Selenium Information-Selenium dev ,Selenium Version-Selenium4.15

What is Selenium

* Selenium is suite of tool used to automate web applications(web based automation tool)
* Selenium supports different programming languages such as Java,C#,Python etc
* Selenium supports different browsers such as Chrome,Firefox,IE,Edge,Safari
* Selenium supports multiple platforms such as Windows,MAC,Linus
* Selenium can be integrated with automation tool such as maven,jenkin,docker
* Selenium supports framework such as TestNG & Junit

|  |  |
| --- | --- |
| Manual Testing | Automation Testing |
| Takes lot of time for manually executing the testcases | Time can be saved |
| Cost Increases | Less cost |
| Report Generation is tedious | Auto generation of reports |
| Less Accuracy | Increase accuracy |

Advantage of Selenium

* Selenium is not commercial its an open source software
* Selenium supports multiple platforms such as Windows,MAC,Linus
* Selenium supports different programming languages such as Java,C#,Python etc
* Selenium supports parallel testing
* Selenium supports framework such as TestNG & Junit

Disadvantage of Selenium

* Barcode and captcha reading cannot be done with selenium
* Mobile testing cannot be done
* No reporting facility
* Limited support for image testing
* It doesn’t provide any test tool for test management

What is framework

Framework is design pattern which has rules and industrial standards.

Page object model we say as an framework

Different components of Selenium

1.Selenium IDE

2.Selenium Grid

3.Selenium WebDriver

4.Selenium RC

Selenium 1-IDE,RC,GRID

Selenium 2-RC,GRID,WEBDRIVER

**Selenium IDE**

* It supports only firefox plugin
* Using IDE we can create,edit,execute testcases
* IDE is used for recording and playing back the user interaction with the browser
* We can export programs written in one language with other language

Advantage

* Create and edit testcases
* Debug the testcases
* Create and edit the Testsuite
* Export in different programming languages
* Non technical people can understand

Disadvantage

* Only supports dynamic web application
* Supports only firefox plugin
* Programming logic is not needed
* Data driven testing is not possible
* Limitation in alerts,file upload,filedownload

**Selenium RC**

* We can write testcase with proper logic
* Whatever code we have written RC checks the code and then pass it to the browser
* Rc server is slow so its time consuming
* Selenium RC has 2 components

1.Selenium RC Server which as an http request-it will pass the command

2.Selenium RC Client-Jarfiles communication-it will pass the command

**Selenium WebDriver**

* WebDriver is an interface
* Webdriver is faster than Selenium RC
* Selenium webdriver supports different browser driver such as ChromeDriver ,FirefoxDriver,IEDriver,EdgeDriver,SafariDriver.
* Selenium supports different programming languages such as Java,C#,Python etc

Drawbacks

* Report generation is not possible
* Object cannot be centeralized

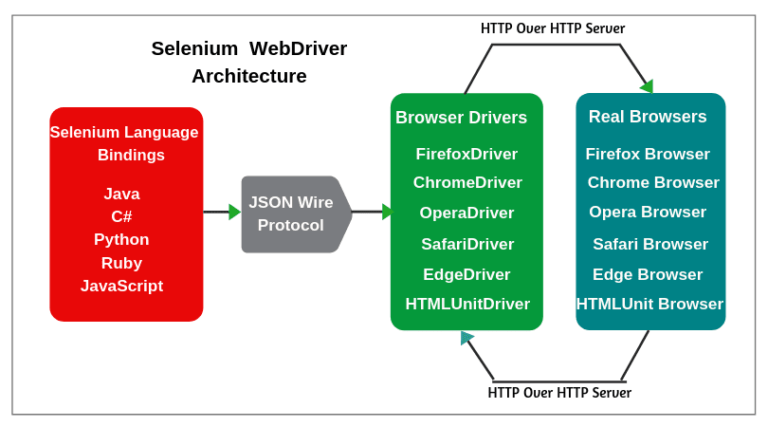
**Selenium Grid**

* Using grid we will be able to run our testcases in different browsers using single machine that we say as parallel execution(cross browser testing)
* Selenium grid need Seleium RC,Webdriver to write testcases

NOTE

In selenium Ide we need not need an programming logic else in all other components we use logic

**Selenium Architecture**



* User can write the code using any programming language
* JSON wire protocol is an communication interface where it reads all the code(might be browser launch,opening the url etc…)
* It will download the broser driverband it will launch and open the browser in real broser with help of http server
* IN selenium 4 Instead of JSON Wire protocol w3c protocol is replaced.
* Here the protocolneed not download the driver ,the driver here directly get created with help of w3c all the information in code will go to w3c open the browser driver and execution happens with http server.

Selenium Commands

1.WebDriver Commands or Browser Commands

2.Navigation commands

3.WebElement Commands

**WebDriver Commands**

|  |  |  |
| --- | --- | --- |
| Commands | Usage | Examples |
| driver.getTitle(); | To get the title of the application | driver.get("url”); |
| driver.getTitle(); | To get the current page title(tab title)  Return type :String | String pagetitle= driver.getTitle();  System.***out***.println(pagetitle); |
| driver.getCurrentUrl(); | To get the current URL of the active web page  Return type :String | String url=driver.getCurrentUrl();  System.***out***.println(url); |
| driver.getWindowHandle(); | Will get unique id of current active webpages  Return type :String | String handleid=driver.getWindowHandle();  System.***out***.println(handleid); |
| driver.getWindowHandles(); | Will get unique id of all active webpages(alphanumeric)  Return type :Set<String> |  |
| driver.getPageSource(); | To get html code of the current webpage | String page=driver.getPageSource();  System.***out***.println(page); |
| driver.close(); | To close the current window | driver.close(); |

**Navigation Commands**

|  |  |  |
| --- | --- | --- |
| Commands | Usage | Examples |
| driver.navigate().back(); | To go back to the previous screen from the current scree(tab) | **package** org.selenium.basics;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** NavigationCommands {  **public** **static** **void** main(String[] args) {  WebDriver driver=**new** ChromeDriver();  driver.get("https://www.seleniumeasy.com/");  driver.manage().window().maximize();  driver.navigate().back(); |
| driver.navigate().forward(); | To move to forward screen from back screen | **package** org.selenium.basics;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** NavigationCommands {  **public** **static** **void** main(String[] args) {  WebDriver driver=**new** ChromeDriver();  driver.get("https://www.seleniumeasy.com/");  driver.manage().window().maximize();  driver.navigate().back();  driver.navigate().forward(); |
| driver.navigate().refresh(); | To refresh the current screen | **package** org.selenium.basics;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** NavigationCommands {  **public** **static** **void** main(String[] args) {  WebDriver driver=**new** ChromeDriver();  driver.get("https://www.seleniumeasy.com/");  driver.manage().window().maximize();  driver.navigate().back();  driver.navigate().forward();  driver.navigate().refresh(); |
| driver.navigate().to(“url”); | To navigate to the url | **package** org.selenium.basics;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** NavigationCommands {  **public** **static** **void** main(String[] args) {  WebDriver driver=**new** ChromeDriver();  driver.get("https://www.seleniumeasy.com/");  driver.manage().window().maximize();  driver.navigate().back();  driver.navigate().forward();  driver.navigate().refresh();  driver.navigate().to("https://www.selenium.dev/"); |

**WebElement**

* WebElement is an interface which represent an html code
* With help of webeleemnt we can perform actions
* To find webelement we use findElement() and findElements()
* WebElement is present inside start and end tag

findElement() method provides an single element its return type is webelement

findElements() method provide list of elements its return type is List

**Locators**

Id(string args)

name(string args)

linkText(string args)

tagName((string args)

partiallinkText((string args)

className((string args)

xpath((string args)

cssSelector(string args)

NOTE:

ID is the always unique and preferable

Xpath OR cssSelector is 2nd most preferable after id

**Xpath**

There are two types of xpath which includes Absolute xpath and Relative xpath

**Absolute xpath**

It starts from the root node

Right click on the slected element copy full xpath

/html/body/div[4]/div[1]/div[1]/div[2]/div[1]/ul/li[2]/a

**Relative xpath**

We can fetch xpath from mid of the DOM(CUSTOMIZE XPATH)

**WebElements method**

|  |  |  |
| --- | --- | --- |
| Commands | Usage | Examples |
| Click(); | Used to click an method | WebElement loginbtn=driver.findElement(By.*cssSelector*("body > div.master-wrapper-page > div.master-wrapper-content > div.master-wrapper-main > div.center-2 > div > div.page-body > div.customer-blocks > div.returning-wrapper > div.form-fields > form > div.buttons > input  loginbtn.click(); |
| sendKeys(); | Used to provide an text | WebElement password=driver.findElement(By.*id*("password"));  password.sendKeys("secret\_sauce"); |
| getText(); | To retrieve the text within the specific element return type is string | Driver.getText(); |
| isSelected(); | To check whether the radio button,checkbox is selected ornot return type is boolean |  |
| isEnabled() | To check whether element is enabled or not return type is boolean |  |
| isDisplayed() | To check element is checked or not  Return type is boolean |  |
| driver.close(); | Used to close the current window |  |
| driver.quit(); | Use to close all the window |  |

Relative xpath

|  |  |  |
| --- | --- | --- |
| Commands | Syntax | Examples |
| using tagname,attribute and value | //tagname[@attribute='value']; | reg-//a[@class='ico-register'];  login-//a[@class='ico-login'] |
| using contains | //tagname[contains(@attribute,'value')] | //a[contains(@class,'ico-login')] |
| using text-if there is no class id u can go with text | //tagname[text()='text'] | //a[text()='Register'] |
| Using and | //tagname[@attribute1='value'and @attribute2='value']; | //input[@name='Gender' and @id='gender-male'] |
| using OR | //tagname[@attribute1='value' or @attribute2='value'] | //input[@id='gender-male' or @for='gender-male'] |
| using starts with | //tagname[starts-with(@attribute,'value')] | //input[starts-with(@name,'FirstName')] |
| from parent to child | //parenttagname[@attribute='value']//childtagname[@attribute='value'] | //div[@class='gender']//input[@id='gender-male'] |
| using sibling-gender demowhebshop next immediate sibling | //parenttagname[@attribute='value']//childtagname[@attribute='value']//following-sibling::siblingtagname | //div[@class='gender']//input[@id='gender-male']//following-sibling::label |
| Using parent | //childtagname[@attribute='value']//parent::parenttagname | //input[@id='gender-male']//parent::divparent::div |

Absolute xpath

/html/body/div[4]/div[1]/div[1]/div[2]/div[1]/ul/li[2]/a

Absolute css

body > div.master-wrapper-page > div.master-wrapper-content > div.header > div.header-links-wrapper > div.header-links > ul > li:nth-child(2) > a

xpath

1.using tagname,attribute and value

syntax

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

Example: demowebshop using this create 5 ex

reg-//a[@class='ico-register'];

login-//a[@class='ico-login']

2.using contains

syntax

//tagname[contains(@attribute,'value')]

ex use 5ex

//a[contains(@class,'ico-login')]

3.using text-if there is no class id u can go with text

Syntax use 5 ex

//tagname[text()='text']

//a[text()='Register']

4.Using and

syntax

//tagname[@attribute1='value'and @attribute2='value'];

//input[@name='Gender' and @id='gender-male']

5.using OR

syntax

//tagname[@attribute1='value' or @attribute2='value']

//input[@id='gender-male' or @for='gender-male']

6.using starts with

syntax

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

//input[starts-with(@name,'FirstName')]

7.from parent to child

syntax

//parenttagname[@attribute='value']//childtagname[@attribute='value']

//div[@class='gender']//input[@id='gender-male']

8.using sibling-gender demowhebshop next immediate sibling

syntax

//parenttagname[@attribute='value']//childtagname[@attribute='value']//following-sibling::siblingtagname

//div[@class='gender']//input[@id='gender-male']//following-sibling::label

9.Using parent

syntax

//childtagname[@attribute='value']//parent::parenttagname

//input[@id='gender-male']//parent::div

**cssSelector**

|  |  |  |
| --- | --- | --- |
| **cssSelector** | **Syntax** | **Example** |
| Using tagname,attribute and value | Tagname[attribute=’value’] | Input[id=’Email]; |
| Using tagname and class | tagName.class  or  .class | Input.email  Or  .email |
| Using tagname and id | tagName#id  or  #id | Input#newsletter-subscribe-button  Or  #newsletter-subscribe-button |