

## Selenium - Naveen Training

- webdriver - Launch browser
- Locators, Utilities
- Total links, findelements
- Dropdown, Jquery Dropdown
- Alert
- frame
- Webtable
- wait
- Javascript Executor
- StaleElement

31st dec

use Remote webdriver

Selenium grid  $\Rightarrow$  Remote mlc  
Run test cases on Remote mlc

Stable mode is 3x.

4x  $\Rightarrow$  beta

Adv  $\Rightarrow$  ① open source / free licence  
(Selenium)

i) source code is available

② supports multiple languages  $\Rightarrow$

only syntax will diff.

③ supports diff platforms  $\Rightarrow$  window, Linux, MAC OS

Can't execute test case on unix os

④ multiple browsers  $\Rightarrow$  ch, ff, opera, edge, ie, safari

⑤ W3C  $\Rightarrow$  rules are defined that we need to follow

⑥ easy setup, easy integration  $\Rightarrow$  maven, testing, junit, jenkins, ciit

drawbacks of selenium  $\Rightarrow$

① No support for desktop application

② No In-built report

③ Selenium is Browser Automation Library

④ just Automation API library

⑤ performance testing  $\times$  (not recommended)

⑥ security testing  $\times$

⑦ Exploratory testing / monkey testing  $\Rightarrow$  keep going doing testing on the basis of your experience  $\times$

- ⑦ API / DB testing  $\Rightarrow$  Post Assured, HTTP client  
only browser can test
- ⑧ No object Repository
- ⑨ No mobile testing  $\Rightarrow$  Appium
- ⑩ mobile browser  $\Rightarrow$  ✓ + but limited
- ⑪ HW  $\Rightarrow$  barcode, captchas can't do with selenium  
Can automate browser activity

### How to setup selenium

working directory  $\Rightarrow$  eclipse, folder, file where your code is there.

create maven project

Configure maven with sel.

pom.xml

sel  
Jdk 1.8 + jars  
can write selenium code

maven repos

new  $\rightarrow$  project  $\rightarrow$  maven  $\rightarrow$  filter  $\Rightarrow$  maven-archetype-quickstart

filter  $\Rightarrow$  maven-archetype-quickstart (1.4) Select it

next  $\rightarrow$  Group Id  $\Rightarrow$  ArtifactId  $\Rightarrow$  group Id  
(project name)

maven project will give you pom.xml which is

pom.xml

Selenium & maven dependency  $\rightarrow$  search in google  
copy dependency & paste in pom.xml

4.1.1

Ctrl + shift + f  $\Rightarrow$  Aligning property

Ctrl + S  $\Rightarrow$  download Selenium Java in your mvc

All dependencies will be stored in maven dependencies

check dependencies in maven

for update according to maven dependency  $\rightarrow$   
right click on project name  $\rightarrow$  maven  $\rightarrow$  update project  $\Rightarrow$  OK

Main java  $\Rightarrow$  new  $\rightarrow$  package  $\rightarrow$  created

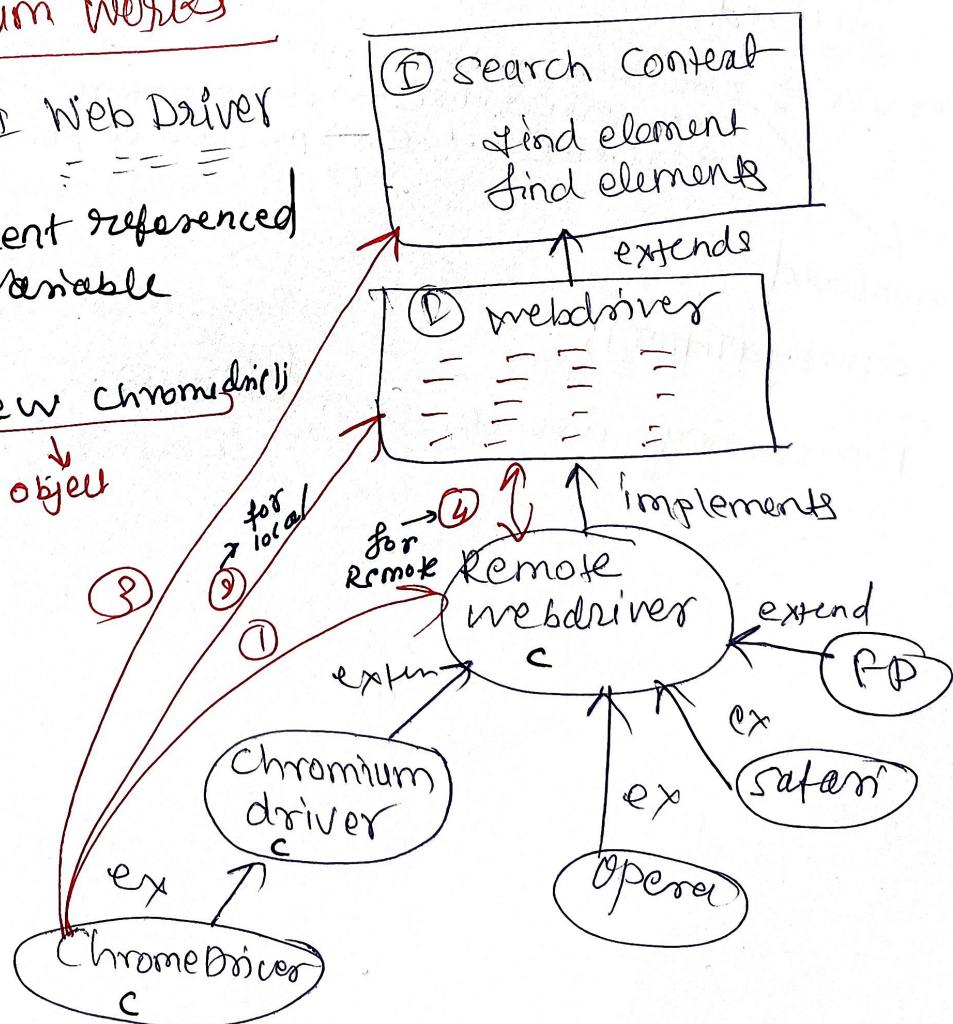
### How Selenium Works

#### I Web Driver

Top casting  $\Rightarrow$  parent referenced variable

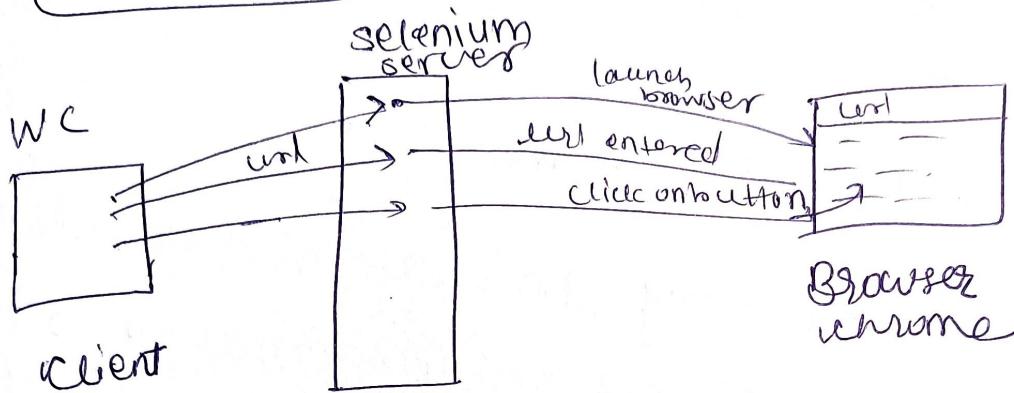
# Top casting

Webdriver driver = new Chromedriver();  
refered by this



## How selenium works

Working copy



algo will provide power  $\Rightarrow$  browser responsibility  
according to vendor/vendor & server will be  
decided

server  $\Rightarrow$  chromedriver.exe  $\rightarrow$  google  
geckodriver.exe  $\rightarrow$  by ~~mozilla~~  
edgedriver.exe  $\rightarrow$  provided by microsoft

download

driver.getTitle();

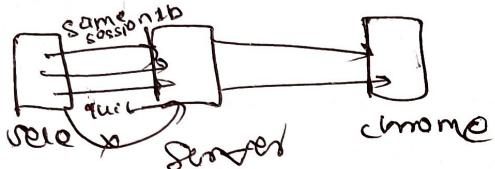
Top casting use in cross browser.

## Quit vs close

If you are writing any Selenium code after

`driver.quit()` → it will throw ~~an~~ exception  
like `NoSuchSessionException`. (session is null)  
→ close the browser [server will keep running].

right click → run as debugger → mouse over on code



`ff` → browser close current browser open  
for every test there will be new session ID  
close browser with `quit`, it will always reinitialize  
the browser.

`driver.close()` # invalid session ID

same session ID will be there.

license of session ID is expired

In both cases, we need to reinitializes the browser.

`driver = new ChromeDriver();`

Check in task manager →

`System.exit()` # shutdown system

diff bet `get()` & `navigate().to()` → both are same.

both are waiting to load the page.

Basic diff →

`navigate().to`

04/01/22

quit() → server is not closed

→ when we launch browser then only session id will create.

press f8 for debugger.

→ if we write code after close() → id is there so, cancelled/expire session id displayed

→ if we write code after quit() → null id is there  
means, browser doesn't have session id. If you still want to open chrome driver still if you want to open chrome driver

have to launch a browser.

System.exit() → shut down Java

→ navigate.to() & get() are same.

only difference is syntax wise

→ to() method calling get(). Internally

ctrl + shift + f

driver.getTitle() # get the title of webpage.

driver.getCurrentUrl()

driver.getPageSource()

default value of driver is null (because it is non primitive datatype)

Safari can work on Mac.

private Driver; # other user cannot see it directly

accessing private variable through public methods called as Encapsulation

What if I want to execute multiple browser at a time?

We can't do with the help of main Java. We need to use TestNG.

### problem Statement

chrome - 96.0 → .exe  
ff → 95.0 → .exe  
safari → .

what will do if you want to execute on 95.0 or 94.0

comes in picture.

### so WebDriver Manager

↳ add this library in pom.xml

- It will first check what compatible browser do you have
- whatever the current version of your browser, WDM will check in chromium site & then according to that update in Script.

• check current browser version

• then according to that WDM will update script.

SWM it is not part of selenium, it is external utility.

go to official website of webdriver manager on github.

(configuration)

→ WDM works only on Java.

How to use ?

(ctrl+shift+f)

Just add dependency on pom.xml

<scope>test</scope> → it will only use in test folder (this proper folder running test).

↳ if remove this line from dependency, you can use in src/main/java or src/test/java

for latest version

① Check which browser version using

② Then check that

version is in system or not

③ If not found then download that

↳ What if version < 97.0 or before 97.0 Version ?

↳ You can't downgrade browser version you have to do it

manually - WebDriverManager.chromedriver().browserVersion("87.0").setup();

① Check compatibility of your system

② Then setup server accordingly

↳ If you don't find version  
or less than 97.0  
download older browser  
and set it

## Locators

① create a web element by using By locator  
create web element + action (click, sendkeys, isDisplayed)

searchContext overridden by webDriver &<sup>10</sup>  
findElement findElements

By is class in selenium.

findElement says give me By

id is method

static  
① created web elements then performed actions.

Xpath ⇒ It is not attribute. It is address of  
webElement.

① Absolute → ② xpath

linktext ⇒ used for mostly hyperlink text

partialLink ⇒ if there is lengthy link then use partial part  
every element having 2 imp things → own html tag  
attribute and attribute tag

① create a webelement

Web element = By locator [Create webelement using By locator]  
Create web element & then perform action on it  
can validate it.

06/01/22

## Locators

input → html tag  
attribute = attribute value

Locators ⇒

① ID ⇒ // create a web element by using By locator  
then perform some action on it.  
action could be click, sendkeys, check element  
displayed or not.

see ⇒ By is class in selenium

\* ~~create web element~~ welement is created with "By" locator

① You have to perform some action on it.

create web element + action (click, sendkeys, isDisplayed)

driver.findElement(By. locator name)

↓  
class name

② ~~① create web element~~

driver.findElement(By.id("input-email")).sendKeys("...");

in this approach everytime we need to write above  
line for creating web element.

problem? ⇒ what if we have 10 or 100 web elements  
then you write above line for 100 times?

⑤ approach  $\Rightarrow$  created webelement and store in object.

with the help of that object we can send or perform action.

```
webElement emailID = driver.findElement(By.id("input_email"));
emailID.sendKeys("nav@gmail.com");
```

- can use diff parameters

\* But only one time created webelement

3rd approach  $\Rightarrow$  By locator strategy

```
By emailID = By.id("input_email"); # never hit server
WebElement emailID = driver.findElement(emailID);
getelement(emailID).sendKeys("nav@gmail.com");
```

$\Rightarrow$  we are not hitting server

4th approach  $\Rightarrow$  By locator with generic method.

```
By emailID = By.id("input_email");
getelement(emailID).sendKeys("nav@123");
```

```
public static WebElement getElement(By locator) {
    return driver.findElement(locator);
```

}

5th Approach  $\Rightarrow$

~~By locator~~ 5. by locator with getelement and actions  
generic method.

By emailId = By.id("input-email");

By password = By.id("input-password");

doSendKeys("emailId", "nav@gmail.com");

doSendKeys(password, "nav@123");

public static WebElement getElement(By locator) {  
 return driver.findElement(locator); } *# find element*

public static void doSendKeys(By locator, String value){  
 getElement(locator).sendKeys(value); }

*use in every* 6. by locator with getelement and actions  
generic method in util class

By emailID = By.id("input-email");

By password = By.id("input-password");

ElementUtil.eleUtil = new ElementUtil(driver);

eleUtil.doSendKeys("email", "a@gmail.com");

eleUtil.doSendKeys(password, "test@123");

7. String locator with By, getelement & actions generic  
method in util class

String id = "input-email";

String pwd = "input-password";

public static void ~~getBy~~ <sup>By locator</sup> getBy(String locatorType, String locatorValue){  
 switch(locatorType){  
 case "id":  
 ~~get~~ By.id(locatorValue);

```
string eid = "input-email";
string pwd = "input-password";
doSendKeys("id", eid, "nav@gmail.com");
doSendKeys("id", pwd, "nav123@gmail.com");
→
public static By getBy(String locatorType, String locatorValue)
By locator = null;
switch (locatorType.toLowerCase()) {
    case "id":
        locator = By.id(locatorValue);
        break;
    default:
        break;
}
return locator;
}
public static WebElement getElement(By locator) {
    return driver.findElement(locator);
}
public static void doSendKeys(By locator, String value) {
    getElement(locator).sendKeys(value);
}
public static void doSendKeys(String locatorType,
String locatorValue, String value) {
    getElement(getBy(locatorType, locatorValue)).sendKeys(value);
}
```

~~Test Control Line~~

## ElementUtil.java

```
public class ElementUtil {  
    private WebDriver driver;  
    public ElementUtil (WebDriver driver)  
    {  
        this.driver = driver; # for initialization  
    }  
    public WebElement getElement(By locator){  
        return driver.findElement(locator);  
    }  
    public void closeSendKeys(By locator, String value){  
        getElement(locator).sendKeys(value);  
    }  
}
```

~~if static remove this~~ then ~~it~~ we can achieve multithreading

if static is there then we can't achieve multithreading

or parallel execution

~~static~~ one copy create for all object in case

~~static~~ of static

only one test case can execute  
because it does not create copy

so remove static

we cannot achieve multithreading



## Total links Images GetAttribute findElements

① NoSuchElementException → can solve by synchronized, wait

② Total links available on page

→ total link.

→ print the text of each link

→ print the href value of each link

findElements → return more than one list  
↳ ArrayList → index base (order base)



① Total No. of links

List<WebElement> linklist = driver.findElements(By.tagName("a"));

(Object) → WebElements with values

② print the text of each link

```
int c=0;  
for(WebElement e : linklist){  
    String text = e.getText();  
    System.out.println(text);  
}
```

③ Remove blank text of links

```
If (!text.isEmpty()) {  
    if (text.startsWith(" ")) {  
        System.out.println(text.substring(1));  
    } else {  
        System.out.println(text);  
    }  
}
```

④ for href value

```
String href = e.getAttribute("href");  
System.out.println(href);
```

⑤ point index also

```
int c=0;
```

## \* Retrieve total No. of Images

### ① total Images

```
= driver.findElements(By.tagName("img"));
```

List<WebElement> imglist

```
S.O.P("Total img: " + imglist.size());
```

# print total img

```
for (WebElement e : imglist) {
```

```
String alt = e.getAttribute('alt');
```

```
String src = e.getAttribute("src");
```

```
S.O.P(alt + ":" + src);
```

# Retrieve <sup>more</sup> text with index No.

# Retrive href value from each

② using for each loop → print href value from each

```
c = 0;
```

```
for (WebElement e : linkList) {
```

```
String text = e.getText();
```

```
if (!text.isEmpty()) {
```

```
S.O.P("c :" + text);
```

```
}
```

```
String hrefVal = e.getAttribute("href");
```

```
S.O.P(hrefVal);
```

```
c++
```

```
I
```

By using for loop

```
for (int i=0; i < linkList.size(); i++)
```

```
String text = linkList.get(i).getText();
```

```
if (!text.isEmpty()) {
```

```
S.O.P("i :" + text);
```

```
}
```

## Total links generic method

```
public List<WebElement> getElement(By locator) {  
    return driver.findElements(locator); # return list of web elements  
} # return list of all web elements
```

```
public int getElementCount(By locator) {  
    return getElements(locator).size();  
} # return count of web element
```

```
public void printElementsList(By locator) {
```

```
List<WebElement> getElements(locator);
```

```
List<WebElement> elist = getElements(locator);
```

```
for (WebElement e : elist) {
```

```
String txt = e.getText();
```

```
s. o. p(text); # point the text
```

```
# print string (this method return list of elements text (String))
```

```
public static List<String> getElementTextList(By locator)
```

```
List<WebElement> elist = getElements(locator);
```

```
List<String> eleTextList = new ArrayList<String>();
```

```
String text = e.getText();
```

```
eleTextList.add(text);
```

```
} return eleTextList;
```

```
By img = By.tagName("img");
```

```
By links = By.tagName("a");
```

```
s. o. p(links); # supply any html tag you will get
```

```
s. o. p(links.size()); # the count
```

```
if (getElementsTextList(links).contains(img));
```

```
s. o. p("Registration is present");
```

Return All ~~with~~ Alt, & Src, href with locator.

S.O.P(getElementsAttributeList("images", "src"));

S.O.P(getElementsAttributeList("images", "alt"));

S.O.P(getElementsAttributeList("links", "href"));

# This method will return the list of elements  
attribute value

```
public static List<String> getElementsAttributeList(By locator,  
String attrName){  
    List<WebElements> elelist = getElementsList(locator);  
    List<String> eleAttrList = new ArrayList<String>();  
    for (WebElement e : elelist) {  
        String attrVal = e.getAttribute(attrName);  
        eleAttrList.add(attrVal);  
    }  
    return eleAttrList;  
}
```

3) Get Shampoo to all footer links

# Language Links (Capture links from specific area)

```
public static void main(String[] args){  
    WebDriverManager.chromedriver().setup();  
    driver = new ChromeDriver();  
    driver.get("https://www.google.co.in/");  
    List<WebElement> langlist = driver.findElements(By.xpath("//div[@id='  
S.O.P(langlist.size());  
for (WebElement e : langlist) {  
    String text = e.getText();  
    S.O.P(text);  
    if (text.contains("English")) {  
        e.click();  
        break;  
    }  
}
```

direct child  
indirect child  
use double slash

## 10/01/22 $\Rightarrow$ Locators

- attribute {
  - ① id :- unique
  - ② name:- can be duplicate
  - ③ class name:- can be duplicate
    - first level: Top to Bottom
    - left to right
  - ④ x-paths:- it is not attribute
    - it is address of element where in HTML DOM
  - too types of x-path  $\Rightarrow$  i) Absolute ii) relative
  - ⑤ css selector  $\Rightarrow$  its not an attribute
  - ⑥ linkText :- only for links
  - ⑦ partialLinkText:- only for links {when more than one}
  - find first matching element.
  - ⑧ tagName  $\Rightarrow$  html tag (always use on html tag) headers, link  
only for tag
  - not work on attribute property

public static String getText(By locator)

return getElement(locator).getText();

getText  $\rightarrow$  capture text of element

getAttribute  $\rightarrow$  return placeholder value

① How to capture text from paragraph?

find webElement + getText();

② How to capture text from textfield?

b.y.getAttribute(pass the value);

```
public void dogetAttribute(By locator, String attributeName){  
    return getElement(locator).getAttribute(attributeName)}
```

Validate text clickable on page or not?

isDisplayed() → ① return boolean

if (driver.findElement(By.xpath("//input[@type='text']")).isDisplayed()) {

driver.findElement(By.ID).sendKeys("Test @g-1");

public boolean isDisplayed(By locator):

return getelement(locator).isDisplayed();

isEnabled() → enable or not.

visible - but disable - false.

Task → demo.opencart.com & use all the utilities  
fill form by using locators

Method  
browserUtil → to launch browser  
ElementUtil → to interact with this element  
validate text success msg  
capture getText then validate  
② orangehrm.com - can't automate due to captcha

12/01/22 ⇒

→ dropdown

HTML tag should be select, so we need to give one argument  
to ~~the~~ select (dropdown)

① selectByIndex → select value in dropdown by using index

~~preferable~~ ② selectByVisibleText → it is preferable, because it is dynamic  
can be change index so use this method.

③ selectByValue → value attribute can be there or not

④ getOptions → return list of webelements

→ Not allowed to use select class to dropdown  
# Handle dropdown without using select.

By using Xpath

⇒ Google Search (class) ⇒ search anything then select  
any specific value from suggestion.

17/01/22

File upload pop up

- Selenium can not automate windows based applictn.
- without click we can't upload.

# html tag :- ~~if~~ type = "file"  
if not there then use to developer to add  
this property through fire

Authentication pop up along with username & password

AutoIT ⇒ only works on windows

- page is not loaded

# CDP tool - Sel4 (only for authentication)  
Auth 1.0 } ⇒ Are the backend authentication  
Auth 1.2 }

21/01/22

Q.W que  $\Rightarrow$  How will you select right/middle side checkbox  
in table?

IoW SV UIB Element  $\Rightarrow$

mess Cntr+S  
Cntr+F

& shadow dom element  $\Rightarrow$

for shadow root, we need to use "copy JS path"

(copy from code)  $\Rightarrow$  then paste in console

JavaScript Executor

Browser  $\rightarrow$  page  $\rightarrow$  iframe  $\rightarrow$  shadow dom  $\rightarrow$  ifp  
field element

use JS executor in Selenium by using  $\rightarrow$

```
( JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript (" query selector" ));
```

Is there any shadowDOM element in web

application? If yes then say No to that company

$\Rightarrow$  SVG element does not support CSS selector  
it only supports expert X-path.

Backend Support  $\Rightarrow$  XPath

Syntax rule  $\Rightarrow$  CSS selector good

26/01/22

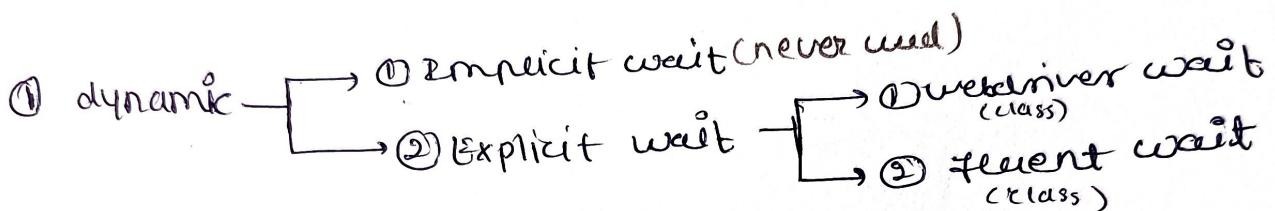
... → 300ms (0.5sec)

24/01/22 ⇒ Implicit Wait

Wait ⇒

- ① static : sleep (5000) # wait for the give time
- ② dynamic: timeout = 10. ... else 3 sec found ... rest  
of time will be ignored

else found  
3 sec still  
it will wait  
for 10sec  
rest



webdriverWait() → extends → fluentWait() → implements wait(I) →  
 "until()",  
 ↳ having no method body

Implicit wait ⇒

- ① It is dynamic wait
- ② It is global wait
- ③ only & only applicable for webelements  
FindElement & FindElements
- ④ Not applicable for non webelements.  
(title, url, alerts)

Global wait ⇒ once it is applied

31/01/22

TestNG - need to write project, so we need plugin

help - new s(c) → generated (eclipse ui) →

→ Unit testing framework - used by <sup>java</sup> dev | automation QA

→ only applicable in Java

→ every lang using it TestNG is available

python → PyTest, Unittest

PHP → phpunit

JS → mocha, Jasmine

C# → Nunit

- It is used to write Test cases with various features are available
- can define, priority, data providers → test data
- can create html report, xml report also provide listeners → extent/allure report
- Annotations - pre condition/post steps/tes
- Assertions - hard /soft
- test runner → testing.xml
- src/main/java → whenever you have any library, utilities or page classes written in main/java
- src/test/java → written only test cases → <sup>test</sup> code
- No main method in TestNG its own mechanism to run test cases.

// Global pre-condition

// precondition

// test steps + assertions (actual vs expected result)

// post condn/steps

@ BeforeSuite (# we need port lib for use this)

# every annotation associated with method

```
public void createUser() {  
    S.O.P("BS-create user");
```

@ BeforeTest

@ BeforeTest

```
public void connectWithDB() {  
    S.O.P("BT-connect with DB");
```

}

@ BeforeClass

```
public void launchBrowser() {  
    S.O.P("BC-launch Browser");
```

@ BeforeMethod

```
public void login() {  
    S.O.P("BM-login to app");
```

01/02/22

- \* Test should be independent--- not to be dependent
- \* try to avoid priority
- \* CRUD
- \* invocation count  $\Rightarrow$  calling count
- \* DataProvider  $\Rightarrow$  it will supply data  
we can define n no. of rows & colm
- \* we can run testNG from Runner class
- \* TDD  $\Rightarrow$  <sup>development</sup> test driven / test centric approach
  - unit test case point of view
  - test cases against requirement  $\rightarrow$  failed  $\rightarrow$  refactor code  $\rightarrow$  passed  $\rightarrow$  test cases  $\rightarrow$  failed  $\rightarrow$  refactor
- Think  $\rightarrow$  Test  $\rightarrow$  code/execute  $\rightarrow$  Refactor

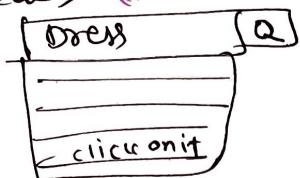
(Ass) footerworks.com =

capture all footerlinks

Google.com =>

+ center all footerlink with print, count & then print  
each on everytext turn click on that  
(position is not fixed there is no method based on position)

usecase



- ⇒ ① search it  
② No. of suggestions are coming  
③ basis on that click on it.  
collect suggestion in a list  
iterate it  
match the respected suggestion  
click on it

enter a value

get the list

capture list of element (driver.findElement)

match respective value

click on it

break loop

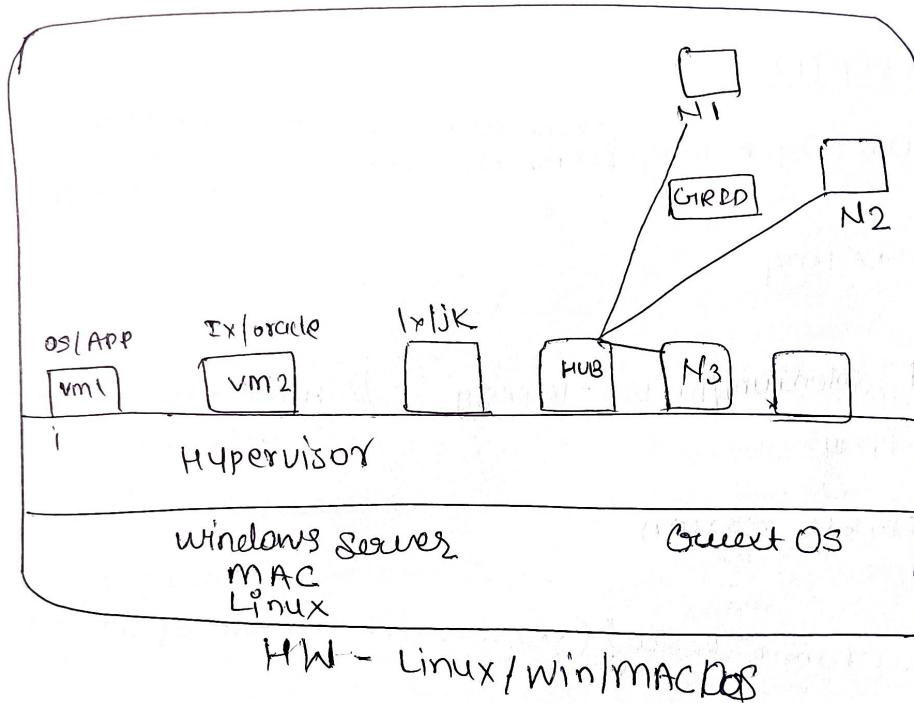
Calling utilities from testing methods  
design framework, writing test case, and generating the  
reports.

## Selenium Grid

Dis →

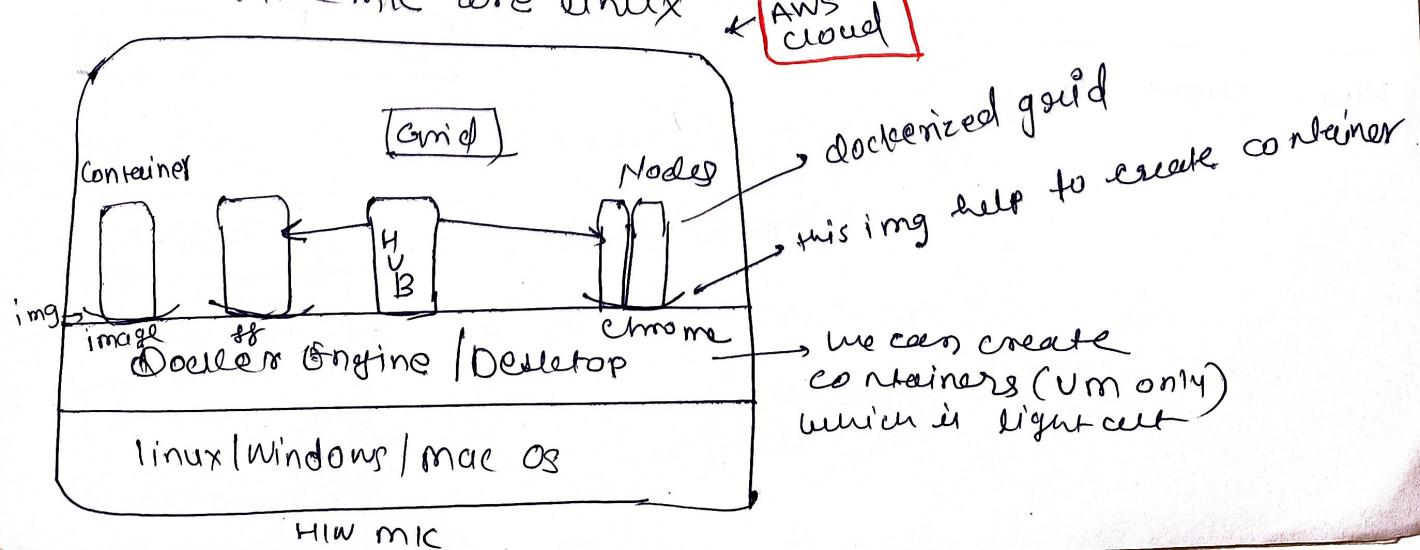
- ① Budget is high (cost of hw is very costly)
- ② maintenance is high

Docker and VMs comes in picture



hypervisor → provided by

- you can take this mlc on AWS
- most of the mlc are linux





docker hub

→ download  
those img  
then we can  
create n no. of container

hub.docker.com →

How to install Docker ⇒ Docker desktop

### Practical ⇒ Step ①

① Create Hub ⇒ get img from browser → search for selenium/hub

② will get linux img

⇒ docker pull selenium/hub:3.141.59 # write on cmd prompt  
# create img

download stable version

⇒ selenium/node.chrome.debug (search on website of docker)

pull img one by one

⇒ selenium/node-edge → for chromium browser

① Downloading img

② Create containers

— — —  
Hub    chrome    ff

## ② Step ② => create container

- report will generate in local/in mlc
- Dockerfile for running
- code is available in `gitlab`, `Jenkins`, local mlc
- Hub is selenium server
- browser is available in container, it will behave like servers

`docker images` # for show images

`docker ps -a` # for see container

⇒ `docker run -d` in detached mode => -d  
→ -p → port number

`docker run -d -p 4444:4444 --name selenium-hub -P`  
SeleniumHub:3.141.59

first 6 digit ⇒ container ID  
↓  
`detached mode`      `link with hub also`  
`link with hub also`      `name of hub`      `image name`  
`docker run -d --link selenium-hub:hub -P selenium/node-chrome-debug:3.141.59`  
↓  
tag name

`localhost:4444/grid/console`

`docker restart` # or restart container

Hub container should be up & running

`docker rm id` # remove container

⇒ but can not remove running container

`docker rmi id` # remove container images

`docker ps -a`

## DevOps Activity

(5)

How will you Run it?

- ① Run through Jenkins (Run in local m/c)
- ② Run through Maven

You need proper infrastructure for given test cases.

Do we need to do Environment setup

- i) push your code in git repository.  
entire code from local to remote side.

Git  $\Rightarrow$  Source code management (just for storage)

- ii) then we need to setup some triggering management. that is done through Jenkins management. to maintain job pipeline, CI/CD such things.

We have to create Jenkins Server.  
with Jenkins we can do Blue Ocean Pipeline, classic, basic pipeline, CI/CD pipeline.

can create single project - can parameterize or scheduling a build, environment configuration, allure reporting | Extent Report. Jenkins will take care of it.

Jenkins pull the test cases, pick the code from git (master branch) to Jenkins m/c.  
one workspace will create entire code will be stored in workspace.

Never run the test cases on Jenkins machine because if someone using browser in Jenkins then another team member can not use it.

Jenkins is just used to trigger the test cases.

Not launching browser on git nor Jenkins. It should be launch on 3rd party or on another machine.

- ① choose one HW or machine then run test cases
- ②

⇒ Every & possible

⇒ So, we can run

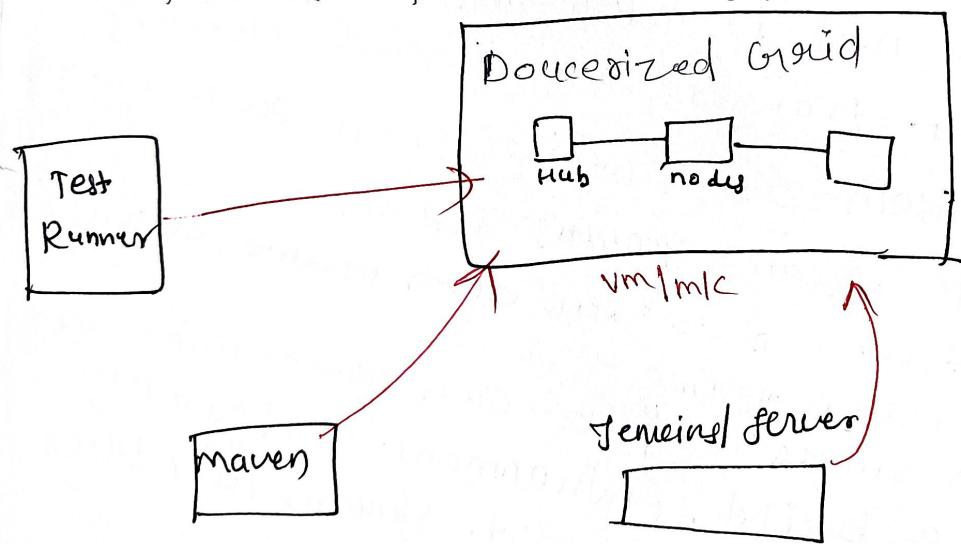
⇒ Remote your own

① Selenium grid / Dockerized grid ⇒

Use RemoteWebDriver class

Remote = true # run test cases on Remote

Remote = false # run test cases on Local.



three ways  
can run our test cases

- ① Right click on testing Class then also can run

maintenance is high cost is high so on VM so we move towards cloud (AWS)

## EC2 - Compute

Linux m/c  
dockerized grid with Selenium

VM / docker

IP : port

→ Every m/c is having their own IP address & port no.

⇒ So, we will configure the IP address with our local / script.

⇒ RemoteWebdriver will decide exactly where your test cases will be executed.  
on AWS or on local or on some HW.

⑥ create cloud on AWS then setup Selenium grid also.

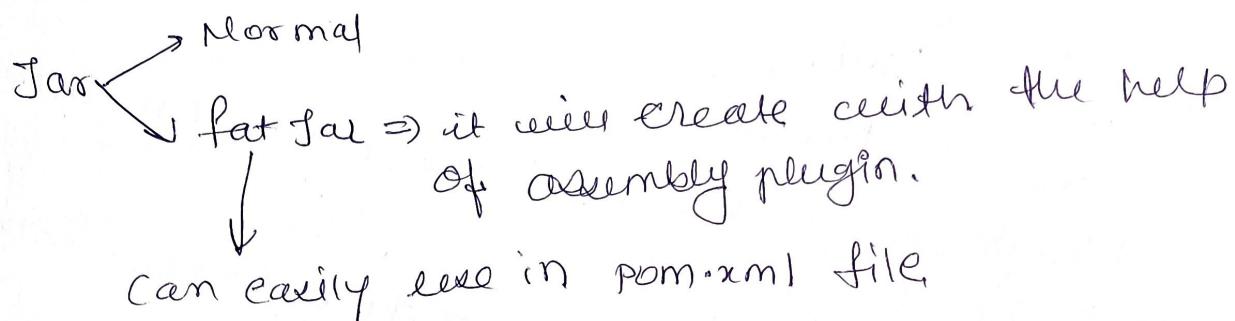
② there are diff grid are available in market like simple, composit, selenium grid.  
In file create our own browser lab.  
on AWS.

Create Infrastructure on AWS cloud.  
we need to whitelist the IP address in AWS so that it will accept.

We can do it execute test cases on AWS through Jenkins,  
i.e. testNG, maven, ,

## Nexus Repo Artifacts →

① After creating maven framework, we need to create jar file of our framework



maven centralized repository ⇒ openly available

Org Specific location ⇒ Nexus repo Artifacts

Execution Environment is also imp.

Autotrigger mechanism will be done by NGRK proxy.

⇒ Running test cases on AWS.