TestNG

-- We can execute multiple test cases in TestNG whereas we can't in Java compiler

-- TestNG is defined by Annotations and followed by methods

Ex: @Test // Annotation (Need to import package using mouseover)

// Followed by method

public void Demo() // where public is access modifier,void is return type, Demo is the method name

{

system.out.println("hello");

}

Run as TestNG

-- XML file is very important where all the tests come in this XML file

-- XML is kind of a test suite which has a module and inside test cases within that module such as Personal loan, Car loan, Credit card loan, etc

-- A Suite can have multiple modules

<suite name="Loan Department">

<test name="Personal Loan">

<classes> // inside classes we can have multiple classes

<class name="test.day1"/> // classname= packagename.classname where packagename is 'test' and classname is 'day1'

<class name="test.day2"/>

</classes>

</test>

</suite>

-- Test Suite-->Test Folder(Module) --> Test Case

-- We can define multiple tests in a single class

-- We can modularize the testcases based on functionality and trigger them accordingly.

-- We can Prioritize test cases using TestNG

-- We can run only Carloan module test cases out of Personal Loan, Credit card loan

-- File --> Save All

-- testng.xml -- right click and Run as TestNG

-- Now we have only 5 to 10 test cases we can comment and run only that module

-- But tomorrow you have 100 to 500 test cases , so comment & removing those test cases is not possible

-- You can switch of and swith on in the framework with the test cases

-- Inside class you want to run only few methods and not entire (E.g. Weblogincar method, Mobilelogincar method, LoginAPIcarloan method)

-- Here in this scenario I want to run only one method -- how to do?

-- Include and exclude mechanism to control test cases

-- Exclude method Ex

<class name="test.day3">

<methods>

<exclude name="MobileLogincarloan"/>

</methods>

</class>

</classes>

-- This is how we can remove a particual method from that particular class

-- Same way we can use include in method also as same as exclude method

-- Out of 100 cases client required to run only 2 classes (here we can't write exclude to 98 test cases instead smartly we should use include method for the 2 test cases to include in the xml file.

-- Tomorrow if you have 30 test cases + 30 test cases , we can't write exclude and include methods

-- So we need to have naming convention is must

Ex : If we are working on Mobile module all the classes should start with Mobile\_classname (E.g. MobileLogin,MobileSignin,MobileSignout,etc)

-- So now if we got requirement to remove or add Mobile module

we can execute the test case as :

<class name="test.day3">

<methods>

<exclude name="Mobile.\*"/>

</methods>

-- Whenever you are creating another xml file- need to create at project level (E.g. testng2.xml)

<suite name="Loan Department">

<test name="Personal Loan">

<packages>

<package name= "test"/>

</packages>

-- We can run at package level

-- It is useful to run regression testing (Scenario we need to run Mobile,Appium,RestAPI,Selenium all together) real time

-- Understanding TestNG Annotations

@BeforeTest

public void prerequiste()

{

Syso ("I will execute first");

}

-- As we have total 10 test cases in day1,day2,day3,day4 and diff methods are present in diff classes

-- Before Annotation will be done on only 1 particular folder (E.g. Personal Loan and for only that particular block.)

-- Before writing @BeforeTest , i.e. at @Test Annotation my program will be executed at 3rd place but after using @BeforeTest Annotation it will be executed at 1st place in output.

@AfterTest Annotation

@BeforeSuite means - it belongs to all folders , such as Personalloan folder,Carloan folder

@AfterSuite

-- General used to set Global environment variables to your framework

-- In general we use if we have different URL's in QA env, for every release we get a new environment, so that env details we will set in @BeforeSuite , so that before each and every test case starts execution we will make sure that you will load all your env variables and trigger the test cases.

-- In real time we use @BeforeSuite we use to test different url in different environments

-- It's not a rule , it's upto you how to define as we are author for this framework.

-- If you have 4 tests(methods in class)BeforeMethods==

BF, 1 BF 2 BF3

@BeforeMethod

-- It is not mandatory to use all the Annotations

-- \*\*\*Famous interview question Difference between @BeforeTest and @BeforeMethod\*\*\*

Ans. @BeforeTest depends on TestNG xml file based on Test Folder scope

@BeforeMethod is specific to class file and executes before each and every (test) method

@AfterMethod - Will execute after all the Annotations are done

--- Execution of test cases will be done according to the Alphabetic order

=================================================================================================================================================================

Example: Write a script to Validate the heading on the page . Also need to validate the quotations of the headings

Hint using getText() & to validate using Assertions

public class WebDriverExcercise1{

Webdriver driver;

@Test

public void f(){

String Text= driver.findElement(By.xpath("/html[1]/body[1]/div[2]/div[1]/span[1]")).getText();

System.out.println("The Text is "+Text);

Assert.assertEquals(Text, "\"Surest way to learn Practical Agile Testing\"");

}

@BeforeTest

public void beforeTest(){

driver=new FirefoxDriver();

driver.get("http://www.agiletestingalliance.org");

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