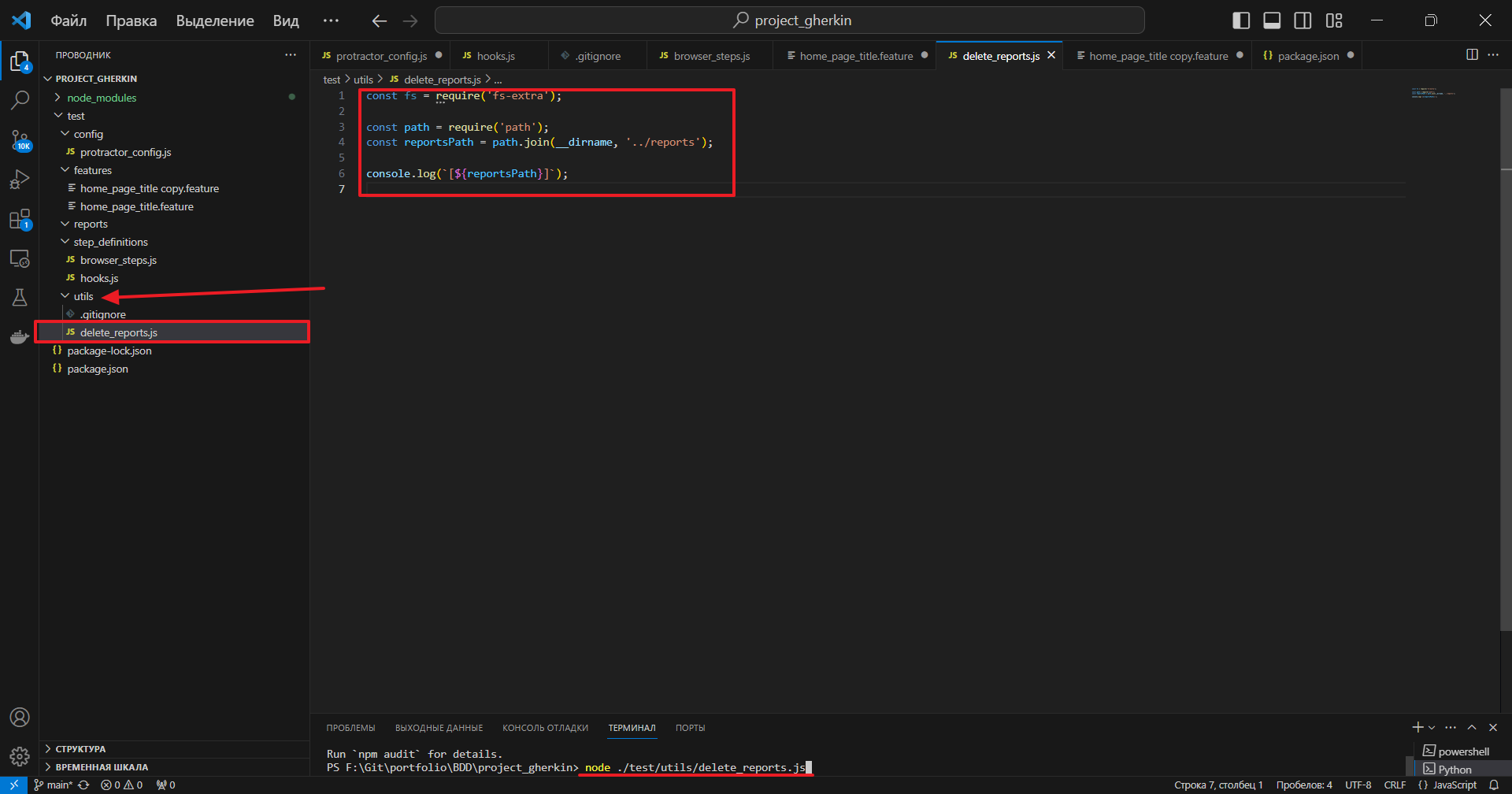
delete\_reports.js

const fs = require('fs-extra');

const path = require('path');

const reportsPath = path.join(\_\_dirname, '../reports');

console.log(`[${reportsPath}]`);



node ./test/utils/delete\_reports.js

delete\_reports.js

const fs = require('fs-extra');

const path = require('path');

const reportsPath = path.join(\_\_dirname, '../reports');

console.log(`[${reportsPath}]`);

fs.emptyDirSync(reportsPath);

В файл package.json добавим строку кода:

"pretest": "node ./test/utils/delete\_reports.js",

package.json

{

  "name": "project\_gherkin",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "pretest": "node ./test/utils/delete\_reports.js",

    "test": "protractor ./test/config/protractor\_config.js",

    "postinstall": "webdriver-manager update"

  },

  "keywords": [

    "BDD"

  ],

  "author": "Artem Vakula",

  "license": "ISC",

  "dependencies": {

    "chai": "^5.0.0",

    "cucumber": "^6.0.7",

    "cucumber-pretty": "^6.0.1",

    "protractor": "^7.0.0",

    "protractor-cucumber-framework": "^7.0.2",

    "yargs": "^17.7.2"

  },

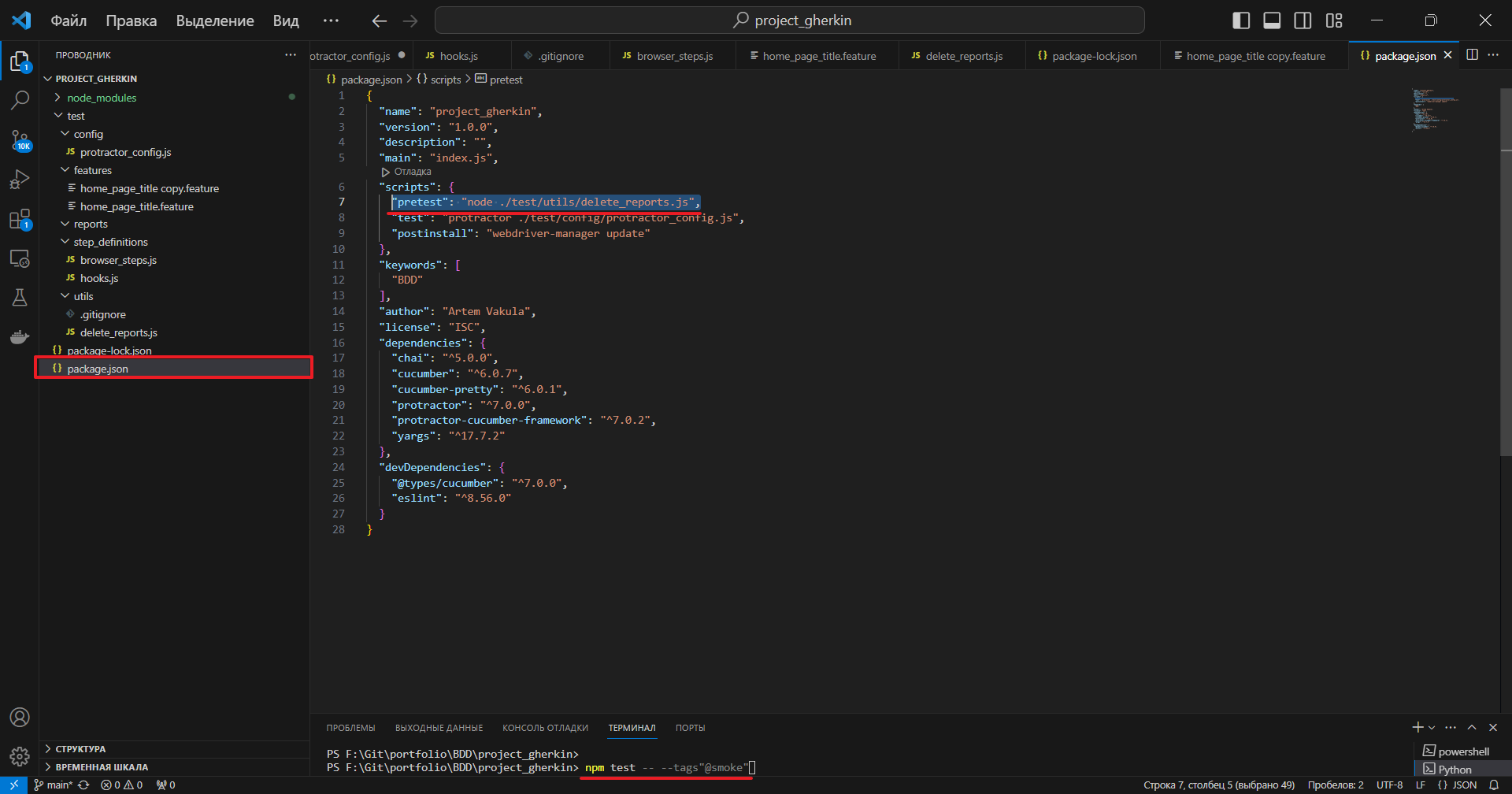
  "devDependencies": {

    "@types/cucumber": "^7.0.0",

    "eslint": "^8.56.0"

  }

}



Запустим тест

npm test -- --tags"@smoke"