Chapter:1

Introduction:

1.what is Automation?

2.Why Automation?

3. Advantages and dis advantages of Automation?

4. Types of Automation?

Introduction to Selenium:

1.What is Selenium?

2. Why Selenium?

3. Advantages and disadvantages of Selenium?

4. Selenium Components?

Selenium Softwares:

Selenium servers, creating maven project

Selenium WebDriver Architecture

WebDriver Architecture

Methods of WebDriver

HTML

Locators

Types of Locators

Synchronization

Implicit Wait

Explicit Wait

Thread.sleep

WebElements

Action, getter, verification methods

Chapter-2

Auto Suggestions

Frames

Select class

Dropdown

Action Class

Right Click

Double Click

Drag and Drop

Mouse hover

Pop_Ups

Authentication Pop up

File upload pop up

File down load pop up

Child browser pop up

Notification pop up

Hidden division pop up

Scroll bar

Screenshots

Chapter:3

Data Driven testing

Pom(Page object model)

TestNg

Chapter:4

Maven project

Framework

Git_hub

jenkins

Java Concepts:

Variables

Methods

Constructors

Loops for, for each ,while loop

If -else

OOPS

Type casting

Access modifiers

Exception handling

Collection

List, set

Array

Eclipse Installation on windows:

Go to Web browser(chrome)----type Eclipse----Click on 1st link(www.eclipse.org)---- click on download(at eclipse official website)----There you can see Eclipse IDE there click on Download*86-64.----Download

Once after downloaded, double click on that downloaded .exe file. So installation page will get open---select Eclipse and IDE for java developer----select JDK version from java 17 + VM dropdown----select the path at installation folder----click on Installing----Once down click on Launch----Eclipse will open.

what is Automation

It is a process of converting any manual test cases into test scripts using automation a tool like selenium and programming language like java.

What is Automation Testing?

Automation Engineer writes code/program by using automation a tool like selenium and programming language like java and runs the script against the application, this script will tests the application and gives us the result pass or fail. Which is nothing but a automation testing.

Why we have to for Automation/Advantages of Automation?

- Reusability of the code
- Avoid Repetition
- Less resources
- Quality of testing will be good
- We can test in a short period of time

Disadvantages of automation?

- 100% of Automation is not possible
- Audio/video related things
- OTP
- Captcha
- Bar code
- Finger prints
- Face recognition
- Anything which need manual intervention, are not automatable.
- Anything which is completely dynamic that cannot be automatable.
- Application or Build which is not stable that cannot be automated.
- It increases Project cost
- Automation Engineer must have a knowledge on Programming language.

When and all we will go for automation?

- Long term projects
- When ever we have to perform Regression testing.
- Whenever we have to do repetitive tasks.
- Automation Resources are necessary.
- Application or Build is stable Only then we can switch for automation.
- When ever we want a qualitative product.
- When we have less time and needs to test the application then we will switch for automation

Types of Automation Tools?

We have 2 types of Automation tools

1. Functional Automation tool
2. Non-Function Automation tools

Functional Automation tool:

To Automate Functional Testcases End to End Test cases then we will go for this Functional Automation Tools.

Eg:

- Selenium WebDriver
- Selenium IDE

- QTP
- Appium
- Winnium
- Rest Assured
- Test Complete

Non-Functional Automation Tools:

Whenever we have to Automate Non functionality of an application like Performance related testcases, then we will switch for Non-Functional Automation Tools

Eg:

- J meter
- No load
- App load
- Neo load

Introduction to Selenium:

Selenium: It is a free opensource tool which is used to automate web applications.

Web Application: Any applications that is accessed by opening the browser and entering the URL, those applications we will called as Web Applications.

Advantages of selenium:

- As it is free tool overall project cost can reduced using selenium.
- Opensource means easy to download and easy to integrate the selenium with 3rd party tool.so customization cost can be reduced.
- Selenium supports multiple programming languages.
- Selenium supports multiple Operating systems
  With this it is easy to perform system compatibility testing.
  Selenium supports multiple browsers with respective driver executables. With this it is easy to perform browser compatibility testing.

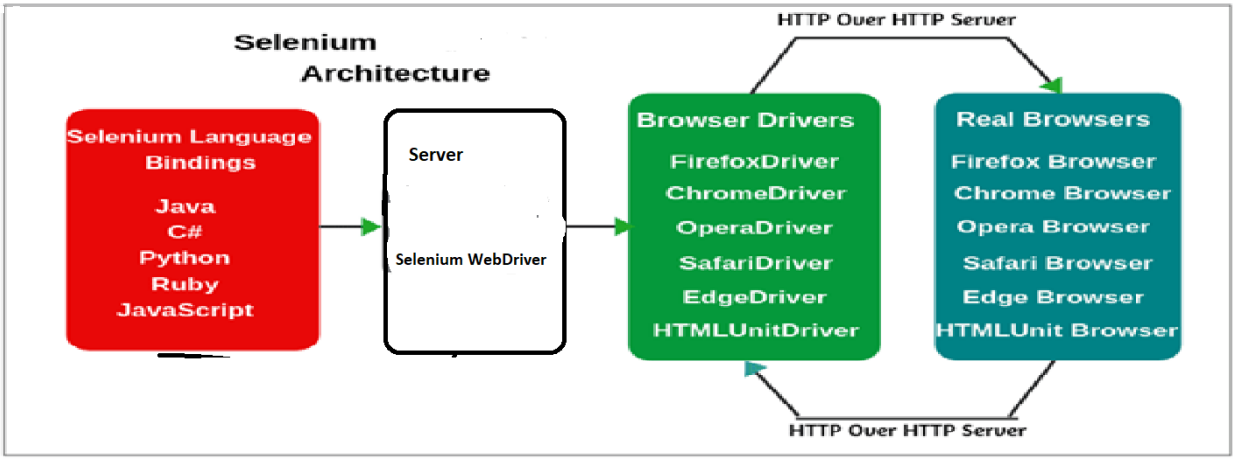
Disadvantages of selenium:

- We can automate only Web Applications.
- Selenium takes help of multiple 3rd party tools to Automate Web Application.
- Through Selenium We cannot Automate OTP, captcha, Finger print related things.

Selenium Components:

- Selenium core
- Selenium IDE
- Selenium WebDriver
- Selenium Rc(Remote Control)
- Winnium
- Selendroid
- Selenium grid
- Appium

Selenium Architecture/Selenium WebDriver Architecture:



Selenium Architecture Explanation:

- Selenium supports multiple Programming languages like java, python, Ruby etc.
- Selenium server internally has java client binding.
- We have driver executables who acts like a translator between server and browse.
- For different browser we have different driver executables.
  Chrome-----> Chrome driver
  Firefox---->gecko driver
  Microsoft edge---> Edge driver

- Using this we are testing the application which is under test (AUT).
- To communicate between the selenium server and browser we use W3C protocol (World wide web consortium).

Maven project:

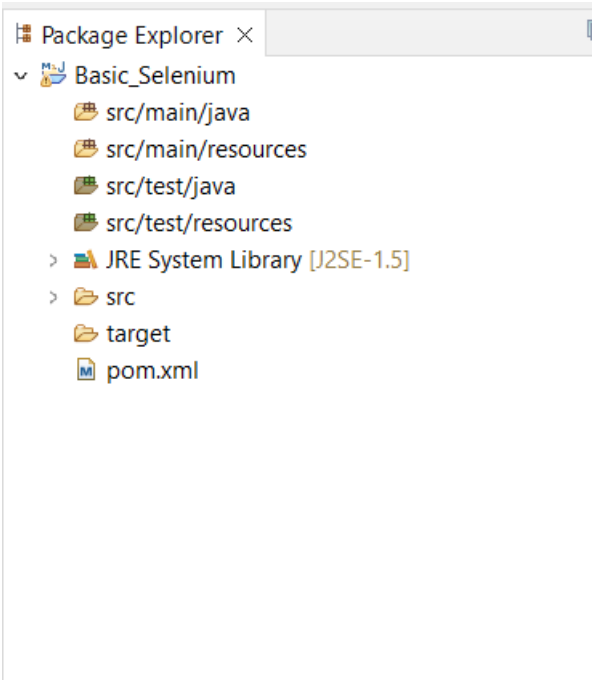
Open Eclipse----File-----New----Project-----Expand maven project-----select maven project---Next-----

Select the check box create a simple maven project---Next---Groupid: organization name(eg:Q_spider)

Artifact Id: Project name(eg: Basic_Selenium_E3)

How

Maven project structure looks like?



<u>Under :</u> **src/test/java create a package with Basic_Programs**

POM.xml file:
POM----Product object model

Follow this if Project Explorer is not Visible on Screen:

Window----Show view---Project Explorer

Adding Selenium Server to our Maven project:

Open google—type maven Repository----select 1st link----type selenium server and enter---
Click on 3.141.59

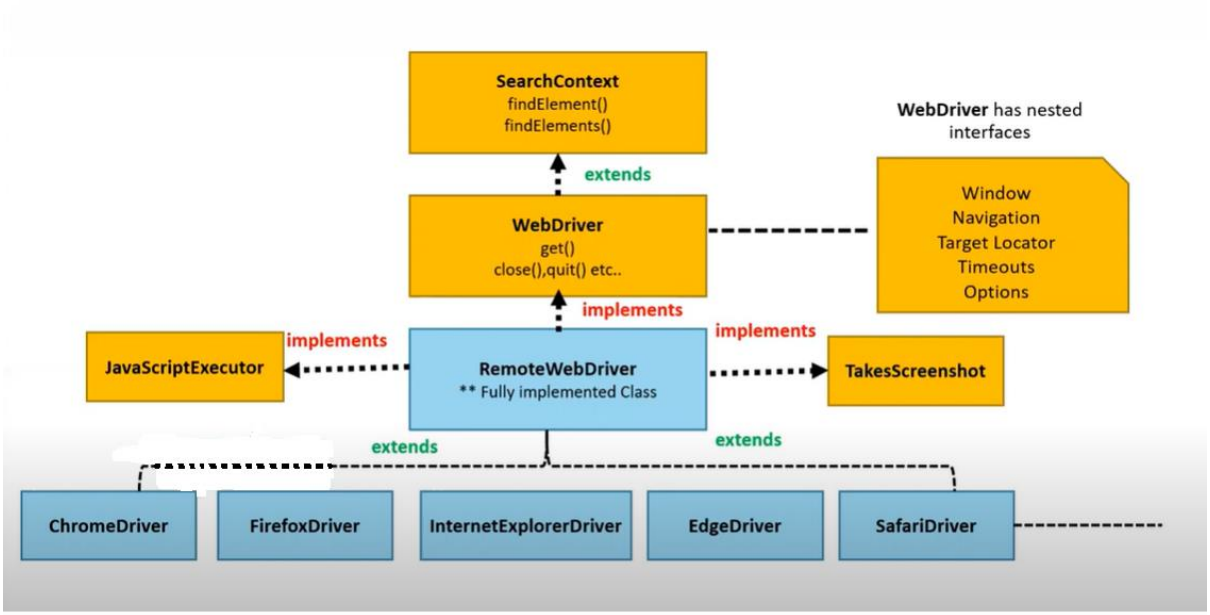
Steps to add WebDriverManager dependencies to pom.xml file:

Open google—type maven Repository----select 1st link----type WebDriverManager and click search---select 1st link---click on 5.0.3---copy and paste dependencies in pom.xml file.

Explanation of
WebDriverManager.chromedriver().setup();
WebDriverManager.firefoxdriver().setup();

- WebDriver Manager is a library which automatically downloads the required driver executables and will sets up the system property.
- So inorder to achieve this we have to add WebDriverManager dependencies to our pom.xml file.
- So that we can call either chromedriver() method or firefoxdriver() method from WebDriverManager library and the we have to call setup() method from WebDriverManager library.
- Exactly WebDriverManager library will set up the chrome or firefox driver executables.

WebDriver Architecture/WebDriver Hierarchy/selenium WebDriver Architecture with respect to java/Class diagram of WebDriver/Hierarchy of Selenium classes and interfaces:



Explanation:

- Search Context is the super most interface in selenium which contains two abstract methods which are below.

Methods	returntype
findElement();	WebElement
findElements();	List<WebElement>

- WebDriver is the sub interface of this SearchContext interface and contains 11 abstract methods.and 2 extended methods of SearchContext. So total 13 methods.

Methods	returntype
1. close()	void
2. get()	void
3. getCurrentURL()	String
4. getPageSource()	String
5. getTitle()	String
6. getWindowHandle()	String
7. getWindowHandles()	set< String>
8. manage()	option
9. navigate()	Navigation
10. quit()	void
11. switchTo()	TargetLocator

- JavaSCriptExecutor is an interface which contains 2 abstract methods

Methods	returntype
1. executescript()	Object
2. executeAsyncScript()	Object

- TakesScreenShot interface which contains 1 abstract method.

Methods	returntype

1. getScreenShotAS() File

ChromeDriver driver=new Chromedriver();

ChromeDriver----->class name

Driver---------->reference variable

= ---------->Assignment operator

new ---------->keyword used to create object

ChroimeDriver()------->Constructor belongs to chromeDriver class

; ------>separator indicates the end of the statement.

- With this statement we can launch the chrome browser

FirefoxDriver driver=new FirefoxDriver ();

FirefoxDriver ----->class name

driver---------->reference variable

= ---------->Assignment operator

new ---------->keyword used to create object

FirefoxDriver ()------->Constructor belongs to FirefoxDriver class

; ------>separator indicates the end of the statement.

- With this statement we can launch the firefox browser

WebDriver driver=new ChromeDriver();

WebDriver driver=new Firefox();

WebDriver ----->interface name

driver---------->reference variable

= ---------->Assignment operator

new ---------->keyword used to create object

FirefoxDriver ()------->Constructor belongs to FirefoxDriver class

ChroimeDriver()------->Constructor belongs to chromeDriver class

; ------>separator indicates the end of the statement.

- These statement launches the browser either in chrome or in firefox depends on the statement what we have given.
- **UpCasting :** Converting child class object to the parent class/parent interface reference variable.
- Here we have done upcasting so that the reference variable is capable of holding multiple objects of multiple classes.

Script: src/test/java create a package with Basic_Topic.

And then create one class with Launching_Browser.

```
package Basic_Topics;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import
io.github.bonigarcia.wdm.WebDriverManager;

public class Launching_Browser {

    public static void main(String[] args)
throws Throwable {

    WebDriverManager.chromedriver().setup();
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

driver.get("https://www.facebook.com/");
```

```
        Thread.sleep(5000);
        String title=driver.getTitle();
        String url = driver.getCurrentUrl();

        System.out.println(title);
        System.out.println(url);
        driver.close();

    }

}
```

Explanation of

`driver.manage().window().maximize();`

This statement is used to maximize the window.

Manage()-------> WebDriver method

Window()-------->method in nested window interface

Maximize()------>method in nested window interface

findElement(): This is used to fetch the address of 1 st matching element.
If it does not fetch the address it will give **No such Element Exception.**
findElements(): This is used to fetch the addresses of all matching element.
If it does not fetch the address it will give **Empty List Exception.**
get(): This is used to enter the url and this will wait until the page get loaded.

sendKeys(): This is used to send the data
click(): this is used to perform click action
getTitle(): This is used to get the title of the page

getCurrentUrl():used to get the current url

close(): It is used to close the application.

contains(): this is used to check the data contains as its arugument.

Navigation API:

Traversing from one page to another page is called as Navigation API.

API --->Application Programming interface

back()--->diver.navigate().back();

farward()---> diver.navigate().forward();

refresh()----> diver.navigate().refresh();

navigate().to() ---->used to navigate from one website to another website.

driver.navigate().to(“url”);

Eg:

```
driver.getTitle().contains("Facebook - log in or sign up")
```

Explanation of

`driver.manage().window().maximize();`

This statement is used to maximize the window.

Manage()-------> WebDriver method

Window()-------->method in nested window interface

Maximize()------>method in nested window interface

Minimize()------>method in nested window interface

fullScreen()------>method in nested window interface

HTML:

HTML is a Hyper Text Markup Language, used by the web designers for developing Web pages.

HTML has 3 elements which are below

1. Tag
2. Attributes
3. Text

Tag: Anything which is enclosed within the angular bracket or 1st element or character enclosed within the angular bracket is called as Tag.

<input type="hidden" name="jazoest" value="2895" autocomplete="off">

- Here input is Tag

Attributes: Anything which is in the form of key and value pair is called as Attributes.

<input type="hidden" name="jazoest" value="2895" autocomplete="off">

- **Here** type="hidden"
  name="jazoest"
  value="2895"

  autocomplete="off" are Attributes

Text: Anything which is written outside the Angular bracket is called as Text.

<a href="https://www.facebook.com/recover/initiate/?priva

cy_mutation_token=eyJ0eXBlIjowLCJjcmVhdGlvbl90aW1lIjoxNjg4OTc4NjA1LCJjYWxsc2l0ZV9pZCI6MzgxMjI5MDc5NTc1OTQ2fQ%3D%3D&ars=facebook_login" waprocessedanchor="true">Forgotten password?

- Here Forgotten password? Is Text

Note: Right click on the web page click on inspect to open Developer tool.

Developer Tool

Inspector