Selenium WebDriver

Contents:

1) What is Automation Testing?

Automation Testing is in trending now a days and selenium is one of the popular automation tool which we come across.

Automation Testing is very important because when the development process is completed and the build is moved from development environment to the testing environment, we do not need to test the entire module present in the application. We only need to test the new features and bug fixes which are added in the current build and we can automate the remaining module. This saves lot of time. Therefore, automation testing comes into the picture.

2) What is Selenium?

Selenium is an open source automation tool which is used to automate web based application across multiple browser and operating system.

We also have specific automation testing tools for web based application, mobile based application and desktop based application.

Web Based Application : Playwright (Open Source), Selenium (Open Source), Appium (Open Source), Cypress (Paid), Katalon Studio (Paid) and Browser Stack (Paid) etc…

Mobile Based Application : Appium (Open Source), Browser Stack (Paid), Katalon (Paid) and Testim (Paid) etc…

Desktop Based Application : Quick Test Professional (Paid), Test Sigma (Paid) and Ranorex (Paid) etc…

3) What are the components of Selenium?

Selenium is an automation testing tool which is used to automate web based application.

Selenium is divided into 3 components. They are 1) Selenium IDE 2) Selenium WebDriver 3) Selenium Grid

1) Selenium Integrated Development Environment (IDE) :

Selenium IDE works on record and playback principle in order to create and automate test scripts across

2) Selenium WebDriver :

Selenium WebDriver is an open source automation library which is used to automate test scripts across multiple browser (Google Chrome, Mozilla Firefox and Microsoft Edge) and operating system (Windows, Mac and Linux)

3) Selenium Grid :

Selenium Grid automate test scripts parallely across multiple browsers and operating systems. It is works on the basis of hub server and nodes concept.

Firstly we have to add downloaded selenium standalone jar.

Then we(user) have to connect to the hub server. After that we have to connect to the nodes. Each node can have multiple operating system and we have to connect node to the hub server. Then the test scripts which we have created should connect to the hub server with the help of Remote WebDriver class, ChromeOptions class and selenium standalone jar. Thus, the test script helps to automate parallely across multiple browsers and multiple operating system.

Each component has its own advantages and disadvantages.

4) What is Selenium WebDriver?

Selenium WebDriver is an automation library which is used to automate web based application across multiple browsers (Google Chrome, Mozilla Firefox, Microsoft Edge) and multiple operating systems (Windows, Mac and Linux)

The main aim of selenium webdriver automation testing is to automate functional and regression test case based on the business requirements.

When we try to interact in browser, HTML DOM gets processed or loaded. Every webpage has webelements in document object model (DOM). Every document object model (DOM) has attributes and values. Thus, It also verifies attributes and values (webelement) in webpage are working fine during automation.

Selenium WebDriver library can also be integrated with third party API's like TestNG for parallel testing, assertions, set priority, group test case and Apache POI (library) to maintain test data using MS Excel and automate test case. We can write selenium webdriver test scripts using multiple programming languages like Java, C#, Javascript and Python

The disadvantage of selenium webdriver is that it cannot automate webservices testing, mobile testing and generate report.
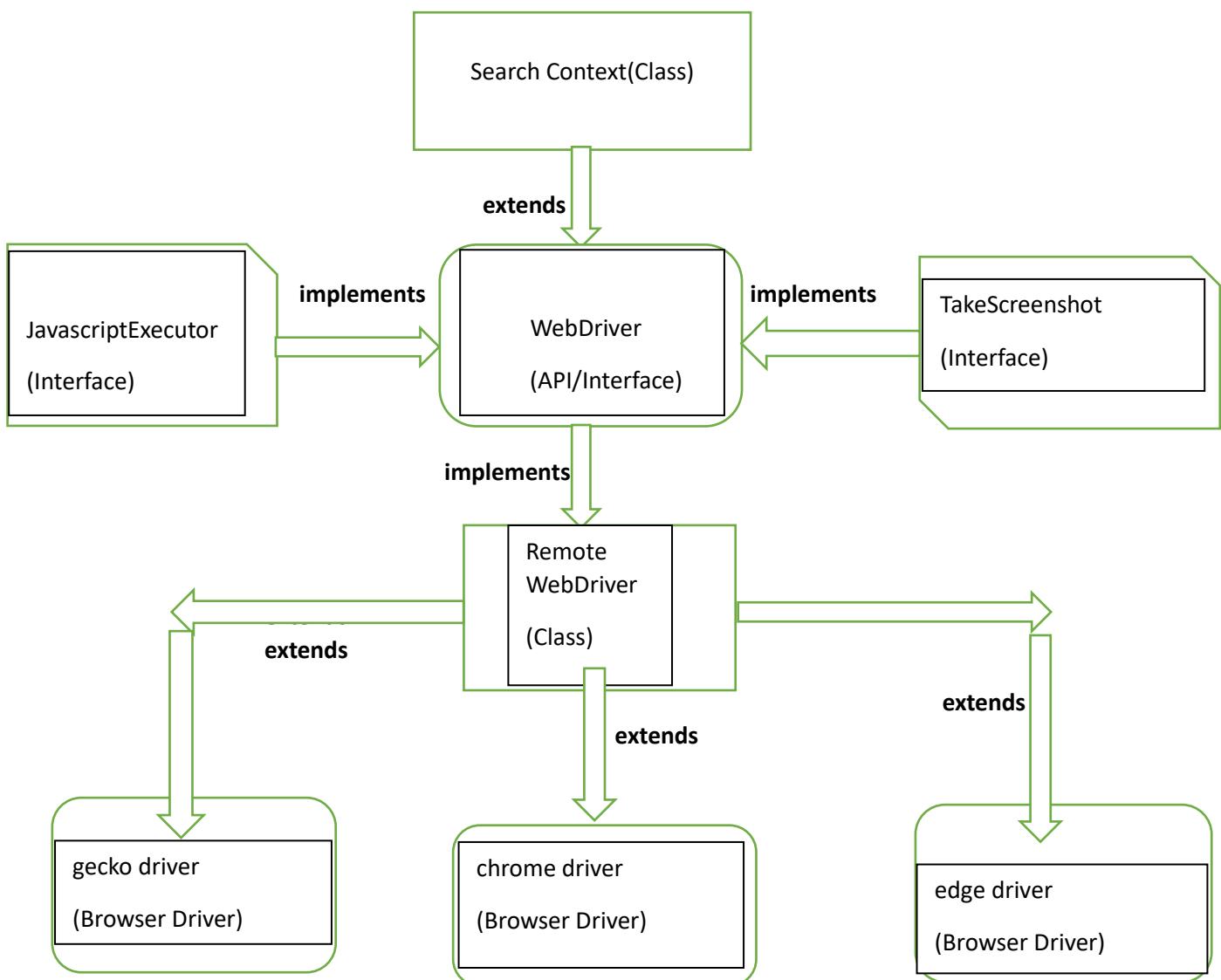
5) Why selenium webdriver is popular compared to other automation testing tools?

Selenium WebDriver is popular because

1) It saves time by maintaining existing test scripts.

2) Increases efficiency and productivity when we automate regression test case.

3) It supports multiple programming languages and operating systems.

4) It is used for validation which makes our test script and framework stable.

5) It supports framework. Thus, the test scripts are stable & robust and will be helpful for future purpose.

6) It helps for reporting where we can integrate with TestNG HTML report, Extent report and Allure report.

7) It can be integrated with continuous integration and continuous delivery tool like Jenkins which helps to automate test case in headless mode and create continuous build.

8) Selenium scripts can also automate cloud based Application.

6) What is the archirtecture of Selenium WebDriver?

Search Context(Class)

extends

JavaScriptExecutor
(Interface)

implements

WebDriver
(API/Interface)

implements

TakeScreenshot
(Interface)

implements

Remote
WebDriver
(Class)

extends

extends

extends

gecko driver
(Browser Driver)

chrome driver
(Browser Driver)

edge driver
(Browser Driver)

Search Context is the super class. It helps to store webelement.

Search Context extends WebDriver. It consists of findElement and findElements method which helps to interact with webelement. WebDriver is an interface which consists of in build methods. These methods also helps to interact with the webelements in webpage.

Apart from this javascriptexecutor and takescreenshot are interface which implements webdriver. Normally when we use click method in webdriver to click on any webelement, It does not work and it throws element intercepted exception. Thus, we use JavascriptExecutor interface to resolve exception and to click on element, scroll up or down and to scroll till particular pixels in webpage.

We also have TakeScreenshot interface for taking screenshot. Selenium is upgraded with version 4 and here we can not only take screenshot for the particular webelement.
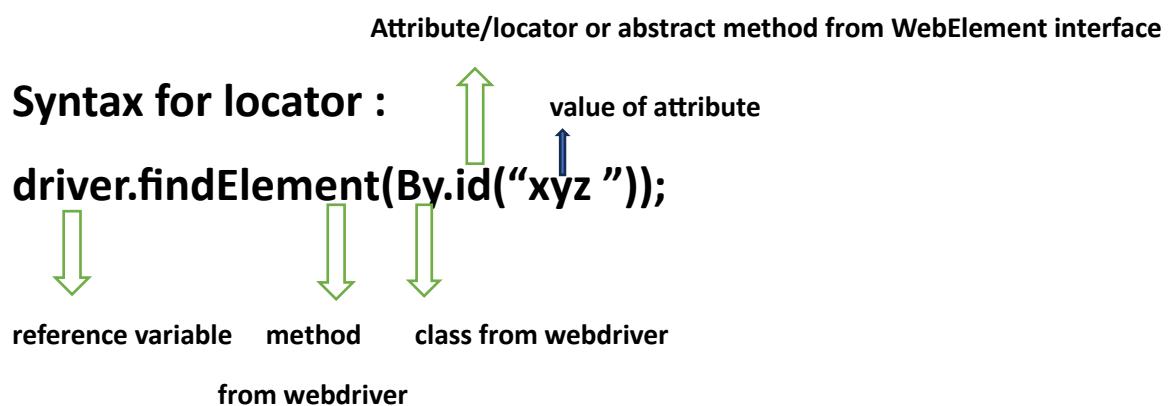
we can also take screenshot the entire section. WebDriver interface implements RemoteWebDriver class which helps to connect or interact with the browser. As it extends chromedriver for google chrome browser, gecko driver for mozilla firefox browser and edge driver for microsoft edge browser, We can automate our test case in browsers.

7) What are Locators?

Locators in selenium webdriver are address (of webelement) which helps to find or interact with the webelement in webpage. It also helps to write relative xpath.

There are 8 types of locators. They are

1) id : It is used when value is uniqu**e**

2) name : It is used when value is unique

3) classname : It is used when value is unique

4) tagName : It is used when we interact with tagName

5) link text : It helps to get the inner text present between the attributes in HTML DOM

6) partial link text : It helps to get partial inner text

7) Xpath : It helps to interact with dynamic values

8) css Selector : It helps to interact with complex webelement

Attribute/locator or abstract method from WebElement interface

Syntax for locator :

value of attribute

driver.findElement(By.id("xyz "));

reference variable   method   class from webdriver

from webdriver

8) What is Xpath?

Xpath is syntax which helps to find address of webelement in webpage (HTML DOM). We also have xpath functions which interacts with dynamic values (which constantly changes)

There are two types of xpath. They are
1)Absolute Xpath
2)Relative Xpath

1) Absolute Xpath : Absolute xpath syntax starts with / slash. It starts from the root node and ends till the particular webelement which we are trying to interact in HTML DOM.

2) Relative Xpath : Relative Xpath is always preferred. Syntax starts from //. It directly jumps to the particular webelement which we are trying to interact in HTML DOM.

Relative Xpath is always preferred because

1) We can write xpath axes like self, parent, child, following, preceeding, preceeding-sibling, following-sibling, ancestor and descendant to keep our xpath stable

2) Syntax is shorter and we can customize relative xpath.

3) It directly jumps to the particular webelement where we are trying to interact.

4) When a new changes are added in web application, the whole xpath may fail. Therefore, relative xpath are mostly used.

5) We can also write xpath functions when values are dynamic or constantly changing in webpage like text, and, or, contains, starts with and end with method

Synatx for relative xpath :

//tagname[@attribute='value']

9) What is Synchronization?

When we run our test script, test script execution is faster but page loads slowly and it throws NoSuchElementException. This is known as synchronization. This issue can be resolved with the help of implicit wait, explicit wait and fluent wait.

Explicit wait is mostly used for synchronization.

Synatx for implicit wait :

webdriver

reference

variable   predefined methods from webdriver   webdriver reference var

driver.manage().timeouts().implicitWait(driver,TimeUnit.seconds(10));

specified time which helps to interact with webelement

Syntax for explicit wait :

Class from selenium webdriver   **reference variable of webdriverwait class**

Keyword helps to create object

WebDriverWait   wait   =   new WebDriverWait(driver,Duration.seconds(10));

reference of webdriver interface expected

specified time waits for the mentioned conditions

method which will wait till the mentioned expected conditions

Wait.untill(ExpectedConditions.PresenceOfAllElementLocatedBy(XYZ));

Condition to set

consist of list of conditions where the reference variable of webdriver will wait for the particular webelement which we are trying to interact

10) What is framework?

Framework is a systematic way which we have to follow in order to get the expected result.
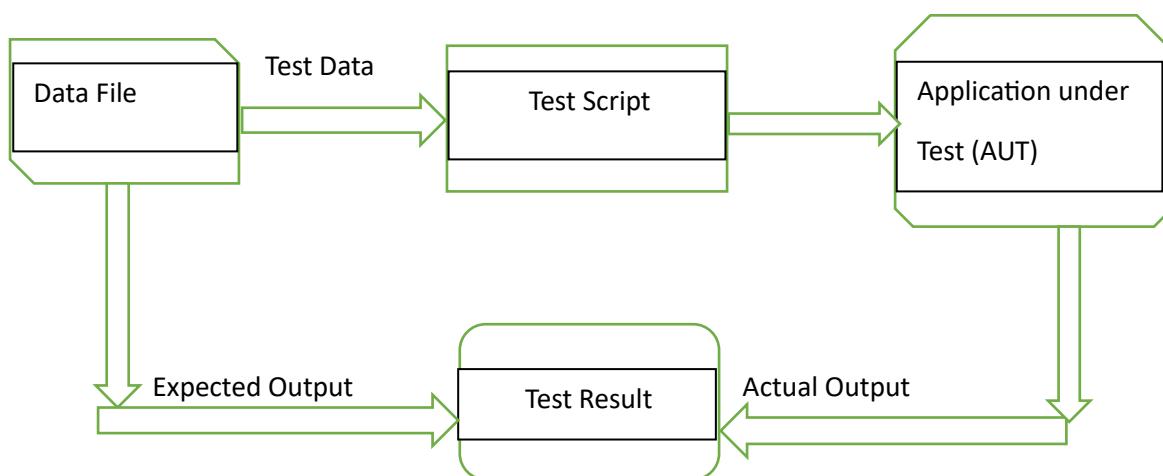
1) Keyword Driven Testing Framework
2) Data Driven Testing Framework
3) Page Object Model Framework
4) BDD Cucumber Framework
5) Hybrid Driven Testing Framework

Keyword Driven Testing Framework :

Keyword driven testing is the framework where action methods written in test script should match with the methods present in excel file. If the action methods are matched then the test script interacts with the browser using browser drivers and automate webelement present in webpage (HTML DOM)

Data Driven Testing Framework

Data driven testing is the framework where we have to maintain test data and expected output in excel file. We can also customize excel file with common action methods and maintaining xpath for particular webelement. Now test data should be integrate with the test script and test script interacts with the browser using browser driver and automate the particular webelement in webpage. If the actual result from the output is equal to the expected output present in excel file then the test case is passed.

```
┌──────────────┐   Test Data   ┌──────────────┐          ┌──────────────────┐
│  Data File   │ ─────────────▶│  Test Script │ ────────▶│ Application under │
└──────┬───────┘               └──────────────┘          │   Test (AUT)     │
       │                                                  └────────┬─────────┘
       │                                                           │
       │   Expected Output    ┌──────────────┐   Actual Output     │
       └─────────────────────▶│  Test Result │◀────────────────────┘
                              └──────────────┘
```

Page Object Model Framework :

Page Object Model is the framework where we separate action methods and test methods in one file. First we have to create constructor in order to connect main test case to the the page object file with the help of reference variable. Action method consists of object repository (set of webelements) and functions. Test method consists of main test where we have to call action method functions with the help of constructor name.

Page Object Model follow page factory approach where it consists of @FindBy annotation and PageFactory.initElements method. @FindBy annotation is used to find the webelement in webpage and PageFactory.initElements() is used to initialize the webelement. The main advantage of following page object model framework is that whenever a new element or new changes is added we only have to update xpath. We do not need to change the existing function.

BDD Cucumber Framework :

Behaviour Driven Development always focus on system behaviour and checks the behaviour of function. We can check the behaviour of function with the help of tool like cucumber.

Cucumber : Cucumber is one of the tool which comes under behaviour driven development. The main advantage of cucumber tool is that it follows plain english so that it can be easily understood by client. It follows gherkin language and keywords like Given, When, Then and Background.

Given : Given keyword is used to launch or setup

When : When keyword is used to write functionality or logic according to the business requirement.

Then : Then keyword is used to write the expected result.

Background : Background keyword is used to write the common steps.

We can also integrate page object model with BDD Cucumber which makes our framework robust.

Hybrid Driven Testing Framework :

Hybrid Driven Testing is the combination of more than one framework.

1st Approach : We can combine page object model with data driven testing framework.

2nd Approach : We can combine page object model with testNG framework.

3rd Approach : We can combine BDD cucumber with page object model framework.