86. What is Cucumber Framework ? How it fits in Playwright needs

Google (cucumber.js) Based upon the language you use. You need to select that particular jars to work on that.

Cucumber framework this is just an another interface top over your existing script.

Now this is your test which does client login, submit the order and verify the order end to end test. So the same test, we will describe it in plain English. So we call it as a gherkin language.

Google ( cucumber.io )

if anybody want to understand what exactly you have written, we will go through that feature file and you blindly run this file.

When you run this file, this particular code will get executed.Underlyingly.

We are still running the same code, but what are we doing is we are just having another nice interface top over it to describe our all automation into nice English code.

(use of this cucumber framework is - We are adding top over it to look it like a plain English so that everybody can understand what kind of automation test you have written )

Google( cucumber.js github ) ( <https://github.com/cucumber/cucumber-js> )

So previously what testing framework you have used to run all this? Playwright with mocha.

Mocha is the framework we have used as a helper.

Now if you go to playwright configuration file here we have timeouts and test retries, screenshot, all that, what reporting you need, Everything. We are depending on the playwright with mocha combination, but

when cucumber comes into picture and when you try to automate playwright with cucumber, the problem is you cannot use mocha in cucumber because both are serving the same purpose.

so when they are serving the same purpose, when you try to use them together, then there will

be a conflict. So for that reason, when you go with cucumber, you will have your own way of data driving your own way of using tags, your own way of defining timeouts and retry failed test taking screenshots and test failure parallel running the test.

All these whatever you saw until now can be achieved differently in Cucumber.

87. Cucumber Configuration and setup feature files into Playwright Project

Step 1:

Google ( <https://github.com/cucumber/cucumber-js> ) in this path

copy the cucumber installation command ( npm install @cucumber/cucumber )

execute this command in playwright after excute go the package.json file check in the dependencies section ( "@cucumber/cucumber": "^10.3.1" ) it should be add automatically after run the command.

step2 :

make sure you have cucumber plugin which is exposed by Visual Studio code.

Click on extensions on the left side icon in your VS code.

( Cucumber (Gherkin) Full Support ) install this and restart ur system.

One you need to install cucumber packages. Two, Cucumber js comes as one node module.

So that node module you are setting up in your project with this command and just have one plugin, So with that you are done with all the necessary setup.

( <https://github.com/cucumber/cucumber-js> )

cucumber feature is more of like test suite. A test suite can contain multiple test cases.

It's like a folder which can hold multiple test cases.

So each scenario we treat as one test case in cucumber, that means a one feature file.

If you want to write one file, you need to give this extension dot feature.

One feature file can hold multiple scenarios. That means a multiple test cases can be written in one single feature file. So these are the keywords. ( Feature, scenario,when,then)

So in project level you need to create one folder called features right click new folder features.

When you have like this, cucumber recognizes it because cucumber by default checks for this folder in your project path. So inside this you can write all the code what you need for cucumber to identify.

Create one new file(Ecommerce.feature) under the features folder

Feature: Ecommerce validations

  Scenario: Placing the Order

  Given a login to Ecommerce application with "anshika@gmail.com" and "Iamking@000"

    When Add "Zara coat 3" to Cart

    Then Verify "Zara coat 3" is displayed in the Cart

    When Enter valid details and place the Order

    Then Verify order in present in the OrderHistory

88. Create Step Definition files and associate with Cucumber.js config file

Now to generate corresponding steps Skeleton. How would you do that?

Without giving any code if you run the above feature file it will ask you how would I run In that process it will give you the all the skeleton mentioning that implement this line, this line,

all that structures.

It will give in the output saying that I cannot run until you implement all those step definitions.

By taking that log information, that's where it generates that skeleton by complaining that these are missing.So you can simply take that copy paste skeleton and place it in the new file.

By default, it won't generate you will you have to run this feature file. And then Cucumber complains that those are missing. That's how we will capture those default skeletons.

So in Cucumber to execute any scenarios are feature files,

Run cucumber ( npx cucumber-js )

 npx nothing but it will try to find the cucumber executable file in the node modules.

So as you installed NPM cucumber, go to your node modules bin directory and here is the your executable file. So basically you need to run this file.

So when you run this file, what happens is it will scan your entire project and it will see if there

is any features folder present. If it's present, it will go inside the folder and it will again see if there are any files with a dot feature extension. If it finds any, it will blindly execute that.

That's the concept.

After the run the cucumber command create one new folder(step\_definitions) under the features folder inside this step\_definitions folder create one file (steps.js)

Import the packages

const { When, Then, Given } = require('@cucumber/cucumber')

and add the skeleton template

If you write in double quotes, Cucumber treats it as a dynamic data and what it does is here it will place like this in curly braces.

That means this data, what we are getting in this spots is a dynamic and this can be changed anytime.

89. Implement Code login into StepDefinition file and run Cucumber

const { When, Then, Given } = require('@cucumber/cucumber')

const {POManager} = require('../../Pageobjects/POManager');

const {test, expect, playwright} = require('@playwright/test');

Given('a login to Ecommerce application with {username} and {password}', async function (username, password) {

    // Write code here that turns the phrase above into concrete actions

    const browser = await playwright.chromium.launch();

    const context = await browser.newContext();

    const page = await context.newPage();

    const poManager = new POManager(page);

     const products = page.locator(".card-body");

     const loginPage = poManager.getLoginPage();

     await loginPage.goTo();

     await loginPage.validLogin(data.username,data.password);

  });

  When('Add {string} to Cart', async function (string) {

    // Write code here that turns the phrase above into concrete actions

    const dashboardPage = poManager.getDashboardPage();

     await dashboardPage.searchProductAddCart(data.productName);

     await dashboardPage.navigateToCart();

  });

  Then('Verify {string} is displayed in the Cart', async function (string) {

    // Write code here that turns the phrase above into concrete actions

    const cartPage = poManager.getCartPage();

    await cartPage.VerifyProductIsDisplayed(data.productName);

    await cartPage.Checkout();

  });

  When('Enter valid details and place the Order', async function () {

    // Write code here that turns the phrase above into concrete actions

    const ordersReviewPage = poManager.getOrdersReviewPage();

    await ordersReviewPage.searchCountryAndSelect("ind","India");

    const orderId = await ordersReviewPage.SubmitAndGetOrderId();

   console.log(orderId);

  });

  Then('Verify order in present in the OrderHistory', async function () {

    // Write code here that turns the phrase above into concrete actions

    await dashboardPage.navigateToOrders();

   const ordersHistoryPage = poManager.getOrdersHistoryPage();

   await ordersHistoryPage.searchOrderAndSelect(orderId);

   expect(orderId.includes(await ordersHistoryPage.getOrderId())).toBeTruthy();

  });

90. What is World Constructor? Its usage in Playwright Cucumber

whatever variables you access here, those can be accessed in any step of the scenario

So one way is to make it global variable, declare on top and make it access.

So without using global variables, also you can talk between your scenario step definitions.

so all these are now interconnected.

You can talk to each other and share your variables happily without even declaring them in globally. That's what these guys have brought this constructor.

Const poManager to this.PoManager

what happens is when you use this dot, this world constructor gets activated and it will place in its world.

And again, if this PoManager, if you are trying to access any other method, as long as that method stays in this scenario, world constructor is happy to share.

Now what happens is scope(this.POManager)will not end here.The scope will carefully stored in the world constructor and whoever asked this, they will keep sharing.

In step.js file where every you see poManager replace with this.poManager

Const dashboardpage replace with this.dashboard

So by default, Cucumber tries to execute your method in five seconds.

If it could not find in five seconds, your test will fail.

So that's why whenever you write any method, make sure that method piece piece of information gets executed. Not more than five seconds.

so we can override the default timeout 10 sec by simply saying like this - {timeout : 100 \*1000}

npx cucumber-js –exit - of you use once the entire execution is done it come out of the loop.

In the terminal it is shown one block after executing if you want to disable it in project level create one file(cucumber.js) in the what add this (module.exports = {default : ‘--publish-quiet’}

If you want to run your step.js file in headless mode then

const browser = playwright.chromium.launch({

        headless : false

    });

91. Understand Cucumber Hooks and its implementation process in framework

what are hooks in cucumber?

So it's just like setting up the prerequisite data and the tear down data in your test case.

we have already set up these hooks in playwright Mocha framework.

Now, if you go to any test, there is something called before all.

so that means this executes before executing any test. So you can do data setup here,

So in similar way, Cucumber also has something called before and after annotations where you can actually make some data setup or delete the data again after the test case is completed.

( <https://github.com/cucumber/cucumber-js?tab=readme-ov-file> )

Created one folder under the features ( support ) write your hooks inside this support folder.

Inside this support folder create one file( hooks.js )

So you can write one hook saying before each scenario that go and login and then start with your actual test case.

There is one difference between before all and before.

So when you use before annotation, it executes before every scenario.

If you have five scenarios, it will run five times before each and every scenario.

But if you use before all that means it runs only once before all the five scenarios.

And again, it won't run for each scenario. It just runs once.

And after all, Will runs run once after all scenarios are executed.

In hook file

const { Before,After,BeforeStep,AfterStep,Status } = require('@cucumber/cucumber')

const {POManager} = require('../../Pageobjects/POManager');

const playwright = require('@playwright/test');

Before(async function () {

    const browser = await playwright.chromium.launch({

        headless : false

    });

    const context = await browser.newContext();

    this.page = await context.newPage();

    this.poManager = new POManager(this.page);

  });

  BeforeStep(function () {

    // This hook will be executed before all steps in a scenario with tag @foo

  });

  AfterStep( async function ({result}) {

    // This hook will be executed after all steps, and take a screenshot on step failure

    if (result.status === Status.FAILED) {

      await this.page.screenshot({ path : 'screenshot5.png'});

    }

  });

  After(function () {

    console.log("I am last to execute");

  });

92. Implement Cucumber Tags for features and also demo on Tagged hooks to filter

Create one new file (ErrorValidations.feature) under the features folder

Feature: Ecommerce validations

  @Validations

  @foo

  Scenario Outline: Placing the Order

  Given a login to Ecommerce2 application with "anshika@gmail.com" and "Iamking@000"

    Then Verify Error message is displayed

 if I write this step all the feature files, whatever you have in your project will run. So can I run only specific feature file I'm interested in?

npx cucumber-js features/ErrorValidations.feature –exit

So tomorrow in your framework, there could be some 100 positive end to end test cases, all where you can tag with regression and all the error validation scenarios. You can use a tag called validation.

Now you have a requirement that you need to run only the positive end to end test cases.

So that means you need to release or you need to certify your QA in just ten minutes.

So that means you cannot run everything, You need to choose a set of test cases to tell how your application is. So in that case, let's say you just need to run with all test cases which are tagged with @regression and exclude remaining.

So that's how you can tell from your terminal.

If you want to add the tagname for few of the testcase then @Regression like then in the terminal you write command like ( npx cucumber-js –tags “@Regression” )

So what does Cucumber does is it will go and find all the scenarios which are tagged with regression and it will run and execute only those scenarios and it will skip all the remainings.

That's what tagging means.

Scenario can have N number of tags Now you can tell that run this block only when that scenario having two tags.(and )

(or) means if any one tag is there that is enough

93. Parameterization with Scenario outline & run tests Parallel in Playwright

Step 1 :

So first of all, if you are planning to parameterize your scenario with the different data sets instead of scenario, then you have to write scenario outline. This is the keyword.

This keyword represents that this guy is trying to run this scenario with a different data sets.

If you don't put this outline, then this cucumber cannot recognize that this is a parameterized

Step 2:

you have different data sets here. You need to write all those data sets in this syntax. You have to start with examples

Feature: Ecommerce validations

  @Validation

  @foo

  Scenario Outline: Placing the Order

  Given a login to Ecommerce2 application with "<username>" and "<password>"

    Then Verify Error message is displayed

    Examples:

    |   username       |    |   password   |

    |anshika@gmail.com |    |Iamking@000   |

    |hello@123.com     |    |Iamhello@12   |

So like that two times it will run with two different data sets.

And every time it picks the data and it replaces it, runtime and execution will be done.

npx cucumber-js --tags "@Validation" --exit

94. Generate HTML reports for Cucumber Playwright & Rerun failed scenarios

cucumber can only run scenarios in parallel, not in feature files.

so here you have two feature files. You cannot run these two files. Parallelly.

That is like limitation with cucumber, but you can run multiple scenarios. Parallelly.

If you think you can group this feature file with N number of scenarios, go ahead and write it.

All those scenarios, whatever you define in one feature file can run parallelly.

In this particular feature file, now we can run these two scenarios. Parallelly. That is allowed.

but we cannot run files. Parallelly.

(npx cucumber-js features/Ecommerce.feature –parallel 2 –exit )

Generate HTML report

npx cucumber-js features/Ecommerce.feature –parallel 2 –exit –format html:cucumber-report.html

rerun the failed TC’s

So sometimes we might have some scenarios, some test cases which might behave flaky, that it might get passed on the subsequent rerun.Maybe because of environment issues or some issues, it could turn as flaky, but it will immediately pass.

When you rerun again, if you have such scenario, then you can give this option hyphen hyphen retry one.

npx cucumber-js --tags '@Regression' --retry 1 --exit --format html:cucumber-report.html

npm run CucumberRegression