166. Creating Base Test which holds Common Test configuration methods.

Creating Base Test which sets browser configuration details & global properties

Abstract component file - So whatever they want, they will take it from here bcze this is a file which exclusively store all reusable content. And this is for page objects.

lly, for test cases also, take one base test. which holds all the reusable things,

like setting up ChromeDriver, FirefoxDriver, or invoking the driver browsers, this is commonly used in every test, These four lines of code, that's setting up waits, maximizing the browser, and all these will be used in every test.

if you have hundred test cases, all the test cases will simply call that method

to initialize this, everything, instead of, you know, rewriting all these lines of code in each and every test.

Note : we have written page object files in the main Java bcze that is not related to test.

These page objects just give you the locators and action methods which needs to be performed. But there is no test inside this. That's why these files packages went in source main Java.

Now we are dealing with here base test related files. Test components.

Src/test/java -> create new package (rahulshettyacademy.TestComponents) -> finish

Create new java class (BaseTest)

Create one method in BaseTest - all initialization of driver, what you need for your test, everything, we can write in that method

technically when you're developing a framework, you could have worked in multiple browsers, So you can maintain one global properties file, where you store which browser you want to execute, based upon the global property you set.

If you set global property to Firefox, then the whole framework will run in Firefox browser.

how to set up these global properties ?

in Java there is one class called properties. So that properties class can read the global properties. and decide at runtime, on which browser your framework has to execute test cases.

Under main java create another package ->(rahulshettyacademy.resources)

Create a new file inside this package -> ( GlobalData.properties ) -.Finish

Remember that the file name, what you are creating, should have .properties extension.

The reason, If you write any file name with .properties extension, then using properties class in Java, you will be able to pass this file and extract all global parameter values.

In global properties file just write - browser=chrome

in Java there is a class which comes in util package,I just created properties object for it.

Properties prop = new Properties();

prod.load(fis);

.load() – here global data .properties file should upload. When you load, this method will automatically pass that properties file and extract all key value pairs from it.

how you can send this file as an input stream ?

if we write prod.load() in that it is asking to accept only inputstream that why we added this line.

in java there is one more class

FileInputStream fis = new FileInputStream(“paste the global properties file path”);

String browserName = prod.getProperty(“browser”);

whatever it is set in global properties, using that browser only,we want to execute our whole framework.

Global.Data.Properties file ->Right click -> properties -> copy the path

there is one method called system dot getProperty, so here you have to simply give

user dot dir. ( to reduce the link size )

( FileInputStream(system.getproperty(“user.dir”)+”//src//main//java//rsa//resources”)

So make sure you put double slashes .so that Java recognizes them as a path.

Public WebDriver driver; // global level

Public void initializeDriver()

{

Properties prop = new Properties();

FileInputStream fis = new FileInputStream(“paste the global properties file path”);

prod.load(fis);

String browserName = prod.getProperty(“browser”);

If(browserName.equalsIgnoreCase(“Chrome”))

{

WebDriverManager.*chromedriver*().setup();

driver = **new** ChromeDriver();

}

Else if (browserName.equalsIgnoreCase(“edge”)) {

System.setProperty(“webdriver.edge.driver”,”edge.exe”);

WebDriverManager.*chromedriver*().setup();

driver = **new** ChromeDriver();

}

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

driver.manage().window().maximize();

return driver;

last two line I write outside bcze above two line are common for all the browsers

167. Initialize Driver and create utility to launch App with BeforeMethod anottaion

when you are working on an app - no matter what test you run, you have to first get onto this page(url)

create one method in our base test. Launch application.

Reason - if you have hundred test cases for one app all the test cases, first step is

they want to create all this browser driver everything and they have to land on landing page.

create one method, which is commonly applicable to all the test cases

delete the landing page 2 lines of code in main class

public void Launchapplication(){

driver = initializeDriver()

LandingPage landingpage = **new** LandingPage(driver);

landingpage.goTo();

return landingpage

}

In main class

Now we don't require static void main let's use actual @Test

it is returning landing page object, right? I'm catching that into one variable

@Test

Public void submitorder()

{

LandingPage landingpage = Launchapplication(); // add this line in main code

}

So now it's a TestNG test. It's no more Java. It's still Java only, but the framework

what we are using to run is based on testNg runner.

168. Create new Error Validation Test as per framework Standards developed until now

create this object on top

public LandingPage landingpage;

@Before Method

public void Launchapplication(){

driver = initializeDriver()

landingpage = **new** LandingPage(driver);

landingpage.goTo();

return landingpage

}

@After Method

Publc void teardown(){

Driver.close()  
}

incorrect email ID password. But if you carefully observe when I right click and inspect,

you see another class is coming and that is getting disappeared when error is also gone.

So we have to catch that, another class.

169. Implement Test Strategy for Framework on how tests are divided based on modules

Note : you can take around 20 Java classes and you see how you can club all 100 test cases

into these classes.

each method represents one test case .If you mark them with @test annotation.

Now, in the login page, you have five test cases which are specifically working

on error validations, then I will include those five tests in one particular file. Because here it represents one file, represents one module or one functionality.

Let's say you are working on sprint number two they'll just put Java class name as sprint two

and all tests they'll write in that sprint two.

So all the negative related scenarios, I'm wrapping up in one test called error validations

and all positive tests I am writing in the submit order.

take multiple email IDs bcze, when both tests run together/parallelly

at a time on same account and trying to add same product, there would be overlap and, you know, there could be issues. So just make sure you give unique account, so that way you don't see the any cart missing items here.

So all the three tests, I want to run together. there is TestNG XML file,

where you can give all the entries of these two files

and that will trigger your test cases one by one.

how to create TestNG XML file as a test runner for this framework ?

right click on your project, click on TestNG, convert to TestNG,->next->finish

And all the three classes are wrapped under one test. So instead, what I will do, for each test entry, let's put one class.

I recommend you to create one entry of test for each Java class.

All three will run with one single click for running this TestNG.xml.

170. Create new Test methods with Dependency attribute based on Test strategy design?

how to write dependency tests ?

sumitorderTest -  I want to write another test to verify if that Zara Coat 3 product is displaying in the orders page.

Now one functional validation is done of submitting the order, now there is another functional validation where you want to verify if that product is sitting in orders.

change to dependsOnMethods and Now what happens is when you ask to run this test,

first it'll make sure that above test should run After that only this test will execute.

You cannot run this now test individually. So that's why I'm calling this a dependent test.

In abstract component just take the orders button element and place it there.

@FindBy(Css=”[routelink\*=’myorders’]”)

WebElement order Header

Create one method in the orders page

**public** CartPage goToOrderpage() {

OrderHeader.click();

OrderPage orderheader = **new** OrderPage(driver);

**return** OrderPage;

}

For every new page we are going to create one classwhere all the objects of that page goes to this class.

but as of now we have not built any order page class name. so we will create one class.

@Test(DependsonMethods={“submitOrder}}

Public coidOrderHistioryTest()

{

ProductCatalogue productcatalogue = landingpage.loginapplication("harisankar722@gmail.com", "H@rish777");

OrderPage orderPage = productCatalogue.goTOOrdersPage();

Asser.assertTrue(orderpage.VerifyOrderDisplay(productName));

}

171. How to run tests/classes in Parallel & apply Groups using TestNG.xml

So basically from TestNG perspective, there are two tests here. Each test to take one single class.

If you want to run test from SubmitOrderTest and ErrorValidationsTest, both parallelly.

That means you are intended to run both tests parallelly.

In that case, on suite level you can just say parallel equals to tests, that's all.

we have our two files. So that means two tests will run parallelly at a time. That means you will see two browsers open.

So one attribute set in our framework will help us to run tests in parallel. you have 20, 30 tests in your framework. Making sure to run them parallel will really help you to speed up your execution

not only just test, you can run methods also parallelly.

… now let's say this method and this method in the same file also you want to run parallelly,

then you have to select a parallel equals to methods. not only Java files will run parallelly,

the methods present inside the file also will run parallel. Now there is another attribute here

called thread-count of 5. So that means maximum, it will allow five methods to run parallelly

from this Java class file.

how to execute selective tests - using groups - use of group is if we want to run only the specific Tc’s in few on the classes there we will use groups. ( @Test(groups= {"ErrorHandling"})

Note: if you add BeforeMethod are afterMethod(alwaysRun=true) then if you method any specifc group also it should run before