**Selenium Automation Notes**

Contents

[1 Selenium feature 3](#_Toc105924760)

[1.2Different between Selenium and web driver? 3](#_Toc105924761)

[2 Selenium Webdriver architecture 3](#_Toc105924762)

[3 Selenium simple project 3](#_Toc105924763)

[4 Testing Framework 3](#_Toc105924764)

[4.1Java class kaise banaye 3](#_Toc105924765)

[4.2First selenium project 3](#_Toc105924766)

[5 Web Driver 3](#_Toc105924767)

[6 Locators in Selenium 3](#_Toc105924768)

[6.1What is Locators in Selenium? 3](#_Toc105924769)

[6.2Different Types of Locators In Selenium WebDriver 3](#_Toc105924770)

[**6.2.1 ID Locator** 3](#_Toc105924771)

[**6.2.2Name Locator** 3](#_Toc105924772)

[**6.2.3Class Name Locator** 3](#_Toc105924773)

[**6.2.4Tag Name Locator** 3](#_Toc105924774)

[**6.2.5 Link Text Locator** 3](#_Toc105924775)

[**6.2.6 PartialLink Text Locator** 3](#_Toc105924776)

[**6.2.7 CSS Locator** 3](#_Toc105924777)

[**6.2.8 Xpath Locator** 3](#_Toc105924778)

[**6.2.9 Xpath Axes** 3](#_Toc105924779)

[7 Selenium project and imp concept 3](#_Toc105924780)

[8 Synchronization problem in selenium(Wait mechanism) 3](#_Toc105924781)

[8.1 Selenium Web Driver Waits 3](#_Toc105924782)

[8.1.1 Implicit wait 3](#_Toc105924783)

[8.1.2 Explicit Wait in Selenium 3](#_Toc105924784)

[8.1.3 fluent Wait in Selenium 3](#_Toc105924785)

[9 What are Assertions inTestNG? 3](#_Toc105924786)

[10 Automate browser navigations and window 3](#_Toc105924787)

[11 Static dropdown 3](#_Toc105924788)

[12 Dynamic dropdown 3](#_Toc105924789)

[13 Auto suggestive dropdown 3](#_Toc105924790)

[14 handling the check boxes 3](#_Toc105924791)

[15 Importance of the Assertions 3](#_Toc105924792)

[16 Current date calender 3](#_Toc105924793)

[17 What is Alert in Selenium? 3](#_Toc105924794)

[17 Mouse hover action {Action class} 3](#_Toc105924795)

[18 Keyboard action{Action class} 3](#_Toc105924796)

[19 Methods of actions class 3](#_Toc105924797)

[20 Windows handle 3](#_Toc105924798)

[21 scolling the window 3](#_Toc105924799)

[22 SSLcertification in selenium 4](#_Toc105924800)

[23 Extentions of plugins in selenium 4](#_Toc105924801)

[24 Proxy server related issues 4](#_Toc105924802)

[25 Cookies and cache file delete 4](#_Toc105924803)

[26 Screenshort 4](#_Toc105924804)

[27 Brokon Link 4](#_Toc105924805)

[28 Special feature for selenium 4 version 4](#_Toc105924806)

[Screenshot 4](#_Toc105924807)

[Open the new tab on the browser: 4](#_Toc105924808)

[Open the new window on the browser: 4](#_Toc105924809)

[Object Location: 4](#_Toc105924810)

[Relative Locators: 4](#_Toc105924811)

[29 Poject of rahul shetty or his instraction 4](#_Toc105924812)

# 1 Selenium feature

* Selenium is an open source and portable Web testing Framework.
* It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.
* Selenium supports various operating systems, browsers and programming languages. Following is the list:
  + Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
  + Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
  + Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
* Selenium can be integrated with frameworks like Maven for source code compilation.
* Selenium can also be integrated with testing frameworks like TestNG for application testing and generating reports.
* Selenium ne kafi sare tools relese kiye hai like selenium IDE and selenium webdriver
* But selenium IDE koi use nahi karta because yeh sirf quick bug reproduction scripts,ko create karta hai.
* So sabhi selenium “Webdriver” use karte hai because “If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver”
* “Selenium remote control” kafi jada out dated ho gya hai so aaj ke time me koi bhi “Selenium remote control” use nahi karta hai.Uske ander kafi sari error the isliye uska extended version hi “Selenium Web driver” hai.
* Aaj ke time “Selenium webdriver” hi sabhi jagah use hota hai

## 1.2Different between Selenium and web driver?

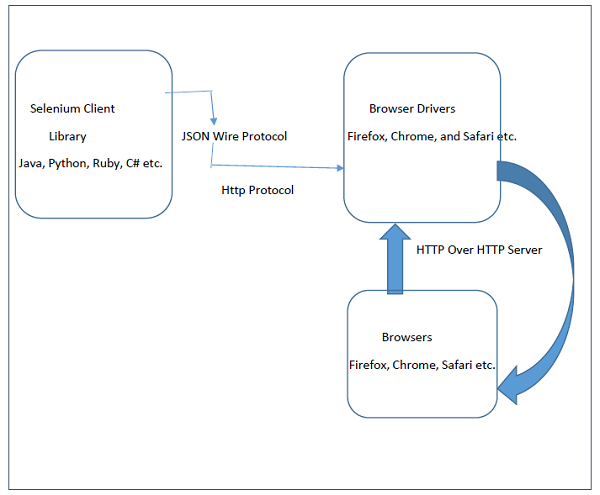
* Selenium is a suite of tools to automate web browsers across many plateform
* Selenium ne two tools launch kiya hai

1) Selenium webdriver 2) Selenium IDE

* Selenium indono ka parent tool hai
* Today Selenium IDE use nahi hota hai
* Initial time me “Selenium Remote control” launch kiya gya tha but kafi sare issue hone ke vajah se uska updated version “Selenium webdriver” market me launch kiya gya hai
* No one now using Selenium RC right now

# 2 Selenium Webdriver architecture

* “Selenium WebDriver API” enables interaction between browsers and browser drivers
* This architecture consists of four layers namely the Selenium Client Library, JSON Wire Protocol, Browser Drivers and Browsers.
* Selenium Client Library consists of languages like Java, Python, C# and so on. After the test cases are triggered, entire Selenium code will be converted to Json format.
* Generated Json, server {Browsers Drive} ke pass jata hai through http protocol
* Note each browser contain separate browser driver
* Jab code browser driver ke pass pahuchta hai to voh usko interpreted karke browser ko send karta hai
* Browser jo bhi response karta hai voh vapas browser driver ke pass jata hai
* Browser driver usse vapas json me convert karke client {Jaha humne code likha hai} ko share karta hai
* Code direct Browser se interect nahi karta hai



# 3 Selenium simple project

* First hume “Browser Driver” apne System me install karna hoga
* Java based project banana ke liye jo bhi library ki need hoti hai voh hame “maven repository” se milta hai.
* Selenium project bana hai to hame “maven based project” par work karna hota hai
* So eclipse par maven project banao
* Step for how to create project in eclipse

File->New->Java Project->{Mention name of the project}->Next -> Finish-{agar koi pop window dikhe}-> Don’t create

* Ab hum is project ko maven based banayege

Step : Right click on the project->configue->convert to maven project

Hume ek pop up window dikhega “Create new POM”

Group ID -> Mention

Artifact ID-> Mention

Version-> Given

Pacaging ->Given

Click on the finish

* Now maven based project ban gya
* Automaticaly POM.xml folder is reflected in the project
* Ab POM file me apna selenium ka four line ka code add karege jo hume “maven repository” se milega
* Maven base project me POM name ka folder hoga jisme “selenium java” ka 4 line ka code likhna hi hota hai

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.1.2</version>

</dependency>

* Ab humare Project ko selenium tool ka knowledge ho gya hai

# 4 Testing Framework

* TestNG ek testing framework hai jo unit testing ke liye use hota hai
* Cucumber bhi ek testing framework hai

## 4.1Java class kaise banaye

* Ek project banao
* Right click on the scr->Click on New-> Class->{Class name}->finish
* Apna Selenium ka code hum java class me likhege

## 4.2First selenium project

* Src me java class ka folder banao
* Chrome browser ko invoke karna hai to hume “chrome driver class” banana hoga
* Chrome driver class ke object ki help se hum uske method ko use karege jo chrome browser ko invoke karega

ChromeDriver driver = new ChromeDriver();

# 5 Web Driver

* Webdriver ek interface hai jiske pass empty body method hota hai sath hi sath sabhi “browser ke class” uss method ko impleament karte hai jo web driver unko bolta hai

WebDriver driver = new ChromeDriver();

* Hum aisa likhte hai agar hum [ChromDriver driver = new ChromeDriver ()] likhe to driver object webDriver interface ke method ko call karega sath hi sath chromeDriver ke class ke methods ko bhi call karega jiske vajah se issue aa skta hai isliye hum WebDriver driver = new ChromeDriver(); hi likhege and second reason aisa likha to humara code dusare browser me nahi chalega
* Selenium direct chromeDriver ko call nahi kar skta hai isliye hume apne system me ChromeDriver ka “exe file” download karna hoga
* .Exe file ko invoke karne ke liye code System.setproperty(“webdriver.chrome.driver”,exe file ka path likho jo aap ne download kiya hai and .exe )
* **Note System.setproperty(“property”,”Value”) ka combination hai**
* Note: hum jo exe file download karte hai voh zip file hota hai to pahle usse unzip karege fir uska path use karege
* Kisi Web page par land karna hai to driver.get(“URL”)
* Page ka title janna ho to driver.getTitle()
* Current page ka Url janna ho to driver.getCurrentUrl()
* Current page ko close karne ke liye driver.close();
* Sabhi page ko close karna hai to driver.quit();
* Note : hum dono close method ko ek sath use nahi kar skte
* Mozila fire fox ka driver name Gecodriver hai jise hume download karna hota hai

System.setProperty("webdriver.gecko.driver", "/Users/rahulshetty/Documents/geckodriver.exe");

* Microsoft Edge

System.setProperty("webdriver.edge.driver", "/Users/rahulshetty/Documents/msedgedriver.exe");

# 6 Locators in Selenium

## 6.1What is Locators in Selenium?

In Selenium we are using locators to find the elements in a web application. So locators are nothing but the address by using we can identify the web elements uniquely in a web page.

we are using “findElement/findElements” syntax {methods} which parallel helps us in locating the elements.

## 6.2Different Types of Locators In Selenium WebDriver

In a web page, we can find out different types of web elements like text box, id, radio button, etc. for locating those elements we need an effective and accurate approach to identify these elements.

Types of locators. They are as follows:

* ID
* Name
* Class Name
* Tag Name
* Link Text & Partial Link Text
* XPath
* CSS Selector

### **6.2.1 ID Locator**

In a web page for some elements, ID’s are unique, which is a common way to locate the element using the ID locator. ID’s are supposed to be unique for the page so that it makes the most reliable locator. [**ID locators**](https://www.softwaretestingo.com/id-locators-selenium-webdriver/) are one the fastest and safest locators out of all locators.

--->> **findElement(**By.id**(**"IdName"**))**;

### **6.2.2Name Locator**

Like ID locator sometimes we can use Name locator to identify the elements on the web page. For a page the name locator details are not unique, if there are multiples elements have the same name locator on a page then your test may fail.

--->> **findElement(**By.name**(**"Name"**))**;

### **6.2.3Class Name Locator**

Its the same as the Name locator, but with this locator, you can find the element which matches the value specified in the attribute name “class”.

--->> findElement(By.className("Element Class"));

Note : Class Name locater ka use karne par kafi error aata hai

Senario : agar DOM me class atribute ki value "inputtext \_58mg \_5dba 2ph-“ aise me di hoto hum class locater ka use nahi karege because is value me three space diya hai jo three different different class ko represent kar raha hai to aise situation me hum kisi ek class ka use karege ya to hum Xpath ka use karege | If it is not marked with spaces hence it is considered as a single class. To hum use kar skte hai.

--->> findElement(By.className("inputtext "));

OR

--->> findElement(By.className("58mg "));

OR

--->> findElement(By.className("5dba 2ph- "));

OR

--->> findElement(By.className("inputtext .s58mg .5dba 2ph "));

Note: agar xpath use kar rahe hai to hum pure class na mane dege like :

//tagname[@class=’ inputtext \_58mg \_5dba 2ph-’]

### **6.2.4Tag Name Locator**

Like the other locator, we can use the Tag name locators to find the elements which have matching the specified tag name. If in the web page that does not have ID, class, or name and you need to locate the elements. In that case, we are going to use the tagName locator only.

--->> **findElement(**By.tagName**(**"HTML Tag Name"**))**;

Note : DOM page me ek jaise sare tag ko identify karne ke liye hum tag locator ka use karte hai

1. driver.findElements(By.tagName (<"button">))
2. driver.findElements(By.tagName (<"input">))

* kafi rare cases me iska use hota hai but kafi helpful hai

### **6.2.5 Link Text Locator**

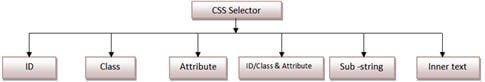
* Most of the web pages we can see the Link, So to handle such hyperlinks (links) elements in web page user can use the Link Text locators. If there are multiple link text elements are present in a web page then the first element will be selected.
* **findElement(**By.linkText**(**"LinkText"**))**;
* Note : DOM page me kafi sare button hote hai jine click karne par voh hume dusre page par le jata hai to aise cases me hum link text ka use karte hai
* Yeh sirf link ke liye use hota hai
* Iska html tag “a” hota hai
* Note:Humesa yad rakhna href ki property kabhi bhi use nahi karna chahiye
* Agar kisi attribute ke bina ki koi value hume dikh rahi ho to hum uska use linkText me kar skte hai

### **6.2.6 PartialLink Text Locator**

* Aise to iska use kafi kam hota hai Yeh LinkText locator jaisa hi hai | Jab Dom page me koi link ki value kafi badi likhi ho to hum partial Link Text ka use karke value ka begnning vala part likh skte hai jiske vajah se apna script achha dikhega |

### **6.2.7 CSS Locator**

Css locator ka use hum below combination ke base par karte hai

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Using-CSS-Selector-as-a-Locator.jpg)

* Jab kisi web element tak pahuchne ke liye koi bhi attribute present na ho sirf or sirf “Tag”hi diya ho to hum css ki help se uss tag ka use karke bhi hum us web element tak reach kar skte hai

EX: form input[2]

Parent tag child tags

#### **6.2.7.1 CSS Selector: ID**

**Syntax**

css=<HTML tag><#><Value of ID attribute>

* **HTML tag** – It is the tag which is used to denote the web element which we want to access.
* **#**– The hash sign is used to symbolize ID attribute. It is mandatory to use hash sign if ID attribute is being used to create CSS Selector.
* **Value of ID attribute**– It is the value of an ID attribute which is being accessed.
* The value of ID is always preceded by a hash sign.
* Tag+ID ex -> input[id='firstName'] or “css=input#Email”

#### **6.2.7 .2 CSS Selector: class**

**Syntax**

css=<HTML tag><.><Value of Class attribute>

* **.**– The dot sign is used to symbolize Class attribute. It is mandatory to use dot sign if a Class attribute is being used to create a CSS Selector.
* The value of Class is always preceded by a dot sign.
* Tag+Class ex-> input[class='lastName'] or css=label.remember

#### **6.2.7 .3 CSS Selector: Attribute**

**Syntax**

css=<HTML tag><[attribute=Value of attribute]>

* **Attribute**– It is the attribute we want to use to create CSS Selector. It can value, type, name etc. It is recommended to choose an attribute whose value uniquely identifies the web element.
* **Value of attribute**– It is the value of an attribute which is being accessed.

#### **6.2.7 .4 CSS Selector: Tag+Class+Attribute**

**Syntax**

* css=<HTML tag><. Or #><value of Class or ID attribute><[attribute=Value of attribute]>
* Two or more attributes can also be furnished in the syntax.
* EX=input#Passwd[type=’password’][name=’Passwd’]”*.*

### **6.2.8 Xpath Locator**

**What is XPath in Selenium?**

**XPath in Selenium** is an XML path used for navigation through the HTML structure of the page. It is a syntax or language for finding any element on a web page using XML path expression. Xpath ke use me hum index ka help le skte hai.but note : indexing ka use hum sbse last prioritypar use karte hai..jab sare locator ke concept help na kar paye tab..

**Syntax for XPath selenium:**

Xpath=//tagname[@attribute='value']

1. **//**: denotes the current node
2. **tagname**: denotes the tagname of the current node
3. **@**: is the Select attribute
4. **Attribute**: denotes the attribute of the node
5. **Value**: denotes the value of the chosen attribute

#### **6.2.8.1 Types of XPath in Selenium**

The XPath is the language used to select elements in an HTML page. XPath can be used to locate any element on a page based on its tag name, ID, CSS class, and so on. There are two types of XPath in Selenium.

1. Absolute XPath
2. Relative XPath

#### **6.2.8.2 Absolute XPath**

* Absolute Xpath is the simplest form of XPath in Selenium. It starts with a single slash ‘/’ and provides the absolute path of an element in the entire DOM.
* Note: log absolute xpath use nahi karte hai kyuki agar developer ne web element ka location change kardiya to absolute xpath ka flow ek dum change ho jayega jiske vajah se hamara script fail ho jayega
* Script dekhne me bhi achha nahi lagta hai

Ex: /html//div/div/div/div[1]/div/a/img

* Note:Yaha hum atribute and value ka use nahi karte hai

#### **6.2.8.3 Relative XPath**

* In the case of relative XPath in Selenium, the XPath expression starts from the middle of the DOM structure. It is represented by a double slash ‘//’ denoting the current node.

Ex: //img[@alt='LambdaTest']

* Jab kisi web element tak pahuchne ke liye koi bhi attribute present na ho sirf or sirf “Tag”hi diya ho to hum xpath ki help se uss tag ka use karke bhi hum us web element tak reach kar skte hai

EX: //form/input[2]

//Parent tag/child tag

#### **6.2.8.4 Basic XPath in Selenium**

* This is the common and syntactic approach to writing the XPath in Selenium, which combines a tagname and attribute value

EX:XPath = //tagname[@Attribute=’Value’]

#### **6.2.8.5 XPath using Contains**

* Contains() is a very useful method in XPath. It can be used for all such web elements whose value can change **dynamically**. The syntax for using Contains() method in XPath is

EX: //tagname[contains(@attribute,constantvalue)]

#### **6.2.8.6 XPath using Logical Operators: OR & AND**

* In the case of OR, any one of the conditions should be true or both, whereas, in the case of AND, both the conditions should be true.

The syntax for using these operators are:

OR

XPath=//tagname[@attribute1=value1 OR @attribute2=value1]

AND

XPath=//tagname[@attribute1=value1 AND @attribute2=value1]

#### **6.2.8.7 XPath using Text()**

* The text() method is used in XPath whenever we have a text defined in an HTML tag, and we wish to identify that element via text.

EX: //tagname[text()=’Text of the Web Element’]

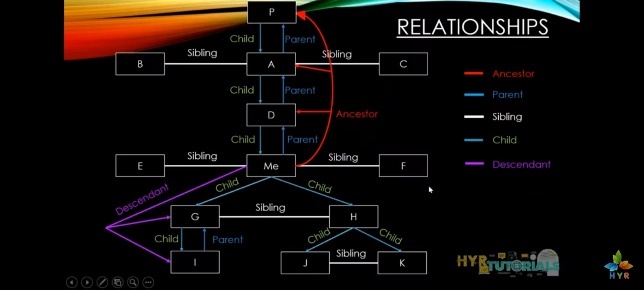
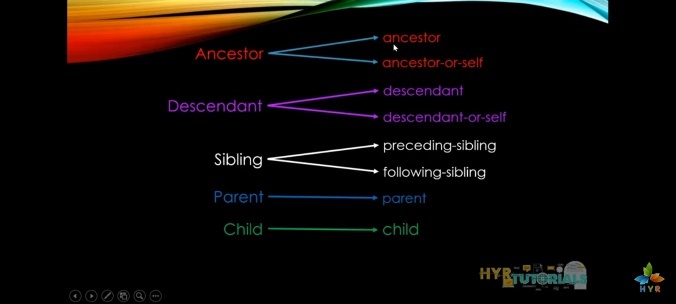
#### **6.2.8.8 XPath using Starts-With()**

* The Starts-With() method is similar to the Contains() method. It is helpful in the case of web elements whose attribute value can change dynamically. In the Starts-With method, the starting value of the attribute’s text is used for locating the element.

EX: //tagname[starts-with(@attribute,value)]

### **6.2.9 Xpath Axes**

* **XPath axes in Selenium** are methods to identify those dynamic elements which are not possible to find by normal [XPath](https://www.scientecheasy.com/2020/07/selenium-xpath-example.html/) method such as ID, Classname, Name, etc.
* Axes are so named because they tell about the axis on which elements are lying relative to an element.
* XPath axes help to find elements based on the element’s relationship with another element in an XML document.
* These XPath axes methods are used to find the complex or dynamic elements.
* Jab kisi web element ka hum xpath nahi nikal pate hai tab hum xpath Axes ki help se uss web element ke parent ya child ki helps se uska xpath nikalte hai



Note: <https://youtu.be/aAWvwGFkySI> link for more study

#### **6.2.9.1 XPath Axes Methods in Selenium**

* To navigate the hierarchical tree of nodes in an XML document, XPath uses the concept of axes. The XPath specification defines a total of 11 different axes that we will learn in this section.
* A list of 11 XPath axes methods in Selenium WebDriver is as follows:

1. Child Axis
2. Parent Axis
3. Self Axis
4. Ancestor Axis
5. Ancestor-or-self Axis
6. Descendant Axis
7. Descendant-or-self Axis
8. Following Axis
9. Following-sibling Axis
10. Preceding Axis
11. Preceding-sibling Axis

|  |  |
| --- | --- |
| ancestor | Selects all ancestors (parent, grandparent, etc.) of the current node |
| ancestor-or-self | Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself |
| child | Selects all children of the current node |
| descendant | Selects all descendants (children, grandchildren, etc.) of the current node |
| descendant-or-self | Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself |
| Self | Selects the current node |
| preceding | Selects all nodes that appear before the current node in the document |
| preceding-sibling | Selects all siblings before the current node |
| Following | Selects all nodes that appear after the current node in the document |
| following-sibling | Selects all siblings after the current node |
| Parent | Selects the parent of the current node(yaha sirf ek result milega) |

#### **a) Following:**

* Selects all elements in the document of the current node( ) [ UserID input box is the current node] as shown in the below screen.

Xpath=//\*[@type='text']//following::input

Xpath=//\*[@type='text']//following::input[1]

You can change the XPath according to the requirement by putting [1],[2]…………and so on.

#### **b) Ancestor:**

* The ancestor axis selects all ancestors element (grandparent, parent, etc.) of the current node as shown in the below screen.
* In the below expression, we are finding ancestors element of the current node(“ENTERPRISE TESTING” node).

Xpath=//\*[text()='Enterprise Testing']//ancestor::div

#### **c) Child:**

* Selects all children elements of the current node (Java) as shown in the below screen.

Xpath=//\*[@id='java\_technologies']//child::li

#### **d) Preceding:**

* Select all nodes that come before the current node.
* In the below expression, it identifies all the input elements before “LOGIN” button that is **Userid** and **password** input element.

Xpath=//\*[@type='submit']//preceding::input

#### **e) Following-sibling:**

* Select the following siblings of the context node. Siblings are at the same level of the current node as shown in the below screen. It will find the element after the current node.

xpath=//\*[@type='submit']//following-sibling::input

#### **f) Parent:**

* Selects the parent of the current node as shown in the below screen.

Xpath=//\*[@id='rt-feature']//parent::div

* Note: yaha sirf or sirf ek element milta hai because ek node ka sirf ek hi parent ho skta hai

#### **g) Self:**

* Selects the current node or ‘self’ means it indicates the node itself as shown in the below screen.

Xpath =//\*[@type='password']//self::input

#### **h) Descendant:**

* Selects the descendants of the current node
* In the below expression, it identifies all the element descendants to current element ( ‘Main body surround’ frame element) which means down under the node (child node , grandchild node, etc.).

Xpath=//\*[@id='rt-feature']//descendant::a

# 7 Selenium project and imp concept

* Inspect karne par jo window dikhta hai usse developer tools kahte hai
* Har attribute ka koi value hota hai some time koi attribute aisa bhi mil skta hai jiska koi bhi value n ho and koi aise bhi value mil skte hai jista koi bhi attribute n ho
* Find Element and Find Elements dono ek method ka name hai
* “Sendkeys” ek action perform karne ka method hai jisme hum web page par input dete hai
* Agar class attribute ki value me space diya hoto hume sumjhna hai ki us patricuter web element ko represent karne ke liye 2 class ka use kiya gya hai to hum kisi bhi ek class value ka use kar skte hai note🡪 hum dono class value ka use nahi karege otherwise humara code fail ho jayega

Ex: Class =” submit signBtn” agar hum class locator ka use kar rahe hai to hum in dono me se kisi ek value ka use karege

Driver.findElement(By.className(“submit”)).click();

* Button ko press karne ke liye hum .click() method ka use karte hai
* Generic method for CSS selector :-> Tagname[attribute=’Value’]

Css me jo tagname likhte hai voh optional hota hai likho to bhi chalega na likho to bhi chalega

* Chropath , seleniumHub plugins ko download karne par voh hume unique locator value use karne me help karta hai
* getText() method ki help se hum web page se kuch bhi text le skte hai
* kisi text field ki value ko clean karna ho to hum .clear() method ka use karte hai

# 8 Synchronization problem in selenium(Wait mechanism)

* humara automation script and web page ke response dene ke time kabhi kabhi mismatch hota hai aise time me hume wait mechanism ka use karna hota hai
* Why Do We Need Waits In Selenium?

Most of the web applications are developed using [Ajax](https://www.guru99.com/php-ajax.html) and [Javascript](https://www.guru99.com/interactive-javascript-tutorials.html). When a page is loaded by the browser the elements which we want to interact with may load at different time intervals.

Not only it makes this difficult to identify the element but also if the element is not located it will throw an “**ElementNotVisibleException**” exception. Using Selenium Waits, we can resolve this problem.

## 8.1 Selenium Web Driver Waits

1. Implicit Wait
2. Explicit Wait
3. Fluent wait
4. Thread.sleep

8.1.1 Implicit wait :- yaha hum ek time likhte hai jo second, minute, hour, mili seconds ka ho ta hai iska meaning yeh hai ki jab tak web element na mile tab tak aap ko utne time duration tak wait karna hai fir hi next line ka code execute karna hai

* Agar utne time tak bhi agar web element nahi millta hai tab hi aap “exception throw ”karege otherwise utne time tak aap ko web element ko search karte rahna hai
* ‘n’time tak wait karo uske baad hi aap ko ‘exception throw’karna hai but let suppose koi web element n+2 time par load hota hai fir bhi vaha exception aa jayega because voh sirf n time tak hi wait karega
* Iska default value o “Zero” hota hai
* Implicit wait code ke sabhi line par apply hota hai
* Agar given time se pahle agar web element mil jata hai to aage vali code line execute ho skti hai.
* Yeh hum apne code me sirf ek hi baar use karte hai
* Syntex : driver.manage().timeouts().implictlyWait(Duration.ofSeconds(10));

OR

* driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
* Note “Duration” class ko project me import karvana hoga
* Ek disadvanage hai : given time duration ke baad hi hume exception ke bareme pata chalta hai usse pehle nahi malum padta hai

### 8.1.2 Explicit Wait in Selenium

* Explicit wait ke two type hota hai 1) WebDriverWait 2) fluent wait dono class wait interface me aate hai
* But normally log webDriver wait use kartwe hai fluent wait jyada use nahi hota hai
* The **Explicit Wait in Selenium** is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or maximum time exceeded before throwing “ElementNotVisibleException” exception. It is an intelligent kind of wait, but it can be applied only for **specified elements**
* Explicit wait kisi particuler web element ke liye use kiya jata hai
* Explicit wait , implicit wait se jada achha hai
* Explicit wait kyu use kiya jata hai uske 3 example share karta hu

**Condition 1-** I have a web page which has some login form and after login, it takes a lot of time to load Account page or Home page. This page is **dynamic** it means sometimes it takes 10 seconds to load the homepage, sometimes its 15 second and so on. In this situation, the Explicit wait can help us which will wait until **specific page/page title is not present** it will keep waiting.

* **Condition 2-** You are working on travel application and you have filled the web form and clicked on submit button. Now you have to wait until complete data is not loaded or specific data is not loaded. In this case, again we can use Explicit wait in which we can give wait till specific or set of elements are not loaded.
* **Condition 3-** There are some elements on a web page which are **hidden** and it will be displayed only **when specific conditions get true**, so we have to wait until these elements are not visible. In this case, again explicit wait will help in which we can specify wait till the element or elements are not visible.
* Note:***ExpectedConditions is a class*** in Selenium which has some predefined condition which makes our task easy.

#### 8.1.2.1What is Explicit wait in selenium webdriver

* It is a concept of the **dynamic wait** which wait dynamically for specific conditions. It can be implemented by “WebDriverWait” class.

Steps1: Create object of WebDriverWait class (explicit wait ke liye selenium me ek class predefined hai)

Steps 2: WebDriverWait wait=new WebDriverWait(driver,20); (yaha hum “driver object” ko bracket me likhte hai and time mention karte hai)

Steps 3 : fir hum ek condition add karte hai like “visibilityOfElementLocated” // Wait till the element is not visible

WebElement element=wait.until(ExpectedConditions.visibilityOfElementLocated(**By.xpath("ur xpath here"**)));

Element.click();

Yaha agar condition true hoti hai to voh “element” variable me save hoga but agar condition true nahi hoti hai to voh “element” variable me save nahi hoga and action perform nahi hoga

Second think steps 2 ki vajah se voh 20 sec web page par “Web element” ko search karega agar tab bhi nahi milta hai to fir voh exception throw kar dega

Disadvantage: agar web page par kafi sare dyanmic element hai to hume sabke liye separete code likhna hoga but is disadvanage to java ke code ki help se solve kar skte hai

### 8.1.3 fluent Wait in Selenium

* Fluent wait ek class hai jo selenium ke “org.openqa.selenium.support.ui” package ka hissa hai
* Yeh explicit wait jaisa hi hai
* Exception ko dur karne ke liye hum iska use karte hai
* Video **no 81 and 82**kafi achhe se explain kiya hai and iska use na ke barabar hota hai but interview me kafi jada puchha jata hai
* 99.99% webdriver wait ka hi use kiya jata hai for explicit wait ke liye

# 9 What are Assertions inTestNG?

**Assertions in TestNG are a way to verify that the expected result and the actual result matched or not.**Humare project me testNG ko involve karne ke liye hum “POM” page me “TestNG” ka 4 line ka code likhege tabhi hum testNG ko use kar payege

**Syntax for TestNG**

***Assert.Method(actual, expected)***

The parameter as you see contains three values:

* **Actual**: The actual value that the tester gets like if the tester's assertion is on the title of the page then what was the actual title of the page goes here.
* **Expected**: The value that you expect like if the tester's assertion is on the title of the page then what value of title do you expect goes here.
* **Message**: A string message to display only in case of an error when the assert fails.

# 10 Automate browser navigations and window

* Ek link se dusare link par jane ka code

driver.navigate().to(“URL”)

* Back page par jane ke liye

driver.navigate().back();

* forword page par jane ke liye

driver.navigate().forword();

* page ko refresh karne ke liye

driver.navigate().refresh();

# 11 Static dropdown

* Agar koi bhi static web element dropdown vala hota hai to uska Tag = **“Select”** rahta hai
* Selenium can handle static dropdowns with the help of the **Select class**. A dropdown is identified with **select tagname** and its options are represented with the **tagname “option”**.
* The 'Select' class in Selenium WebDriver is used for selecting and deselecting option in a dropdown. The objects of Select type can be initialized by passing the **dropdown webElement as parameter to its constructor**.

1. WebElement testDropDown = driver.findElement(By.id("testingDropdown"));
2. Select dropdown = **new** Select(testDropDown);

Note : in do line ka kafi jada important hai sabse pahle hume select class ko import karvana hoga. And jo web element dropdown vala hota hai uska locator ko hume as likhna hota hai jai ki yeh likha hai By.id("testingDropdown").

**Static dropdown ko hum hundle karne ke methods**

**1. selectByIndex** - It is used to select an option based on its index, beginning with 0.

EX: dropdown.selectByIndex(5);

**2. selectByValue** - It is used to select an option based on its **'value'** attribute.

**Options ka inspect karoge ko hume “Value” attribute jarur milega**

EX: dropdown.selectByValue("Database");

**3. selectByVisibleText** - It is used to select an option based on the text over the option.

**Jo option select karna ho to us option ka name likh do bus**

dropdown.selectByVisibleText("Database Testing");

# 12 Dynamic dropdown

* This type of drop down as in the above website are termed [dynamic drop downs](https://stackoverflow.com/questions/51222724/how-to-select-an-option-from-a-dynamic-dropdown-using-selenium) because it’s only after a city selection has been made from the ‘From’ drop down list are the city options loaded in the ‘To’ drop down list
* Without selecting a city in the ‘From’ drop down list try to select one from the ‘To’ list, you will find it’s empty
* Kafi simple hota hai jaise baki sabhi element ka locator find karte hai vaise hi isme bhi karege

# 13 Auto suggestive dropdown

* Yaha hum field me staring ka koi words likhege then uske bases par hume suggestion milege un suggestion ko ek list me daal dege fir list mese ek ek element ko nikal kar jo word chahiye hai uske sath compare karege fir mil jane par usse click kar ge
* Note : yaha hum ek generic locator find karege jo sabhi web element ko darshaye.

# 14 handling the check boxes

* Web page par kitne check box hai usse count karne ke liye sabhi check box ke locator me kuchh common hota hai usse use karke hume yeh pata chal jayega ki page par total kitne check boxes present hai
* Koi check box tick kiya gya hai ya nahi uska bhi method hota hai “isSelected” agar tick hai to true value dega agar nahi hai to false value dega

# 15 Importance of the Assertions

* Assertions ke liye hume testNG fremwork ki requirment hoti hai
* Ek to hum mevon reposatory se 4 line ka code “POM” page par likh dege to humare project ho testNG ka pata chal jayega
* Second method memon reposatorary me test ng likho fir usme jo new version hai uss par click karo and vaha jar hoga usse apne system me download kar do
* Now eclipse me apne project par right click karo 🡪click on Property🡪> click on the Java Build path🡪Libraries🡪Add external jars->upload and 🡪 ok
* Now humare project ko TestNG ka knowledge ho gya hai
* Assertion ka use actural result or expexcted result ko compare karne ke liye kiya jata hai
* Method for assertions 1) Assert.assertTrue(XYZ) yaha hum selenium ka voh code likhege jo hume true or false ke bareme batata ho example : isSelected method
* Agar Assert.assertTrue(True) hua to program aage run hoga otherwise aage nahi badhega
* 2) Assert.assertFalse(YZ) and YZ ki value true aa gya to program ruk jayega and agar false aagya to voh chalega
* Agar project me testNG jar upload ho gya hoga to hume 🡪 import org.testng.Assert; dikhega
* 3) Assert.assertEquals(Actual ,Expected);
* “Actual” ke palce me hum code ki line likhege and “expected” ke palce me hum kya chahte hai voh likhege
* Agar actual and expected dono same nahi hua to program fail ho jayega
* Uper ke Assert method ko hum hard assert bolte hai because agar test fail ho jata hai to voh code vahi ruk jata hai so iss issue ko solve krne ke liye hum soft Assert ka use karte hai

SoftAssert soft = **new** SoftAssert();

soft.assertTrue("Welcome".equals("welcomeeee"), "Third soft assert failed");

softAssert.assertAll();

* Yaha humara code continue chalta rahta hai bich me rukega nahi

# 16 Current date calender

* Current Date ko select karna kafi simple hota hai uske locator me koi ek aisa attribute hota hai jiske help se hum essly identifie kar payege ki voh select ho gya hai ya nahi
* Koi field enable hai ya disable hai uss ko janne ke liye isEnabled() method hai jo hume boolean value deta hai
* But aaj ke time me jo web page bane hai uss web page par yeh method kabhi kabhi fail ho jata hai
* But hum dekh skte hau ki enable karne ya disable karne par locatoe me kisi ek attribute ki value change ho rahi hai to hum java ka code likh ke enable ya disable ke bare me bata skte hai
* **Refrence ke liye video no. 62**
* Kisi bhi web element ke locator me kisi attribute ki value find karna ho to hum getAttribute(“class”) jisse hume class attribute ki value mil jayegi
* **System.out.println(driver.findElement(By.id("Div1")).getAttribute("style"));**

**if(driver.findElement(By.id("Div1")).getAttribute("style").contains("1"))**

**{**

**System.out.println("its enabled");**

**Assert.assertTrue(true);**

**}**

**else**

**{**

**Assert.assertTrue(false);**

**}**

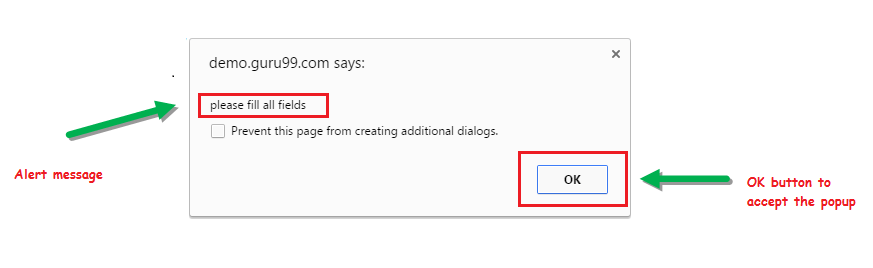
## 17 What is Alert in Selenium?

An **Alert in Selenium** is a small message box which appears on screen to give the user some information or notification. It notifies the user with some specific information or error, asks for permission to perform certain tasks and it also provides warning messages as well.

Here are few alert in Selenium types:

**1) Simple Alert**

The simple alert class in Selenium displays some information or warning on the screen.



Note : Alert message ek alag window hota hai jiska hum ispact karke locator find nahi kar skte and and voh javascript me likha hua hota hai

* + - void dismiss() **// To click on the ‘Cancel’ button of the alert.**

driver.switchTo().alert().dismiss();

* + - void accept() **// To click on the ‘OK’ button of the alert.**

driver.switchTo().alert().accept();

* + - String getText**() // To capture the alert message.**

driver.switchTo().alert().getText();

* + - void sendKeys(String stringToSend) **// To send some data to alert box.**

driver.switchTo().alert().sendKeys("Text");

* + - **Note:** code likhne par selenium check karta hai ki koi alert window “Window” par open hai ya nahi agar open hai to fir uske upper reqiurement ke hisab se action apply karta hai

# 17 Mouse hover action {Action class}

* + - A mouse hover is also called as hover. Mouse hover action is basically an action where a user places a mouse over a designated area like a hyperlink. It can cause some event to get triggered.
    - mouse cursor needs to be placed over an element. As mentioned in the [***Actions class***](https://www.toolsqa.com/selenium-webdriver/actions-class-in-selenium/) tutorial, “Actions class” provides a rich set of APIs for ***mouse events and keyboard events.***
    - We can perform mouseover action on elements in Selenium with the help of Actions class. In order to perform the mouse movement we will use moveToElement () method of the Actions class. Finally use build().perform() to execute all the steps.
    - Actions action = new Actions(driver);

action.moveToElement(“xpath of the element”).build().perform();

* + - Mouse hover action me hum **build().perform();** likhte hi hai agar nahi likha to execute nahi hoga code

# 18 Keyboard action{Action class}

* + - Agar mujhe kahi bhi capital letter me likhna hota hai to hum “shift" press karke keyboard me type karte hai to voh capital letter me likhata hai
    - Vahi concept ko selenium me “Action” class ki help se perform karte hai
    - We can enter a letter in upper case in the edit box in Selenium with the help of Actions. In order to achieve this, we need to first move to the edit box, then click() on that field. Then press SHIFT and enter the letters using sendKeys() method. Finally use build().perform() to execute all the steps.

Actions a = new Actions(driver);

a.moveToElement(driver.findElement(By.xpath(“input[@type=’text’]))).click().keyDown(Keys.SHIFT).sendKeys(“tutorialspoint”).build().perform();

# 19 Methods of actions class

* Action class is useful mainly for mouse and keyboard actions. In order to perform such actions, Selenium provides various methods.
* **Mouse Actions in Selenium:**
* **doubleClick()**: Performs double click on the element likhe hue ko select karne ke liye
* **clickAndHold()**: Performs long click on the mouse without releasing it
* **dragAndDrop()**: Drags the element from one point and drops to another
* **moveToElement()**: Shifts the mouse pointer to the center of the element
* **contextClick()**: Performs right-click on the mouse,kisi web element ke usper double click karne ke liye
* **Note**: yaha hum last me build().perform(); likhte hai
* **Keyboard Actions in Selenium:**
* **sendKeys()**: Sends a series of keys to the element
* **keyUp()**: Performs key release
* **keyDown()**: Performs keypress without release

# 20 Windows handle

* multiple windows open within an application when a button is clicked and the user has to perform some action on all the opened windows. Now user might not be able to work on all windows at the same time and hence need some mechanism through which he can take control over parent and child windows
* **window in any browser is the main webpage on which the user is landed after hitting a link/URL.** Such a window in [**Selenium**](https://www.toolsqa.com/selenium-webdriver/selenium-testing/) is referred to as the **parent window** also known as the **main window**
* All the windows which will open inside your main window will be termed as **child windows.**
* A child window may or may not have any URL.
* **A window handle stores the unique address of the browser windows.** It is just a pointer to a window, whose return type is **alphanumeric.** The window handle in Selenium helps in handling multiple windows and child windows. Each browser will have a unique window handle value with which we can uniquely identify it.
* **Selenium me new window and new tap dono ka meaning same hi hota hai**
* **Hur window ka apna ek unique id hota hai jo dusre window se kafi alag hota hai jo alphanumeric patteren me likha hota hai**
* Each window here will have a unique *ID* that we can get using the methods provided by *Selenium Webdriver* and then use the same to switch the context to that specified window.
* Yaha hum kafi sare methods use karege for window handles
* **getWindowHandle( ):** When a website opens, we need to handle the main window i.e the parent window using driver.getWindowHandle( ); method. With this method, we get a unique ID of the current window which will identify it within this driver instance. This method will return the value of the [**String type.**](https://www.toolsqa.com/java/string-class/)
* **getWindowHandles( ):** To handle all opened windows which are the **child windows** by web driver, we use driver.getWindowHandles( ); method. The windows store in a **Set** of String type and here we can see the transition from one window to another window in a web application. Its return type is [**Set**](https://www.toolsqa.com/java/list-interface/#:~:text=Method%3A%20iterator()&text=For%20the%20reader%20unfamiliar%20with,point%20to%20the%20next%20page.&text=%2F%2F%20Get%20a%20list%20iterator%20over%20the%20elements%20in%20the%20list.,-ListIterator%20listIterator) **<String>.**
* **switchto():** Using this method we perform switch operation within windows.
* **action:** This method helps in performing certain actions on the windows.
* String win= driver.getWindowHandle();

System.out.println(win);

Output : alphanumeric number dikhega jo hur window ka unique hota hai and hur baar run karne par iski value change hoti rahti hai

* Child windows ka id find karne ke liye **set** ka use kiya jata hai following syntex

Set(String) childwindows =driver.getWindowHandles();

List<String>childwin = new ArrayList(childwindows);

String win1 =childwin.get(0);

String win2 =childwin.get(1);

System.out.println(win1);

System.out.println(win2);

* Child window par jane ke liye
* driver.switchTo().window(win1);
* driver.switchTo().window(win2);

# 21 scolling the window

* Hum selenium ki help se window ko scolling nahi kar skte hai. So hum yaha java script ka use karege

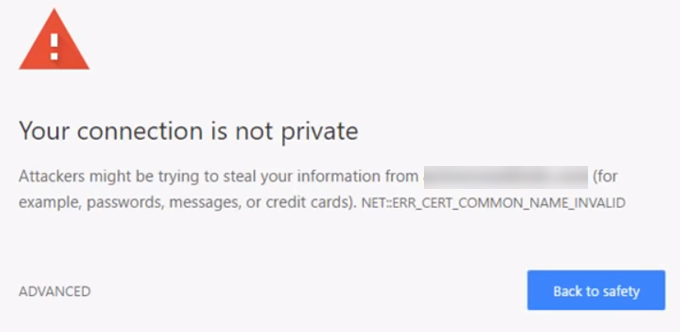
JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,250)", "");

* Vaise hi window ke ander agar koi table hai to usko bhi hum scroll kar skte hai by using javascript
* For practice video no 100

# 22 SSLcertification in selenium

* Below image is ssl certificate
* **import** org.openqa.selenium.chrome.ChromeOptions;



* Jab koi webside safe nahi hoti hai to iss type ka error dikhta hai
* To isse hundle karne ke liye humare pass ek class hai jiski help se hum iss type ke issue ko solve kar skte hai
* Normally jab hum code likhte hai to ussme humare chrome ka behaviour hum set nahi karte hai but agar humne apne chrome ka behaviour fix kar diya to iss type ka issue nahi aayega

ChromeOptions options = **new** ChromeOptions();

options.setAcceptInsecureCerts(true);

WebDriver driver = new ChromeDriver(options)

* Agar “true” ke place par “false” likh dege to voh kabhi bhi isse handle nahi karega and above image ko window par display kar dega. So hume “true” value likhna hota hai

**Note:** “ChromeOptions” ek class ka name hai to hume iss type ke issue ko handle karne me help karta hai and chrome ke behaviour ko set karne me help karta hai

# 23 Extentions of plugins in selenium

* Jab hum apna normal chrome open karte hai to agar humne plugins download kiya hoga to hume top right corner par uska icon dikhega.
* But jab hum code ki help se chrome browser open karte hai to hume koi bhi plugins nahi dikhta so iss issue ko solve kaene ka code

ChromeOptions options = **new** ChromeOptions();

Options.addExtentions(“XYZ”);

WebDriver driver = new ChromeDriver(options)

Note: yaha ‘XYZ’ ke place par hum plugins file ka path(System me jaha se voh) likhege jiske help se humara code usse new chrome window par plugins ko add kar dega

# 24 Proxy server related issues

* Jub bhi hamare project ko proxy ki need hoti hai to hume uss problem ko solve karna hota hai
* Office me techinal support or support engineer hume proxy se baareme batate hai ki kis type ki proxy use karni hai like https vali or koi or to usske according hum apna code likhte hai

ChromeOptions options = **new** ChromeOptions();

Proxy proxy = **new** Proxy();

proxy.setHttpProxy("ipaddress:4444");-🡪 yaha ip adress and port no likha hua hai

options.setCapability("proxy", proxy);

WebDriver driver = new ChromeDriver(options)

# 25 Cookies and cache file delete

* Webside use karne se pahle agar sabhi cookies ya koi separate cookies ko delete karna ho to
* driver.manage().window().maximize();

driver.manage().deleteAllCookies();

driver.manage().deleteCookiesNamed(“XYZ”); 🡪 XYZ ke place par cookies name likho

# 26 Screenshort

* seleniunm 3 ki help se screenshort lege jo ki entire page ka lega
* import org.apache.commons.io.FileUtils
* yaha hum type casing ka use karege
* File src =((TakesScreenshort)driver).getScreenshortAs(OutputType.FILE);
* FileUtils.copyFile(src,new File(“file ka path jaha hume chahiye.png”));

**Note:** import file direct import nahi hota so hume khud se download karke usse import karna hota hai refer the video on youtube

# 27 Brokon Link

* Jab hum kisi link ko open karte hai to voh hume 404 ka error de to is type ke link ko hum brokon link kahte hai
* Agar hum inspact kare 🡪Network🡪XHR(button) click kare to link sahi hoga to hume 200 status batayega but problem hua to more than 400 batayega
* So yaha hum status code ki help se batayege ki code brokon hai ya nahi
* Steps 1 yaha pahle hum link ko locator find karege

String aa = driver.findelement(By.cssSelector()).getAttribute(“href”);

* Ab hum ‘URL’ class and ‘openconnection’ method ki help se check karege and yaha hum type casing karege

HttpURLConnection Aae = (HttpURLConnection) new URL(aa).openConnection();

Aae.SetRequestMethod(“HEAD”);

Aae.connect();

Int QW=Aae.getResponseCode();

System.out.println(QW);

* Aagar QW ki value 400 se jada hai to voh link brokon hoga

**Note:** 111 and 112 ka video dekho for practice

# 28 Special feature for selenium 4 version

**Note:** Selenium 4 version ke feature ko project me use karne ke liye hume ek package import karna hota hai jo eclipse auto suggestion nahi deta hai because yeh ek static package hai and static package ko eclipse kabhi bhi auto suggestion nahi karta.

**import** **static** org.openqa.selenium.support.locators.RelativeLocator.\*;

**Note:** selenium 4 use karne se pahle pom file me selenium ka 4 version ka jar add hai ya nahi voh check karo

## Screenshot

* Selenium 4 se pahle hum pure entire page ka hi screenshot le skte the but selenium 4 me hum jis web element ka chahte hai uska screenshort le skte hai
* WebElement logo=driver.Findelement (By.xpath(“//div[@id=’divLogo’]//img”));

File file=logo.getScreenshotAs(OutputType.FILE);

File destFile =**new** File(“logo.png”);

FileUtils.copyFile(file,destfile);

OR

FileUtils.copyFile(file,newFile(“abc.png”));

**Note:** yaha eclipse me humare project ke ander “abc.png” name ka ek folder create ho jayega jiske under ke path ki help se hum pysical uss image ko dekh skte hai

**Note:** ‘FileUtils’ ko apne project me use karne ke liye hume isse import karna hoga. So use karne se pahle kaise import karte hai please youtube par dekh lo

## Open the new tab on the browser:

* If the user wants to open 2 URLs in two different tabs at the same time, the user can do that with the Selenium 4.
* Agar do link humare pass hai to hum do tab open kar skte hai.
* **Please find the below code for reference:**

|  |
| --- |
| driver.get(https://www.google.com/);  driver.switchTo().newWindow(WindowType.TAB);  Is syntex se humara second tap to open ho jayega but uss tab par humara driver object automatically nahi jayega jiske vajah se hum vaha ke webelement ko identify nahi kar payege to iske liye hume ek link ki need hogi agar knowledge hai to koi issue nahi hai but agar nahi hai to hume new tab ke **unique id** ki requirment hogi.  Set<String> Handles = driver.getWindowHandles();  Iterator<String> it = Handles.iterator();  String parentwindow =it.next();  String child1 =it.next();  Continue aise hi likhte jao jitne bhi child tab hai  driver.switchTo().window(child1);  driver.navigate().to(<https://www.crmpro.com/>);  **Note:** ab humara **driver object** child1 tab par hai |

## Open the new window on the browser:

* If the user wants to access two applications in the same browser, the user can now do this.
* **Please find the below code for reference:**

|  |
| --- |
| driver.get(https://www.google.com/);  driver.switchTo().newWindow(WindowType.WINDOW);  driver.navigate().to(https://www.crmpro.com/); |

## Object Location:

* Users can achieve the coordinates, dimension, height, width, etc. as the location of the web elements or object.
* WebElement el = driver.findElement(By.id("regform"));

Rectangle rect = el.getRect();

System.out.println("X-Axis - " + rect.x ); //from left top corner of element

System.out.println("Y-Axis - " + rect.y );

System.out.println("Element Width - " + rect.width );

System.out.println("Element Height - " + rect.height );

## Relative Locators:

* This functionality is being added to find out the element which is present nearby to other web element or, we can say that it can find the web elements based on GUI location.

**There are five locators added in Selenium 4:**

* **below():** Web element located below for the specified element.
* **toLeftOf() :** Target web element which is present to the left of specified element.
* **toRightOf():** Target web element which is presented to the right of a specified element.
* **above():** Web element located above for the specified element.
* **near() :** Target web element which is away(approx. 50 pixels) from the specified element.

**Note:** All the above relative locators method support “**withTagName**” method only.

**EX:** WebElement password = driver.findElement(By.id("password"))

WebElement email = driver.findElement(with(**By.tagName("input"**)).**above**(password));

# 29 Poject of rahul shetty or his instraction

* Note : video number 42 kafi important hai for the practical perpose
* **“Protractor** is an end to end testing framework specially designed to perform automation testing on**AngularJS based web applications**.”
* “Rahul shetty” sir ne iss course ko karne ko bola hai
* Code ko proper allignment me likhna chahiye agar likhne ke baad usse proper way me arrange karna ho to hume **“Ctrl+Shift+F”**
* **Agar ‘text’ attribute ki value dynamic hai to hume “text” ko as locator find karne ke liye use nahi karna chahiye**
* **14 section dekhna baki hai**
* **Section 15 13 12 ka overview ho chuka hai**
* **Video 92 to 94 kafi importance hai kaise hum driver ko pure page se kisi chote section tak simit kar skte hai**