Test Ng : Pre Defined frame work that Can be applied on Selenium

Test New guration : Java based testing tool.

@Test annotation describes the methods identifies them and help to execute the methods

Adv
test Cases and suits
Grouping of test Cases
Prioritize
Parameterize
Parerale testing
Report's

→ No main method in Test Ng.
→ method execute in alphabetical order. by default.

@priority : Specify which methods to be executed according to requirement.

@TEST(priority=1).

Testng.xml file used to create report and run multiple Test Ng Cases. which is Test-suite

```
<suite name="my suite">
    <test name="my test">
        <classes>
            <class name="  "/>
        </classes>
    </test>
</suite>
```

@Before Class
@ after class
@ Before method
@ after method
@ Before Test
@ after Test
@ Before suite
@ after suite

Test Ng : Pre Defined frame work that Can be applied on Selenium

Test New generation : Java based testing tool.

Adv
test Cases and suits
Grouping of test Cases
Prioritize
Parameterize
Parerral testing
Report's

@Test annotation describe the methods identifies them and help to execute the methods

→ No main method in Test Ng.
→ method execute in alphabetical order. by default.

Testng wont execute if @Test annotation is not added.

@priority : Specify which methods to be executed according to requirement.

@TEST(priority=1).

Testng xml file used to create report and run multiple Test Ng Cases. which is Test-suite

```
<suite name="my suite">
  <test name="my test">
    <classes>
      <class name="  "/>
    </classes>
  </test>
</suit>
```

@Before Class
@ after class

@ Before method
@ after method

@ Before Test
@ after Test

@ Before suite.
@ after suite

before

Before method:- execute the method ~~after~~ every @Test method annotated method.

after method :- execute the method after every @Test method annotated method.

(or)

login @BeforeMethod.
Search @Test
logout @AfterMethod.
login
adv search @Test
logout

Beforeclass : works at class level and execute only once before starting the test method.

Afterclass : executes only once after the ending of the test method.

login @Beforeclass
search @Test
~~login~~ search @Test
logout @Afterclass

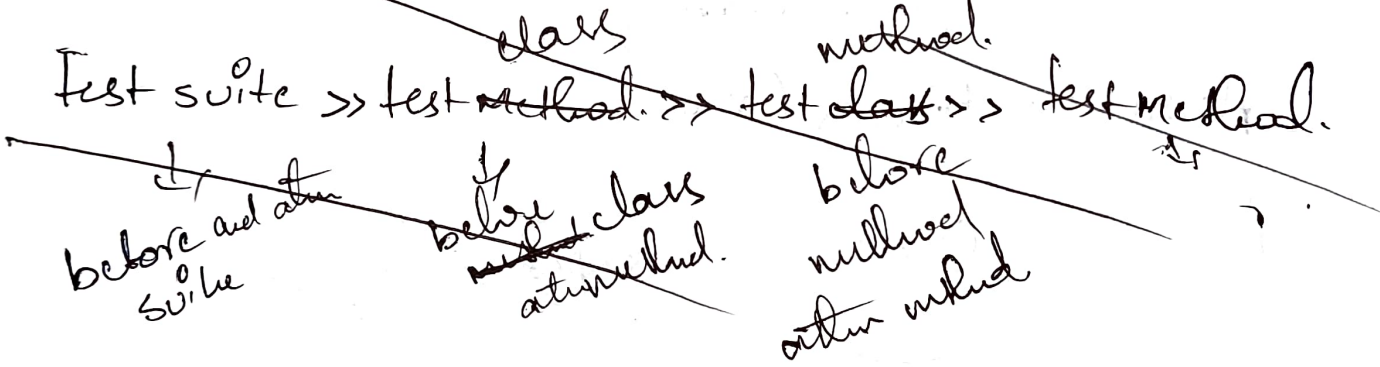
@BeforeTest :- Only one time before Test (group of classes)

steps to do before certain Test's can be done by @BeforeTest

@Before method exists once for every class.

Same goes for @AfterTest

hierarchy of execution



Test suite → Test cases → Test class → Test Method.

Dependent Methods to validate methods

Assert.assertTrue(True) validates the method is True to fail a specific Test Method.

Assert.assertTrue(false)

dependsOnMethods = {"method name"}

(optional)

annotation used when to check the certain method is passed and the annotated method will only execute if the dependson method is passed.

skip: Method not executed

failure: executed but method fails intermediately

Assertions (used for validations)

Assert.assertEquals(x, y);

Assert.assertEquals(actual, expected, description)

only executes this when failure occurs

int a=10
int b=20

(a>b, true, "a is not > than b");

hard and Soft Assertions

all the assertions called by Asserts method directly are hard Assertion

Assert.assertEqual();

if Assertion is ~~passed~~ failed in Soft Assertion. then the rest of the statement are also executed.

Hard assertion

if Assertion is ~~passed~~ failed. on hard Assertion then the rest of the methods (or) print statements wont be executed.

```
Assert.assertEquals(1,2);
S.O.println("hard assertion Completed...");
```

assert fails and print statement won't be executed

soft assert

soft assertion object creation is req	hard assertion direct access from static class
-> if failed next steps are executed	-> if failed next steps wont be executed.

-> create a obj of soft Assertion class

```
SoftAssert sa = new SoftAssert();
sa.assert.Equals(1,2); -> failed assertion
S.O.println("statement");
sa.assertAll(); -> mandatory at end of all softassert
```

Here even the assert fails the print statement is printed.

Grouping

grouping according to type of testings like Sanity, regression etc...

@Test (Priority=1, groups={"Sanity", "regression"})

this this method belongs to Sanity and regression testing.

after annotating groups specify to repeat (or) to do the same again and again should be manually specifying in xml file.

add

```
<groups>
  <run>
    <include name="Sanity" />
  </run>
</groups>
```

all the Sanity test cases are passed.

Parameterization :- @Dataprovider. xml
2 types

Passing Parameter to the test's.

Dataprovider provids data for another method with multiple sets of data for testing multiple times repation with no for loop (or) any loop.

returns 2 dimensional array as output

@DataProvider :- gets data from xl and pass them as multiple sets of data as to test

```
String[][] logindata()
{
    get dat from xl sheet.
}
```

@Dataprovider (name = dp)
↑ uppercase

call this dp name in @Test annotation to pass the values in dp method like

@Test (dataprovider = dp)
lowercase, uppercase

and all the data in dp is given to the @Test

to run only with spcific data from data Provider.

@DataProvider (name="dp", indices = {0, 2})

only 1st and 3rd data from the data set is sent for testing

Parcral testing

invocation count : with no Data provider to run a method certain number of times with no loopings and data provider.

```
@Test (invocationCount=10)
void test()
{
  sop ln ("testing ...");
}
```

used when to check a functionality that wont work/ crash after doing multiple test on same issue use this method to check

Parallel test

1. create Test classes.
2. create a xml file.
3. Pass a parameter browser name from xml file to specific method. to do this create a parameter tag in xml file

* ✓ <parameter name="browser" value="chrome" />

before classes tag and after Test tag

4. add @parameters ({"browser"}) to the specific method. then parameter needs to be passed
5. add (string br) inside the method brackets

ex

```
@parameters ("browser")
void setup (string br)
{ if br.equals("chrome")
  { driver= newchromedriver();
  }
 } else if (br.equals ("Edge")
   { driver=newedgebrowser();
   }.
```

holds the value of name browser which is chrome

To execute cases on multiple browsers ~~at a time~~

(serial browsing)

In xml make two test blocks with diff test name. and diff browser name in the <Parameter> tag

execute parrally

add parameter on Suite tag level

max-5

In <suite name="mysuite" threadCount="5"

Parallel="tests">

threadCount gives more memory location for increasing speed of Test execution and max limit of threadCount will be suggested as "5"

~~to create~~

Passing url from xml similar to going browser name
one more parameter tag
add @<parameter name="url" value="linkurl" />

add this to @parameter ({url})
void test(string url){

}

Test listeners: to perform the post actions (ex) if test method passed do some action this is called Post action.

Report generation is a Post action.

implement Interface

=> ITestListener: a interface of listner with abstract methods to implement post action

=> TestListenerAdapter: a class also used to implement post action.

extend class

Some methods are

=> onTestStart()
onTestFailure()
onTestSuccess()
=> onTestSkipped()

present in but in my ITestListner Plement it

any one

to invoke listeners class we need to add.

```
<listeners>
    <listener class-name = "day38.myListener" />
</listeners>
```

before test tag in the xml file

generate extent report

1. Add extent report dependency
2. create test cases
3. create listener class [new class] with class implements ITestListeners

add this in the xml

also add a utils file for re-usability like in for apache POI

4. create a xml file and execute in xml we this in listener and in class

~~name of the util file name~~

Json And xml

=> Javascript Object Notation extensible markup language

used to represent Complex data

API : Application Program Interface how two Softwares Communicate each other.

Json and xml

Structured data that doesn't fit in to a table

xml: based on html and older.

Json :- less Text and newly introduced.

Data types and structured data

Data types	Json has 3 types	xml has only one.
Int	Number	string
decimals	string	
Text	Boolean.	
True/false		
custom types.		

we can also have Collections

Arrays (lists)

Dictionaries (Lookup tables)

Array have

- -> list of data values
- -> has size
- -> has order.

dictionaries

- -> collection of key values.
- -> we use key to lookup the value.

structured data

Combines datatypes with Collections

- -> Dictionaries of list's
- -> list of dictionaries
- -> Dictionaries of dictionaries.

at the end the structured data is a very Complex Tree of data

JSON

javascript object notation. : to hold structured data to be used in javascript Programme

basic data types in JSON

- → strings.
- → numbers.
- → Booleans.
- → null
- → Arrays. [ ]
- → objects. [Json literals] { }
  ex { one : 1, two : 2 }
- → Nesting. [objects inside arrays
  arrays inside objects etc]

(=) white spaces doesn't matter unless inside the double quotes
(=) indent for every level of curlies.

Description and tables

Json documented on one (or) more tables. response table

element	Description	type	notes.
↓	↓	↓	↓
key/value Pair	describe. a release	datatype	Additional Info

Request table

element	Description	type	required	notes
			↓	
			fill with required / optional for developers to see	

xml : orgins in html used to load structured data.

Tags :- way to handle tag.

<_> - starting tag
</_> ending tag

Content :- goes between a tag

Nested tags : tags inside tags to create nested data.

ex :-
```
<color>
<red> 205 </red>
<green> 123 </green>
<blue> 52 </blue>
</color>
```

Attributes : key/value pairs which are strings with no spaces (or) Punctuation

⇒ key = "value"

ex

<file size unit = "KB"> 346 </file size>

↓
Attribute

Schema: structure of xml file also called as XSD file (xml schema Definition) Describes the tags, attributes, and types are of schemas are very helpful in documentation.

Documenting xml :-

xml has attributes where as JSON doesn't. If API uses both JSON and xml and then both of ~~them~~. xml and JSON doesn't have Attributes

→ when few Attributes : use Notes Column
→ many Attributes : Add an attribute Column
→ One table per element type. works well for xml
→ for request : add required Column.

Documenting structured data

→ System use structured data to create documentation
→ Advantage of being machine readable
→ work best for straight forward API's can't handle the complexities.

① an xml DOM is standard way to access and manipulate the message written inside xml documents

② info form xml DOM retrieved by an JavaScript methods.

③ Parsing text data from an xml file

④ create file obj for xml file (available)

File fxmlFile = new File (Projectpath + File.separator + filename)

⑤ create document builder.

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

⑥ create doc builder obj

DocumentBuilder dBuilder = dbf.newDocumentBuilder();

⑦ create a Doc from a file. Creates an instance of DOM architecture of xml file

Document doc = dBuildg.Parse(new file("file_name"));
doc.getDocumentElement().normalize

step2

-> create xpath obj and xpath exp

Xpath xpath = XpathFactory.newInstance().
newXpath();

-> use xpath.Compile() and get a list of nodes and evaluate via xpath.evaluate()

String exp = "/class/student";
NodeList list = (NodeList) xpath.Compile(exp)
.evaluate(doc,
xpath.Constats.NODE
SET);

step3:- Parse value in web portal

To get value of ID and attribute

(ex) stringID = node.getAttributes().getNamedItem
(ID).
getNodeValue();

Read data from json file

```
JSONParser json = new JSONParser();
FileReader file = new FileReader("Testdata.json");
```

Uread json file

```
Object obj = jsonParser.parse(reader);
```

To pare json which starts with Array braketes

```
JSONArray user = (JSONArray) obj
s.o.p ln (user);
```

Page objut model: manage the pattern of page element's and maintain page objut's.

in Page objut model contains the Page objut classes for every Page to reduce the duplication.

(ex)

```
Public class loginPage {
    WebDriver driver;

// Constructer.
LoginPage (WebDriver driver)
{
    this.driver = driver
}
// locators
    By logo_img_loc = By.xpath (" ");
    By txt_Name_loc = By.xpath (" ");
    By txt_Pass_loc = By.xpath (" ");
// Actions
Public void setUsername (String username)
{
    driver.findElement (txt_name_loc).SendKeys (User name)
}
```

Similar methods for password and submit button

Validation for the test done in new class

```
Public class login-test {
        WebDriver driver;
    @Test
    Void testlogin()
    {
      loginpage lp=new loginpage
      lp.
      Assertassertequals(lp.checkloginpage(),
                                        true);
    }.

    @Beforeclass          @Afterclass
    Void setup            Void close()
    {                     {
    setup driver.           close driver
    }                     }.
```

loc and attributes → Page obj

Assert functions/validation → Test class.

Page factory

```
// Constructor
loginPage (webdriver driver)
{
    this.driver = driver;
    PageFactory.initElements(driver, this);
}
// webelement locator identification
@FindBy (xpath="") WebElement img_logo;
@FindBy (xpath="") webElement txt_name_loc;
@FindBy (xpath="") webElement txt_Pass_loc;
@FindBy (xpath="") webElement btn_submit;

// Action methods
Public void setUsername (String username)
{
    txt_name_loc.sendKeys(Username)
}
```

driver is given access to the whole code through the PageFactory

Similar for other methods → and in class loginTest create obj and call methods

Automation Framework

Built-in framework: ex. testng, junit, Cucumber which gives the additional features.

Customized (user defined); ex modular framework, data-driven

depends on the type of Projects and requirements that changes.

Objective of Automation framework

(i) re-usability
(ii) maintainability
(iii) readability

(2) implementing, POM, hybrid driven framework, testing in using the framework

(⇒)* Using test cases for automation

(i) sanity (P1)
(ii) data driven test cases / re-tests (P2)
(iii) Regression test cases (P2)

(⇒)* Design and develop framework.

(*) execution in local system.

local : Test cases pass locally then go to remote execution

remote : jenkins

(*) maintenance

maintain project as in a server so others can access like github repository etc..

logging :- records all events from the test.

log levels - All < Trace < Debug < INFO < Warn < Error < fatal < off

level = Trace then all the logs including trace are selected (logged)

mostly info and Debug are used.

logs file has Appenders, loggers.

Appenders : where to generate file (Console/File)

loggers : what type of log to generate like trace, debug, info etc

Data driven testing in automation framework

Prepare test data — (1) in excel file

name	Password	res
-	-	valid/invalid

create a dataprovider to read data from excel and read in 2 dimensional array.

keep the excel data in test data folder

keep the data provider in utilities

this method used only when data needs to be taken from excel sheet and given to automation.

grouping test's in test case (class)

@Test (groups = {"regression", "master"})

Applicable for all groups

↳ all tests belong to both master excludes all the test cases

@Beforeclass (groups = {"Sanity", "regression", "master"})

@Afterclass (groups = {"Sanity", "regression", "master"})

) belong to base test class and Before and after class need to be added to the groups because they need to be executed everytime regardless of which group that is need to be executed by application.

add the groups in the xml

```
<groups>
<run>
    <include name="Sanity" />
    <include name="regression" />
</run>
</groups>
```

Runs all Sanity and regression groups

before class tag

1. How to run only failed test cases on 2nd run

in testng-failed.xml from test output folder Contains the file which has a xml data of only failed test cases

Selenium grid : to run the test cases remotely in different OS and browsers

① Standalone method

where the hub and the node in the same system.

② Hub and node set-up

more practical setup works in realtime grid execution with more remote systems Control in the network

→ download. Selenium-Server-4.15.0.jar

and run. java -jar selenium-Server-4.15.0.jar Standalone

Using Stand-alone:

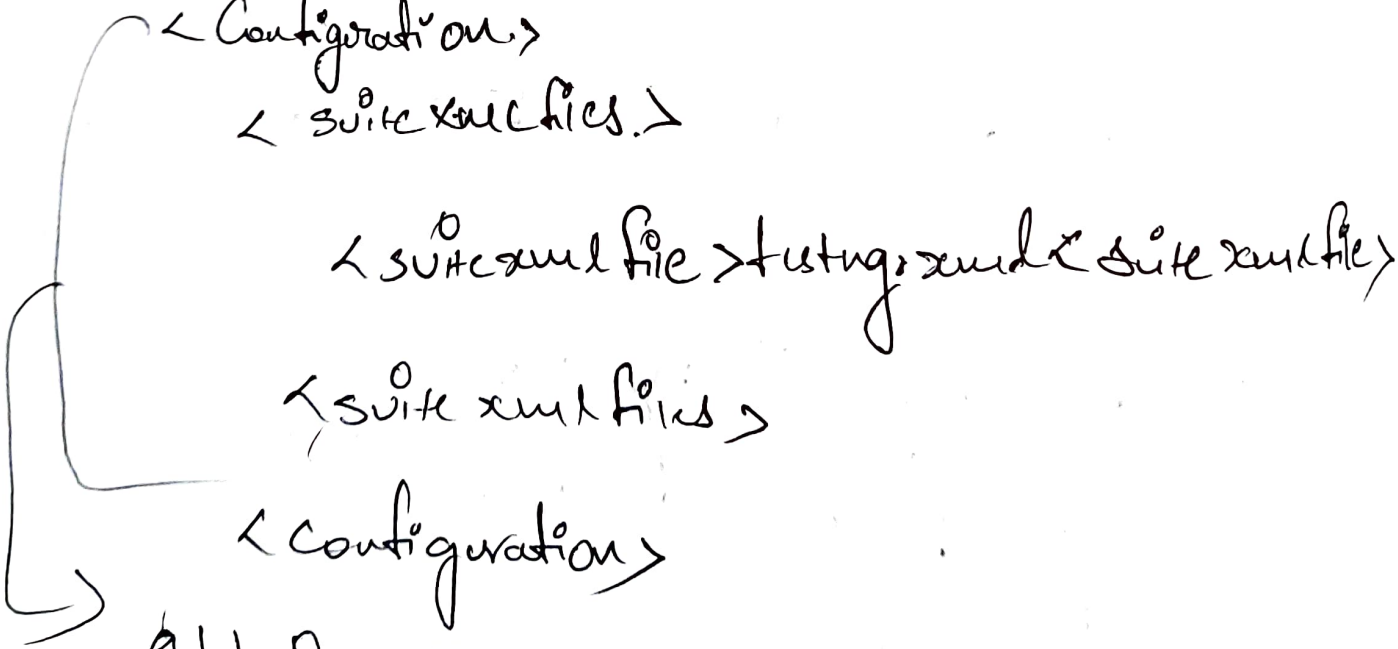
```
Public class Seleniumgrid {

        P.S.V.M (String.args[]) throws exception.
{
  String nodeurl = "http://localhost:44444/wd/hub";
  DesiredCapabilities cap = new desiredCapabilities();

  Cap.setPlatform (Platform.MAC)
  Cap.setBrowserName ("chrome");

  WebDriver driver = new RemoteDriver (new URL(nodeurl)
                                       , cap);

     driver.get ("https://www.google.com");
     S.O.P.ln (driver.getTitle());
     driver.quit();
  }
}
```

1. executing the application

(i) using xml -> normal way

(ii) using Pom.xml
by
running the app through running Pom.xml file by adding Plugins (used to Compile & run application)
(i) maven-Compiler plugin
(ii) maven-Surefire plugin

and in Sure fire plugin add Configuration like



add P.M. Pom.xml, inside Sure fire Plugin.

To run application on Cmd prompt

download and install maven on System level

In Command prompt specify location of project

cd location

then use mvn test to run application.

Version-Config systems

- git - local repository (storage area to maintain diff files) → Single access
- github - Remote repository → group access

The flow of Code