

BDD Framework:

BDD stands for Behavior Driven Development framework. We can achieve BDD framework with the help of different tools such as Cucumber, JBehave and SpecFlow but in my current organization we achieve BDD framework with the help of Cucumber tools and java platform.

First we create maven project, at the time of creating maven project we have to mention group id and artifact id. Group Id is nothing but the package name and artifact id is nothing but the Project name. Once we create the maven project it will generate two main source packages as like src/main/java and src/test/java and there is one most important XML file is pom.xml.

POM.xml is hear of maven project inside the pom.xml file here we add different types of dependency as like selenium maven dependency, test-NG maven Dependency and Cucumber-test-NG dependency. For selenium maven dependency her we use 3/141.59 or 4.1.0 or 4.1.2 version, for test-NG maven dependency (6.8.0 or 7.4.0 version and for Cucumber-maven dependency here we use 6.8.0 version, For generating extent report here we use 5.0.7 version.

inside the src/main/java we create different types of packages as like 1st like Base Layer package, 2nd Page Layer package, 3rd is config layer package, 4th is test data layer package and 5th is utils layer package and inside the src/test/java we create 2 different packages as 1st is Step Definition package and 2nd is test Runner package, and to maintain the all feature files we create Feature file folder , and to maintain pass and fail screenshot we create 1st Pass Screenshot folder, 2nd is fail screenshot folder, and to maintain the driver we create drives folder and to maintain reports we have created reports folder.

There 3 main file presents in cucumber tool, 1st is Feature File and 2nd is Step Definition file and 3rd is Test Runner file.

For maintain all the feature file here we create feature folders. Inside the feature folder we can create multiple feature file as per the project requirements wise but extension of each and every file is dot feature. Inside the feature we use different types of gherkin keywords such as feature, scenario, scenario outline, given, when, then, and, but,* background and Examples keyword.

Inside the feature file first keyword must be Feature followed by Colon. And then we write short description of feature, If you want to achieve normal scenario then we are use Scenario keyword followed by Colon and then we write what is exact reequipments in short line, scenario is nothing but the what to test or what is your exact requirements. And if you want to achieve data driven

testing in cucumber then we use Scenario Outline keyword followed by Colon and then we short description of Scenario outline and at end of scenario steps we write examples keyword followed by colon and there we maintain multiple set of data test data with the help of pipe symbol.

Then we are write scenario or scenario outline steps with the help of Gherkins keywords, there are different types of gherkin keyword available to write the Scenario steps, as like Given keyword is used to mentions preconditions, then When keyword are used to describe an event, or an *action*, *then* Then keyword are used to describe an *expected* outcome, or result, then And keyword is used to combine the 2 different steps, then But keyword are used declare multiple conditions and asterisk (*) global keyword which is used in place of any of the normal step keywords. We can use background keyword followed by colon to mention precondition for each and every scenario in feature file.

Then if create snippets with the help of Feature file or test runner class,

Here we create one package inside the src/test/java as like TestRunnerLayer Package inside this package here we create test runner class. Inside test runner class here we use some class level annotation as like @RunWith() annotation – here @RunWith() annotation tell to Junit to run the test cases with the help of cucumber class. Then next we use @CucumberOptions() annotation: this annotation tell multiple things as like where is feature file location, we declare feature file location with the help of features keyword then we have write path of feature file location, where is Step definition file location here we declare Step Definition file location with the help of glue keyword or extraGlue keyword then we mention step definition Package location, then if you want to checking mapping between feature file step and implemented steps in step Definition class then we use dryRun=true attribute and we can generate different reports by using cucumber as like JSON, XML , text and html report with the help of plugins keywords.

Again inside the source main java package here we declare multiple packages as like

inside the Base Layer package, we create Base Class and this Base class is super parent class of the classes in framework, here we use inheritance concept. inside the Base Class we create static method to mention basic property as like how to connect to the browser basically we connect to the browser by using System.setProperty()method then here we mention up casting concept, as like reference of WebDriver interface and Object of child class that is ChromeDriver class, basically by using above up casting concept we can call all the methods from the WebDriver interface then we mention implicit wait concept because implicit wait it applicable for all the element in webpage or

web object. implicit wait is also known as global wait if we declare implicit wait as 30 sec and if element or object find within the 2 sec then it will ignore remaining all sec and it also known as dynamic wait, then we mention page load time out it is applicable for all the browser related properties and then we write code for maximize the browser and also we mention delete All Cookies code to clear all the cookies and then we get() method to open a specified URL and basically we mention actual URL inside the config layer package inside the config layer package we create dot properties file inside the properties file, we mention different environments URL, Username, password, browser name and database details etc. and we read this properties file inside the Base Class. if u want to work with properties file then first we have to create object of Properties class, then we have to read the all the data from the properties file then we create object of FileInputStream class by passing file path. then we have to load this file inside the class then we use load() method from the Properties class and u want to get single property value from the Properties file then we use getProperty() method by passing property key name.

In Page layer package we create container classes for each and every page in application. inside the page layer package first we extends the base class, already i have already told that base class is super parent class of all the classes in framework, inside the page layer package we create container classes and inside the each and every classes there we create object repository with the help of POM with Page Factory, POM means it is page object model it is a design pattern to design the Object Repository. basically, we can create Object Repository with the help of 2 different ways as like page object model with page factory and page object model without page factory. in my current framework we create object repository with the help of POM with page factory at the time of creating Object repository we use @FindBy() annotation. These @FindBy annotation is used to find the element or object in web page and once object is find then it will return the Web Element and we store these all object as global as private variable here we are using encapsulation concept from java OOPs and once we create object repository then we have to initialize the object then we have to use Page Factory class and from Page Factory class we are use inti Elements method by using WebDriver instance name and this keyword, this keyword is used to access the all global variable of current class and inside the same class we declare the associated functionality methods of the all object repository and as per the requirements wise we are calling utility layer package method in Page layer package.

Inside the utility layer package there we have to created multiple reusable script code classes as like to handle drop we have created handle drop down class, to handle alert pop there we have to created Handle Alert class, to handle the frame we have to created handle frame method, to handle the window we have to created handle window class, to perform keyword board and mouse event we have to created one more class as like Actions class, to handle dynamic calendar we have created utils class inside the utils class we mention all reusable script code, to generates the extent reports we have created extent report set up class and we are configure these extent report set up class to I test listeners and to run the failed test cases we have to created class as like Retry analyzer class and Retry transformer class. And to read the file we have to crated one more class as excel reader class.

Inside the handle drop down we have created different methods , basically we can select drop down value by using 3 ways, 1st is by using select by visible text method then 2nd select by index method and 3rd by using select by value methods and if u to verify which value is selected then we use get first selected options method then we use get text method to capture the values and if u want to check how many value is present then we use get options method and then we use size method to count size of drop down and if u want to print all drop down values then we use get options method and we iterate the loop.

Then we have created Handle alert pop class, inside this class we have created 4 methods to handle the alert pop, if u want to click on ok button we have to use accept method and if u want to click on cancel button then we have to use dismiss method and if u want to capture alert pop text then we use get text method and if u want to send text to alert pop then we use send keys method.

Then next we have to created handle frame class which is used to handle the frames in selenium we can handle the frame by using 3 ways 1st by using frame index and 2nd way we can use frame web element and 3rd way by using frame id or name, and if u want to handle frame compulsory 1st we have to switch our focus from main window to specific frame window with the help of switch to method and then frame method and we can not switch from child frame to child frame directly compulsory 1st we have to switch to parent frame then next child frame, if u want to switch parent frame we use parent frame method and if u want to switch to top frame or main window then we use default content method.

Then next we have created handle window class which is used to handle the window pop up or window tabs or window based alert pop. And I want to handle window we have to use to get

window handle method and if u want to handle multiple window then we use get window handles then return type of get window handles method is Set of String and once we capture all window then use iterator method to iterate value one by one and then we store this value in Array list because we access one by one window with the help of index positions.

Then next we have created handle actions class, by using Actions class we can perform mouse over and keyword board event, if want to work with Actions class so first we have to created object of Actions class by passing driver instance and then we can perform actions , and u want to combine the multiple actions then we use build method and then we use perform method to execute each and every action, and u want to click on element we use click method, u want double click on element then we use double click method and u want to mouse over on element then we use move to element method then if u want to right click on element then we use context click method then u want drag element from one location to another location then we use drag and drop down method and if u want to click on hold the element for some time then we use click on hold method, and u want to enter text then we use send keys method and if u want to perform key board event then we use key down method and key up method, keyDown method is used to press keys and key up method is used to release the pressed the key.

Next we have to created excel reader class to read the data from excel sheet, inside this class we used file class which is used to check file is present or not in desired location then we read the file by using file input stream object then we have to load the excel sheet then we use XSSFWorkbook object, once we load the whole workbook then we have to focus on specific sheet with the help of get sheet at method by passing sheet index to sheet name and once we focus on sheet then we use count how many rows in sheet with the help of get last row Num method and u want count how many columns in sheet then we use get last cell Num method and u want to capture sheet cell value the we use different method as like u wan to capture string value from sheet then we use get string cell value method and u want to capture int value then we use get raw value method and u want to capture boolean value then we use get boolean value method and once we capture value the we convert capture all values to object 2 dimensional array with the help of loops. And we return this object 2 dimensional arrays.

Next we have created listeners class to listener the test cases events in my current framework we have used to I Test Listeners once we implements this listeners it override On Start method on test start method then on test success method then on test failure method then on test skipped method then next on finish method, on start method will automatically when suite is started on test start

method this method call automatically when test cases is started and on test success method this method call automatically when test cases is pass and on test failure method this method call automatically when test cases is fail and on test skipped method call automatically when any test cases is skip and on finish method call automatically when suite is finish. And inside this listeners we have implemented extent reports, u want to store extent report in desired location we have to create object of Extent Spark Reporter class by passing report location and then we start generating report with the help of Extent Reports class once report is start then we attach report to desired location with the help of attachReporter method by passing report location we call this method inside the suite start, once test cases started then we use create test cases methods from the extent reports class and this will return extent test class object and once test cases is started then it will mark test case is pass or fail or skip.so are add these logs inside the extent report with the help of log method and inside the log method we use status class to add log status to extent reports, we have different status in extent report as like pass or fail or skip or info , we add these log inside the extent report and u want to attach screenshot to extent reports then we use addScreenShotFromPath method to it will add screenshot automatically to extent report once the suite is finish then stop generating report then we flush method.

Next we have created Retry analyzer and annotation transformer class to run fail test cases again and again , and from the I Retry Analyzer we override the retry method to run the test cases and from I Annotation transformer we override the transformer method and we use add retry analyzer class name.

Inside the source test layer package there we have created step definition package and inside the this package we stored all the generated snippets and inside this class we call associated functionality methods from Page Layer package..

Then next we have created 3 more different folders, 1st we have created Extent Report folder to maintain the all Extent reports then 2nd we have created Pass Screenshot folder inside this folder we maintain all pass screenshots then 3rd we have created Fail screenshot folder inside this folder we have maintained all fail screenshots.