Interface webdriver:

<https://www.javadoc.io/doc/org.seleniumhq.selenium/selenium-api/2.50.1/org/openqa/selenium/WebDriver.html>

Selenium api 3.141.59

<https://www.javadoc.io/doc/org.seleniumhq.selenium/selenium-api/3.141.59/index.html>

selenium IDE [will generate tests but not code]

<https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=en>

Katalon recorder [will generate tests as well as code]

<https://chrome.google.com/webstore/detail/katalon-recorder-selenium/ljdobmomdgdljniojadhoplhkpialdid>

selenium server grid[to run remote selenium webdriver]

<https://www.selenium.dev/downloads/>

Ports used for computer networks

<https://www.lifewire.com/port-numbers-on-computer-networks-817939>

\*Appium- Mobile automation basics

<https://youtu.be/Fx3ud5wMHYc>

TestNG suite

<http://javarticles.com/2015/02/example-of-testng-configuration-xml-testng-xml.html>

TestNG documentation

<http://testng.org/doc/documentation-main.html>

class level annotations in TestNG

<http://www.seleniumeasy.com/testng-tutorials/class-level-annotations-in-testng>

Handle pop ups / chat bots

<https://stackoverflow.com/questions/34721359/chrome-profile-to-disable-know-your-location-pop-up>

<https://www.youtube.com/watch?v=Gu9PPDbHTGo>

MySql installer

<https://dev.mysql.com/downloads/installer/>

Download chrome driver

<https://sites.google.com/a/chromium.org/chromedriver/downloads>

SET env variables for java

1. JAVA\_HOME = C:\Program Files\Java\jdk1.8.0\_301
2. PATH = C:\Program Files\Java\jdk1.8.0\_301\bin

Group\_id = package name and artifact id = project name in maven projects

XPath =

Syntax:

1. //tagName[@attribute=’value’]
2. //tagName[text() = ‘text value’] [ by text- if attr not present]
3. //tagName[contains(@attribute,'value')] [by regular expression – if text is too large and if spaces are there]
4. //tagName[@attribute=’value’][index] [by index – if attr and value r same for 2/more web elements]
5. Absolute xpath: we need to use “/”. To navigate from root of parent to immediate child
6. Relative xpath: we can use “/” or “//”. To navigate from parent to any chid

Notes:

1. Every object may not have id, name, className so xpath and css is preferred
2. Alpha numeric id may vary on every refresh
3. Do not use class name with spaces e.g. class=”abc def” else throw an error “Compound classes cannot be accepted” / “no such element: Unable to locate element” – confirm which error will be thrown
4. Generate xpath- rightclick copy on blue highlighted html code to generate xpath
5. Firepath deprecated from firefox

Firepath for firefox..deprecated

Chropath plugin for chrome to get xpath and css /can be used to validate customize xpath/css

1. To check if you have written correct xpath/css selector….inspect->go to console -> write $x(“ ”) for xpath and $(“ “) for css selector
2. Customized xpath/css generation techniques:

Syntax: xpath - //tagName[@attribute=’value’]

Css- tagName[attribute=’value’]

tagName#id

tagName.classname

[class=’abc def’]

#id

.classname (if there is space between class name replace space with .(dot)

We can skip tag name e.g. #email

-Standard pattern

-tag name traverse

-Regular expression :

|  |
| --- |
| //tagName[contains(@attribute,'value')] - xpath regular expression |
| tagName[Atrribute\*='value'] - Css regular expression  Note: if text is dynamic never recommended to use text based..if static we can use text based locator  //select element based on text displayed on FE  //$x("//button[text()='ADD TO CART']") |

1. Css selectors execute faster as compared to xpath
2. Xpath : //div[@id='start']/button

Css selector : [id='start'] button

1. Selenium Automation:

* Handling static dropdown
* Handling dynamic dropdown
* Handling checkbox
* Handling radio buttons
* Handling text buttons
* Handling alerts- java popups
* Selenium webdriver form methods

1. TestNg is testing framework…assertion will help us to validate if result what selenium returned is expected/not..if not it will fail test
2. Ctrl + shift +f – for formatting(alignment) of code in eclipse
3. Class name- start with capital letter, variable should start with small letter
4. To debug code=> eclipse=>

-click on line no..right click “toggle breakpoint”..

-it will sart execution line by line…

-now instead of run as…right click debug as….script will run…on eclipse confirm perspective switch window will open..

-click switch

-new window will be opened at right side of code….if script pas it will highlight line

- select step over icon….check activity opened on window of chrome

- to jump into function…click step into

1. Synchronization in selenium

* **Implicit wait** – wait for n no of sec before throwing exception….but say if script loads in n-1 seconds if will come out and resume remaining script
* **Explicit wait** – wait for n no of sec for all searches…but wait ‘m’ no of sec for specific locator
* **Thread.sleep** – its part of java(and we r telling that hold below script for n ms)...not selenium…its not recommended to use even in frameworks
* **Fluent wait –** explicit wait achieved in 2 way

1. **WebDriverWait**
2. **Fluent wait –** it finds the web element repeatedly at regular intervals of time until the timeout/ till the object gets found.

Unlike web driver wait, we need to build customized wait methods based on condition.

Both webDriverWait and FluentWait classes implement wait interface

<https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/FluentWait.html>

<https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/Wait.html>

1. Handling Ajax/Mouse interactions

Actions

1. How to mouse over an object with selenium
2. Perform mouse and keyboard interactions with selenium
3. Context click on element
4. Double click on element
5. Drag n drop element

Frames

1. What r frames
2. How to identify frames in application
3. How to handle frames
4. Best practices when working with frames application.
5. \*JavascriptExecutor – to scroll webpage
6. Handle HTTPS certification
7. How to take screenshot
8. Soft assertion
9. Java streams
10. Selenium WebDriver:
11. Selenium WebDriver is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.
12. Operation System Support – Windows, Mac OS, Linux, Solaris  
    Browser Support – Mozilla Firefox, Internet Explorer, Google Chrome 12.0.712.0 and above, Safari, Opera 11.5 and above, Android, iOS, HtmlUnit 2.9 and above
13. Selenium webdriver architecture mainly divided into three parts
14. Language level bindings  
    Selenium Webdriver API  
    Drivers  
    1) Language Level Bindings :  
    You can see at the Left hand side here we’ve got some bindings and these are language level bindings and with which you can implement the Selenium webdriver code. In simple words these the languages in which are making an framework, will interact with the Selenium Webdriver and work on various browsers and other devices. So we have a common API that we use for Selenium that has a common set of commands and we have various bindings for the different languages. So you can see there’s Java, Java, Python, Ruby, there’s also some other bindings and new bindings can be added very easily.  
    2) Selenium Webdriver API:  
    Now these bindings communicate with Selenium Webdriver API and and This API send the commands taken from language level bindings interpret it and sent it to Respective driver. Right now don’t worry about how it works. I will explain them in upcoming posts. In basic term it contains set of common library which allow to send command to respective drivers.  
    3) Drivers:  
    Drivers here at the right hand side, you see we have various internet browser specific drivers such as IE driver, a Firefox, Chrome, and other drivers such as HTML unit which is an interesting one. It works in headless mode which make text execution faster. It also contains mobile specific drivers as well. But the basic idea here is that each one of these drivers knows how to drive the browser that it corresponds to. So the Chrome driver knows how to handle the low level details of Chome browser and drive it to do things like clicking button, going into pages, getting data from the browser itself, the same thing for Firefox, IE, and so on.
15. Selenium Grid: steps (ip address of hub <http://192.168.0.105>)

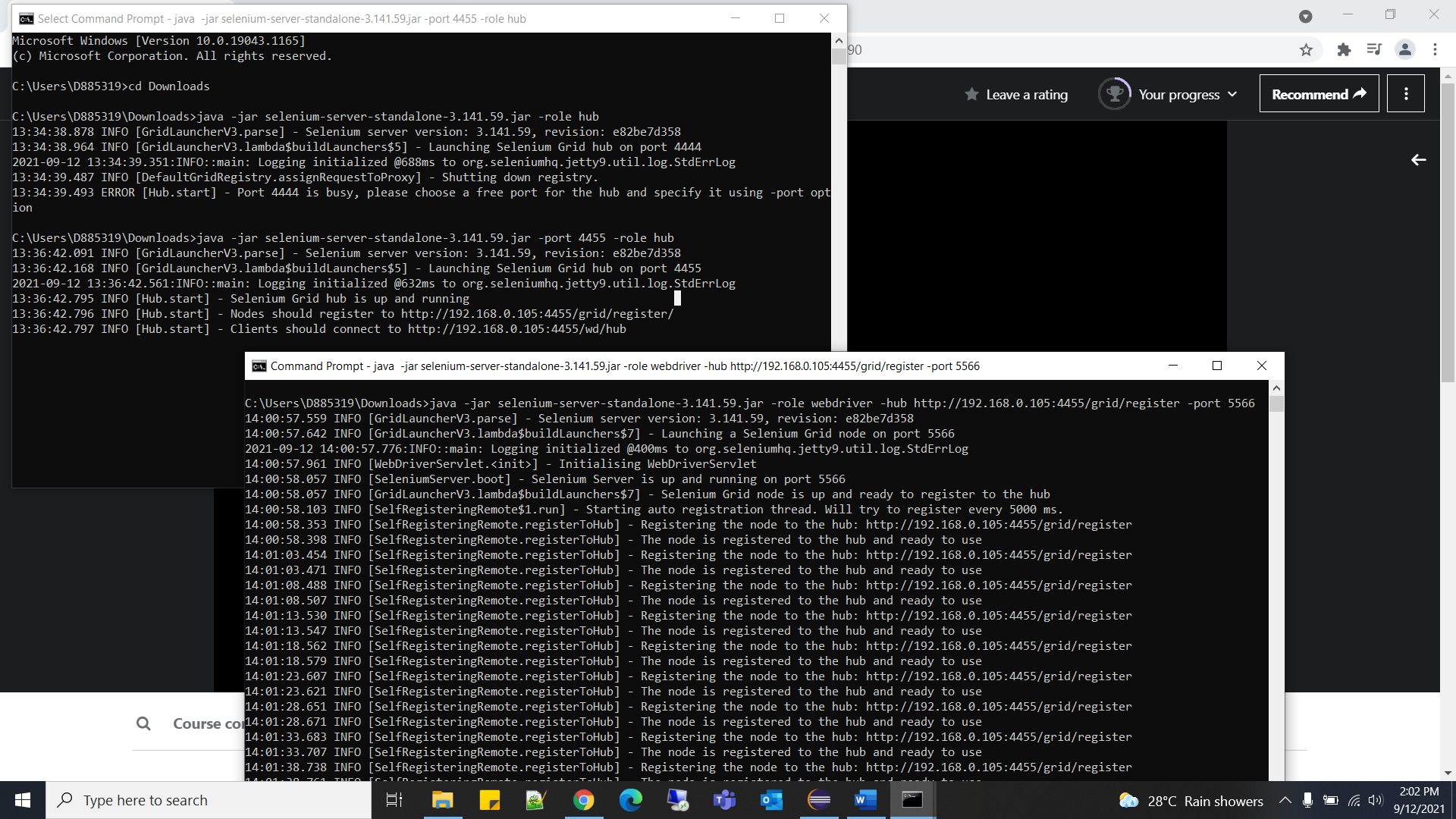
We can configure Hub and node on single machine..open 2 different cmd prompts

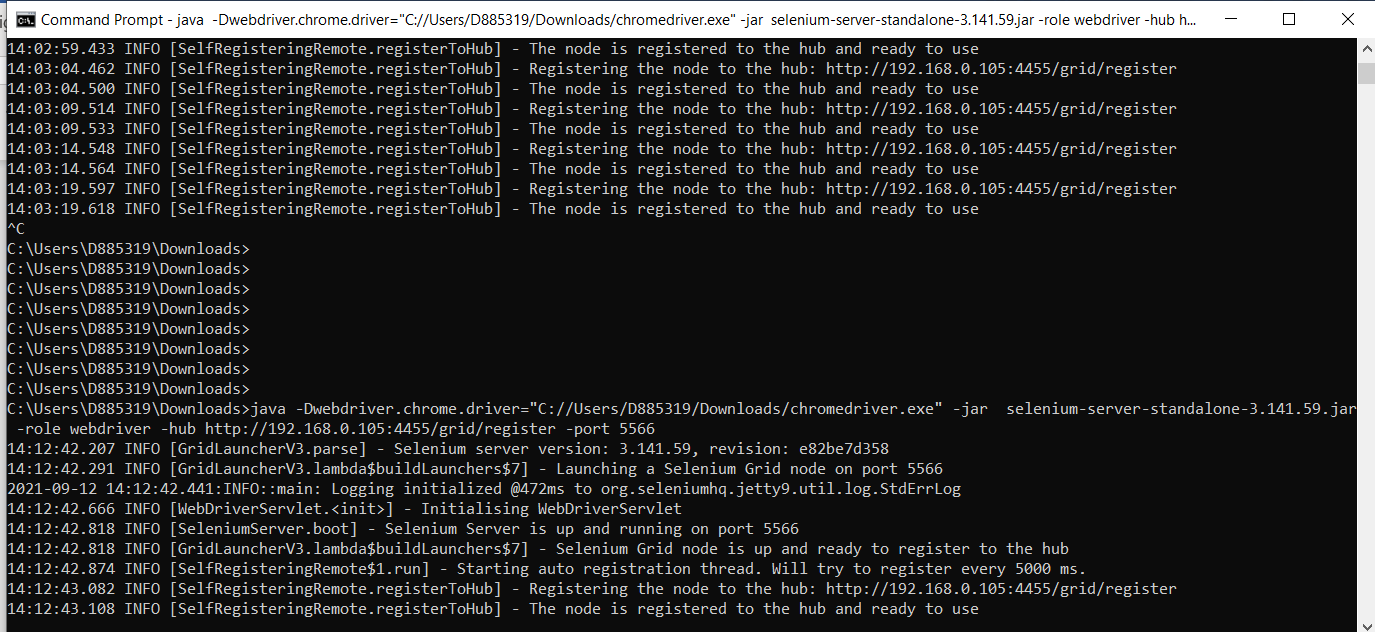
* Download selenium server jar
* Open cmd prompt->navigate to where you have selenium server standalone located-> after locating type

java -jar selenium-server-standalone-3.141.59.jar -role hub

Enter…..if port 4444 is busy type below cmd

java -jar selenium-server-standalone-3.141.59.jar -port 4455 -role hub

* <http://192.168.0.105:4455/grid/console> ……type on browser
* Login to another machine and register it as node for hub…..selenium server should also be downloaded in another machine/node machine as well……….open cmd prompt in node machine and type
* java -jar selenium-server-standalone-3.141.59.jar -role webdriver -hub [http://192.168.0.105:4455/grid/register -port 5566](http://192.168.0.105:4455/grid/register%20-port%205566) enter (on hub port is 4444/4455 and on node machine port will be different)
* after successful, on node machine =>the node is registered to hub and ready to use msg will be displayed, on hub machine=> Registered a node http://192.168.0.105:5566
* before that check java is configured in system variables
* java -Dwebdriver.chrome.driver="C://Users/D885319/Downloads/chromedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role webdriver -hub [http://192.168.0.105:4455/grid/register -port 5566](http://192.168.0.105:4455/grid/register%20-port%205566)
* 



1. Relative locators/friendly locators [available in selenium 4 version ]
2. Above() – element located above with respect to the specified element
3. Below() – element located below with respect to the specified element
4. toLeftOf – element located to left of specified element
5. toRightOf – element located to right of specified element

syntax:

driver.findElement(withTagName(“xx”).above(WebElement))

1. TestNG:
2. We can run test cases without using public static main method, instead we can use @Test annotations followed by method name

- To run test case, right click Run as->TestNG Test

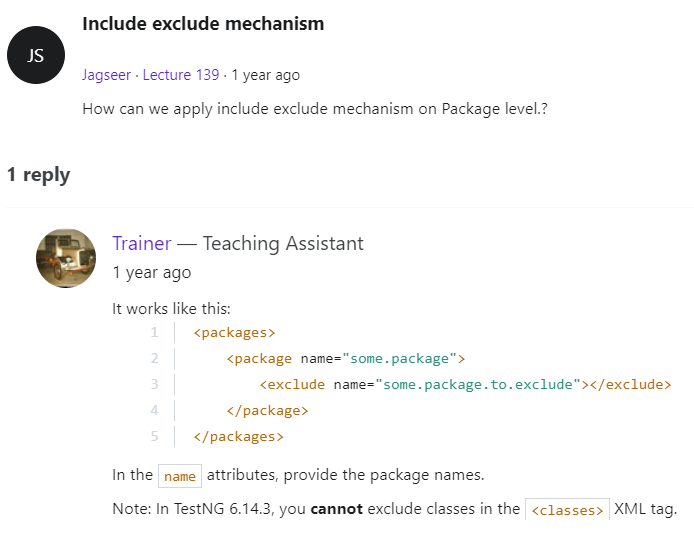
e.g. @Test

**public** **void** Demo() {

System.***out***.println("program without main method");

}

1. We can define multiple tests from single class
2. We can modularize the testcases based up on functionality and trigger them accordingly
3. We can exclude particular test case from class, using exclude tab inside mathods tag
4. **@BeforeClass** is the first one to execute when you run java class and **@AfterClass** is last to execute
5. @Beforeclass for login and @Afterclass  for deleting cookies at the end
6. Can there be 2 before test and 2 after test in same class?If yes then what is the priority => If you given the Priority in then it execute priority wise to show results. If priority it not given then it executes in FIFO order.[usually it will take alphabetical order]



1. Can TestNG run multiple suites?

<https://stackoverflow.com/questions/4501215/can-testng-run-multiple-suites>

1. Groups =>you want to access the test methods of different classes.( if we have 100 test cases in diff classes and want to run only 4 test cases which r present in diff classes so will use groups[we can’t use include/exclude here…cz those test cases belongs to diff classes])

<https://www.javatpoint.com/testng-groups>

1. Helper attributes: (description, enabled, timeOut, priority, dependsOnMethods, groups)
2. (enabled = true) and (enabled=false) =>if there are bugs in my test case and we don’t want to execute that test so will use enabled = false
3. If one of test case is taking more time than usual to execute then will use ‘timeOut’ attribute (timeOut=4000)
4. Parameterization in TestNG: global environment variables can be incorporated through selenium
5. \*DataProvider Annotations
6. TestNG liteners : to get unimplemented methods(Right click(on the Listeners class ) -> go to source-> click on overide/implement methods -> select the **check boxes** for the ITest listener (make sure all check box inside it should be checked )->click on oK)
7. Global env variables setup and reusable components

* How to drive global env values from external files to tests in java
* (create class file give it name. again right click on src->new->other->expand general->file->next->filename.properties)

Graphical user interface, application

Description automatically generated

1. Maven:

* Maven.docx
* <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html> - standard proj structure
* <https://maven.apache.org/surefire/maven-surefire-plugin/usage.html>
* <https://mvnrepository.com/>
* Binary zip archive - <https://maven.apache.org/download.cgi>
* Configure env variable for maven =><https://mkyong.com/maven/how-to-install-maven-in-windows/>
* C:\Users\D885319\apache-maven-3.8.2-bin\apache-maven-3.8.2 MAVEN\_HOME
* C:\Users\D885319\apache-maven-3.8.2-bin\apache-maven-3.8.2\bin Path->edit>add path

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Text

Description automatically generated

* Create maven project = > right click->project->maven->maven project->next->next->org.apache.maven.archetypes quickstart select->next
* Graphical user interface, text, application, email

  Description automatically generated
* Surefire plugin- helps to execute test cases from test folder in Maven project
* Integration testing with maven
* Add following code to pom.xml for surefirle plugin..to run testing.xml
* <configuration>
* <suiteXmlFiles>
* <suiteXmlFile>testng.xml</suiteXmlFile>
* </suiteXmlFiles>
* </configuration>
* <https://maven.apache.org/surefire/maven-surefire-plugin/examples/single-test.html>

1. mvn -Dtest=SeleniumTest test -> to run single test
2. mvn test -PRegression ->if wanna run only regression profile
3. Jenkins

<http://localhost:9090/login?from=%2F>

1. C:\Users\D885319\Downloads> java -jar jenkins.war -httpPort=9090
2. If above one throws error use this one

java -jar jenkins.war --enable-future-java --httpPort=9090

1. 88283cd243624bc58d6f7925ed2c3700 – password

Admin – username

* Create new job in Jenkins
  1. Enter url <http://localhost:9090/>
  2. Enter username and password
  3. Click new item
  4. Enter name of item in textbox and select freeStyle project. Ok
  5. For now no need to select source code management as we are working on localhost, but in real time select git and enter url
  6. Click on advanced(above src code mangmnt)-> use custom workplace ->in directory enter path of project(C:\Users\D885319\Desktop\diksha\udemy\Selenium\E2EProject)
  7. How u are going to build -> if maven then select Invoke top-level maven targets->select maven version - >Goals(what commands need to run): test (instead of mvn test write test)
  8. If want to set time => build triggers->build periodically
  9. Save

To Build project : click on build now on dashboard

1. Extent Reports
2. Excel data driven testing utilities
3. Apache POI API: used to connect Excel to java test cases

Add - Poi ooxml maven dependency in pom.xml

-poi maven dependency in pom.xml

b. Strategy to access excel data

- create object for XSSFWorkbook class

- get access to sheet (cz in real time there will be many sheets)

- get access to all rows of sheet

- access to specific row from all rows

- get access to all cells of row

- access the data from excel into arrays

1. Log4j:

<https://www.apache.org/dyn/closer.lua/logging/log4j/2.14.1/apache-log4j-2.14.1-bin.zip>

What is apache log4j?

Ans: log4j is reliable, fast and flexible logging framework(APIs) written in java, which is distributed under the apache soft license.

Why log 4j?

Ans: send entire log of file?

Log only when there is error for Package A

Log entire message of package B

I want logs with timestamp

I want the log of last week

How will I know if there is an error by just looking at logs?

**Use log. Error()** to log when elements are not displayed in the page or if any validations fail

**Use Log. Debug()**

When each Selenium action is performed like click, SendKeys, getText()

**Use log.info()**

When operation is successfully completed ex: After loading page, or after any successful validations

It’s just counterpart to log. Error()

* <https://logging.apache.org/log4j/2.x/manual/configuration.html> log4j config
* Appenders and loggers

Where to log – appenders tag(appenders ref = console/file…)

What to log – loggers(root level=error/trace)

Error- will print only errors, trace will print all things

How to log – appenders()

-<RollingFile name="File" fileName="${basePath}/prints.log" filePattern="${basePath}/prints-%d{yyyy-MM-dd}.log">  
     <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>  
      <SizeBasedTriggeringPolicy size="500" />  
          </RollingFile>

Explanation : 1. Where to log console/file….file

1. sizeBasedTriggeringPolicy = 500…..aft 500 kb it will create new file…but what should be file name after 500kb so that is defined in filePattern
2. it will create file e.g. prints-yyyy-MM-dd.log(same like console..what should be pattern)
3. keep file in folder at project level

* Page object Pattern: also known as POM(page object model) : is **a design pattern in Selenium that creates an object repository for storing all web elements**. It is useful in reducing code duplication and improves test case maintenance.

(in Laymon lang: if one/more objects are present in multiple test cases so instead of changing in multiple cases will change in one class so it will reflect in all test cases where object Is present)

1. Easy to maintain
2. Easy redability of scripts
3. Reduce/eliminate duplicacy
4. Re-usability of code
5. Reliability

<https://multichoice.udemy.com/course/selenium-real-time-examplesinterview-questions/learn/lecture/7414470#questions/9023044>

\*Object repository mechanism:

* Database Connection

Add mysql connector jar to project

<https://downloads.mysql.com/archives/c-j/>

To connect to DB info to selenium

* 1. Import selenium-server-standlaone to project

Cucumber

* To add cucumber plugin(Eclipse->Help->eclipse marketplace->type cucumber->go->select green colored cucumber eclipse plugin->install)
* Add cucumber-java and cucumber-junit maven dependencies in pom.xml
* Create .feature file: To create .feature file right click on package ->new->other->file->General:File->filename.feature
* Map step definition to feature file :
* To auto generate step definitions : tidy gherkin (<https://chrome.google.com/webstore/detail/tidy-gherkin/nobemmencanophcnicjhfhnjiimegjeo?hl=en-GB>)
* To check above plugin added to chrome/not use: chrome://apps/
* Click on Tg -> will open new window -> paste ur features in that

windowGraphical user interface, text, email

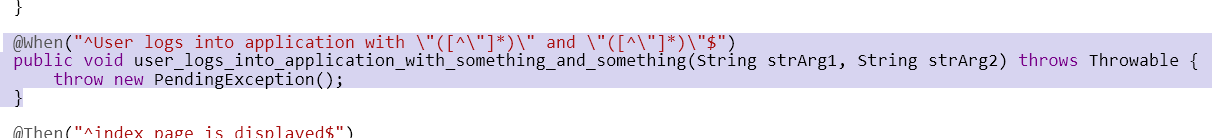
Description automatically generated

and select java steps -> it will automatically create step definitions -> now u can copy and paste the code in step def.

* There should be one mapping implementation for each Gherkin line
* Tidy gherkin plugin to generate stepDefination file
* Cucumber options in test Runner: features path and glue
* Regular expression: for same implementation but different(dynamic) data

Text

Description automatically generated



1. Importance of BDD
2. Cucumber framework architecture and its core functionalities
3. Selenium integration with cucumber.
4. QA Role:

TestCases Specifications in Behaviour Driven Development

Step 1:

Testcases should be defined in a Business level by following a ubiquitous language

Ubiquitous language is a (semi-)formal language that is shared by all members of a software development team — both software developers and non-technical personnel.[15]

Syntax :

**In order to**(Achieve something)-> Business outcome

**As a** (Role)

**I want to** Do this

Example:

In order to Pay Credit Card Payment

As a NetBanking sole owner who have credit section access

I want to Navigate to credit card section, Enter amount and process my Payment

Step 2:

Its time to Decide what Test cases are sufficient to certify Business Scenario working

Syntax for Writing Testcases:

**Given** (what you need to have to perform action), -Prerequists

**When** (performs action)- Action

**Then** (the desired outcome for the user).- Validation

Example:

Given : An account with zero balance

When: I navigate to Credit card Payment Section and click on submit bu giving amount

Then : It should throw error message –fund

Given : An account with sufficent balance who does not have credit card

When: I navigate to Credit card Payment Section

Then :You don't have to access as your are error message

Given : An account with sufficent balance who does not have credit card

When: I navigate to Credit card Payment Section and amount

Then :You don't have to access /error message

Advantages:

This can be used as Standard Template where all QA can stick to one common standards of defining Testcases

Each Scenario reflects a Business Value

We can estimate the Test coverage happened for Each Business outcome by going through Test Cases

We can tag these Annotations to Selenium Automation and execute the Business Testcases

Common Standardised Testcase template for both Manual and Automation testing

Questions:

<https://www.interviewbit.com/selenium-interview-questions/>

<https://www.guru99.com/top-100-selenium-interview-questions-answers.html>

Exp:

<https://www.softwaretestinghelp.com/selenium-interview-questions-answers/>

<https://svrtechnologies.com/selenium-testing-interview-questions-and-answers-pdf/>

<https://www.edureka.co/blog/interview-questions/selenium-interview-questions-answers/>

<https://www.onlineinterviewquestions.com/selenium-interview-questions/>

<https://www.interviewbit.com/automation-testing-interview-questions/>

<https://www.guru99.com/automation-testing-interview-questions.html>

1. Difference between

-RC, webDriver and Grid

-close and quit

- driver.get vs driver.navigate().to()

-xpath (can traverse back) vs css(can’t traverse back)

-relative(does not depend on parent nodes) and absolute xpath(parent/child)…most preferred is relative cz parent may change in future

- assertTrue vs assertFalse (in testNG)

- implicit wait vs explicit wait vs thread.sleep vs fluent wait

- BeforeMethod vs BeforeTest [TestNG]

- == vs .equals

2. What is API

3. What is DefaultSelenium class

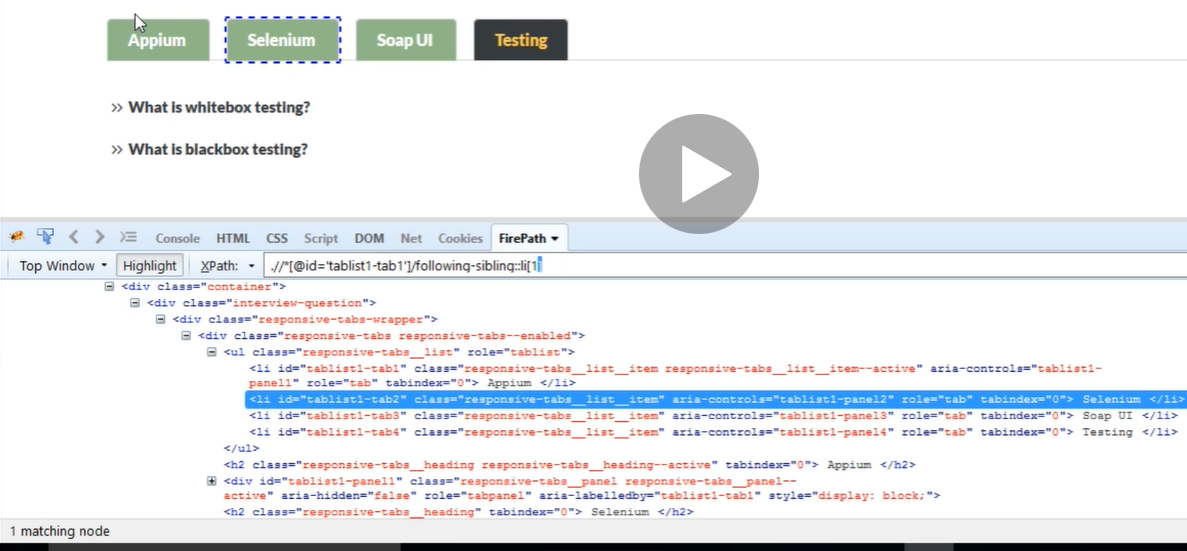
4. if we don’t have any unique attribute to locate the how to locate object

Ans. Using parent-child relationship……like define xpath for parent/tagname

5. How to traverse to sibling element using xpath?

Syntx=> 1st child/following-sibling::tagName[no]

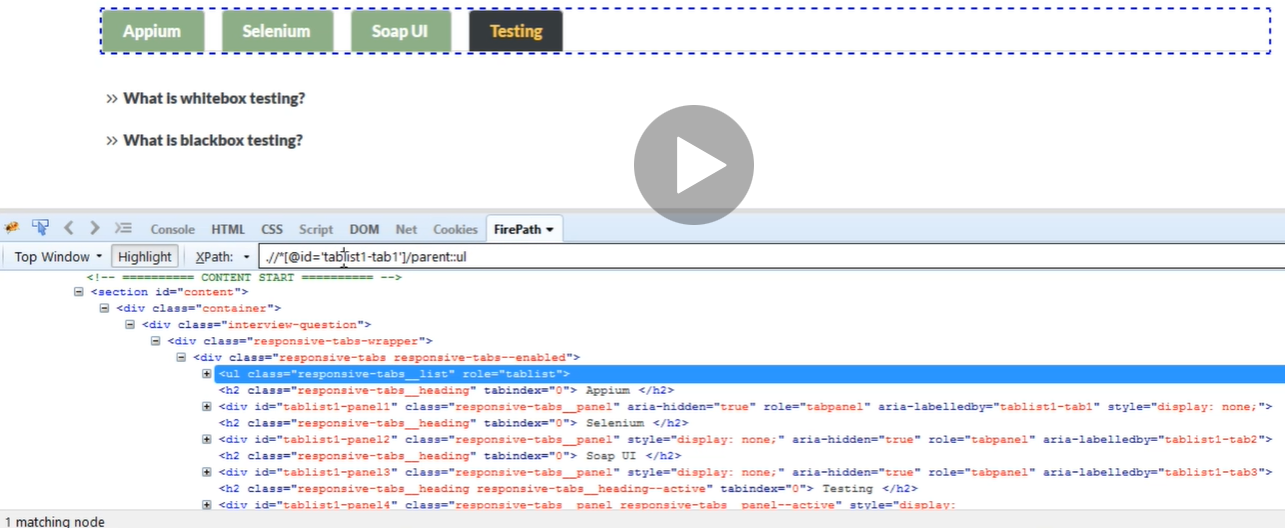
.//\*[@id='tablist1-tab1']/following-sibling::li[2]



6. How to traverse back to parent element from child element using xpath

Ans. We can traverse back to parent using only xpath not css

.//\*[@id='tablist1-tab1']/parent::ul



1. how to identify element with text based

Ans. This scenario is only applicable in agile, before code given to you, u have to write test cases using text displayed on Front End(i.e. UI text). This is considered as hard-coded.

Syntax: //\*[text()=’ Selenium ‘]

Spaces included before and after selenium.

1. What is regression testing and smoke testing
2. Length – to get size of array and size() to get size of arraylist
3. Where we use fluent wait
4. How to write xpath/css selector without using indexes(<https://the-internet.herokuapp.com/dynamic_loading>)
5. How to switch frames => id, web element, index
6. Java collection framework used to store window ids.=> set
7. <https://www.rahulshettyacademy.com/AutomationPractice/>

for these question refer scope.java

get count of all links in web page

get count of all links present at footer

count of links of only 1st column

click on each link in column and check if pages are opening

1. ChromeOptions are used for?
2. How to maximize window and delete cookies
3. Session cookie logout and user should be logout – how to do that
4. How to take screenshot
5. How to check broken links
6. Take amazon/flipkart website...apply filters/pagination/search…as test scenario………..use java streams to test
7. What makes Selenium Unique from other Automation tools?
8. Explain WebDriver Architecture.
9. Why not older Version Selenium RC 1.0?
10. What are different Versions of Selenium?
11. What are different browsers does Webdriver Support?
12. In how many languages we can write Selenium code?
13. Scenario: invokeMultipleTabs\_Windows.java

Navigate to : <https://rahulshettyacademy.com/angularpractice/>

Fill ‘name’ field with the first course name available

<https://rahulshettyacademy.com/#/index>

- capture webElement screenshot using selenium

- Get Height and width of web element using selenium

27. How to pass unique id and password every time through automation(use timestamp with date)…google it

28. what is maven? why maven?

29. using what plugin you will execute all test-cases in maven project?

Ans: maven surefire plugin

30. Suppose we want to publish our project in Maven, what is the mandatory piece of information that is needed for doing that? => groupID, artifactId and version

31. 1. What are technical challenges you have faced in the automation project and how did   you overcome those

2.How do you perform Automation Code Review/ Walk Through in your project

3.How do I debug a bug that you found, whether that bug is product bug or your automation script bug

4.If you were given a chance to develop a framework what framework you will be choose(BDD(Cucumber), POM) why?

32. How to read and write data from excel file using selenium

34. How to handle popups

driver.navigate().to("https://www.flipkart.com");

Actions actions = **new** Actions(driver);

Action action = actions.sendKeys(Keys.***ESCAPE***).build();

action.perform();

1. Common Exceptions:
2. NullPointerException()- **thrown when an application attempts to use null in a case where an object is required**
3. ClassNotFoundException() –
4. NoSuchElementException() –
5. FileNotFoundException() –
6. ArrayIndexOutOfBoundException() –
7. To run test cases parallel use ThreadLocal
8. \*\*Handle popup whether it is displayed or not(cz sometimes popup is displayed and sometimes its not displayed) (lecture - 315) (if size > 0 then element is present)

By popup = by.xpath(“ ”);

Public List<WebElement> popupSize(){

Return driver.findElements(popup); }

Public WebElement popup(){

Return driver.findElements(popup); }

IndexPage ip = new IndexPage(driver);

If(ip.popupSize.size()>0){

Ip.popup().click(); }

1. How to locate svg elements
2. File download code -> handle window authentication popup.docx

Links to questions

<https://www.journaldev.com/26429/selenium-webdriver>

<https://www.google.com/search?q=quit+vs+close+vs+dispose+in+selenium&rlz=1C1GCEU_enIN958IN958&sxsrf=AOaemvJkiTJvIjE0--PR_Up9Xqaf-oMgkw%3A1633238961790&ei=sT9ZYYbAL9XY1sQP18qmMA&ved=0ahUKEwjG3Jmhwa3zAhVVrJUCHVelCQYQ4dUDCA4&uact=5&oq=quit+vs+close+vs+dispose+in+selenium&gs_lcp=Cgdnd3Mtd2l6EAMyBQgAEM0CMgUIABDNAjIFCAAQzQIyBQgAEM0CMgUIABDNAjoHCCMQsAMQJzoHCAAQRxCwAzoECCMQJzoGCAAQCBAeSgQIQRgAULKUAVjnpgFg5agBaAFwAngAgAHpAYgB0A-SAQUwLjcuNJgBAKABAcgBCcABAQ&sclient=gws-wiz>

Summary:

1. List Box: (pg no: - 50 automation\_notes.pdf)

* Identify the list box which need to be handled and store in ref var

i.e. webelement b = driver.findelement(locator)

* Create obj of select class which accept webElement arg
* By using select class obj select by Byindex()/Byvalue()/Byvisibletext()

1. Print total no of options presention list

* Identify the list box which need to be handled and store in ref var
* Create obj of select class which accept webElement arg
* Get all options/elements present in list, call getOptions function and store in ref variable

List<webelement> a = s.getOptions();

* Print using loop

1. Check whether the given list box is single selectable/multi selectable

* Identify the list box which need to be handled and store in ref var
* Create obj of select class which accept webElement arg
* Check list box is single/multiple selectable call ismultiple() and store In ref variable having Boolean return type

1. Verify value which we selected

* Identify the list box which need to be handled and store in ref var
* Create obj of select class which accept webElement arg
* Identify element using getFirstSelectedOption() function and store in ref variable having web element as return type

Webelement b = s. getFirstSelectedOption();

1. Screenshot(pg 61)\*
2. Fetch data from excel\*(pg 65)
3. Iframe(pg 71)
4. Popup(pg 75)
5. Customize listBox(pg 80)
6. @findBy(pg 100)
7. Pom (pg 111)