**5 Step by Step Instructions to run first basic Selenium Program**

1. Install Java and Set Java Home Path in System variables
2. Install Eclipse and Create new Maven Project with Selenium Dependencies
3. Understand creation of WebDriver object and its related classes
4. Run the First Selenium WebDriver Program with Browser Invocation
5. Different ways of setting Browser Driver executable files.

**What can Kind of Interview Questions I can expect from above Concepts?**

1. **What is Interface in Java?**

An interface is a group of related methods with empty bodies.

Its class responsibility to implement the methods declared in the Interface

When class agreed to implement the interface, they must need to provide implementation/bodies to all the defined methods in Interface

In simple terms, Interface enforces the Contract to class to follow.

2. **WebDriver is an Interface which provides Set of Browser Automation methods with empty bodies (Abstract methods)**

Classes like ChromeDriver, FirefoxDriver, MicrosoftEdgeDriver , SafariDriver etc implement the WebDriver Interface

and provide their own implementation to the WebDriver methods

3. **We need to create the object of the class to access the methods present in the class.**

ChromeDriver driver = new ChromeDriver ();

driver object here has access to all the methods of Chrome driver

WebDriver driver = new ChromeDriver ();

driver object here has access to the methods of Chrome driver which are defined in web Driver Interface.

Selenium Web Driver Locators

* As part of Automation, Selenium Performs actions (such as click, typing) on the Page HTML Elements.
* The Locators are the way to identify an *HTML* element on a web page.   
  Selenium WebDriver uses any of the below locators to identify the element on the page and performs the Action

ID

Xpath

CSS Selector

name

Class Name

Tag Name

Link Text

Partial Link Text

<input type="text" placeholder="Username" id= “inputUsername” value=" ">

Input -> tag name

Red-> attribute   
Green-> attribute associated value.

**Css Selector-**

* **Class name -> tagname.classname ->** Button.signInBtn -> .error
* **Id -> tagname#id** -> input#inputUsername
* **Tagname[attribute=’value’]**

Input[placeholder=’Name’]

<input type="text" placeholder="Username” value=" ">

Input [placeholder=’ Username’]

* **//Tagname[@attribute=’value’]:nth-child(index). - Child items**
* **Parenttagname childtagname**
* **input[type\*='pass'] – CSS**
* **tagname**

<button class="reset-pwd-btn" xpath="1" style="">Reset Login</button>

**Xpath –**

* **//Tagname[@attribute=’value’]**

**//input[@**placeholder=’ Username’’]

**<input type="text" placeholder="Name">**

**//input[@**placeholder=’ **Name’**]

* **//Tagname[@attribute=’value’][index]**
* **//parentTagname/childTagname**
* **//button[contains(@class,'submit')]. – Regular expression**
* **//tagname**
* **//header/div/button[1]/following-sibling::button[1]**
* **//header/div/button[1]/parent::div**

WebDriver driver = **new** ChromeDriver();

So in this chrome driver class(after new word ), there is a method called find element. Web driver interface exposed to this find element method, Chrome Driver Class is implementing this interface and implementing the body of find element exposed by a web driver interface.

Xpath and css selector locators are generic locators where you can construct these with any html element on the page.

Use of thread.sleep(2000) basically, script will pause for one second before it actually act on that new view of forgot your password bit.

by reading this one second, things will come into the stable mode and then now you run, you won’t get the error.

So when you get a question like this, you have to tell that probably when that application is in changing state from one view to another.You might run into that interception error at that time. You have to wait until that will get stable.So in this case, I have waited for one second. You can also use it explicit. Wait, at this time, we are still not in that topic when we talk about our synchronization section. Then I will explain what is explicit

**Code download**

import java.time.Duration;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Locators {

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

// TODO Auto-generated method stub

//implicit wait - 2 seconds time out

System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

driver.get("https://rahulshettyacademy.com/locatorspractice/");

driver.findElement(By.id("inputUsername")).sendKeys("rahul");

driver.findElement(By.name("inputPassword")).sendKeys("hello123");

driver.findElement(By.className("signInBtn")).click();

System.out.println(driver.findElement(By.cssSelector("p.error")).getText());

driver.findElement(By.linkText("Forgot your password?")).click();

Thread.sleep(1000);//

driver.findElement(By.xpath("//input[@placeholder='Name']")).sendKeys("John");

driver.findElement(By.cssSelector("input[placeholder='Email']")).sendKeys("john@rsa.com");

driver.findElement(By.xpath("//input[@type='text'][2]")).clear();

driver.findElement(By.cssSelector("input[type='text']:nth-child(3)")).sendKeys("john@gmail.com");

driver.findElement(By.xpath("//form/input[3]")).sendKeys("9864353253");

driver.findElement(By.cssSelector(".reset-pwd-btn")).click();

System.out.println(driver.findElement(By.cssSelector("form p")).getText());

driver.findElement(By.xpath("//div[@class='forgot-pwd-btn-conainer']/button[1]")).click();

Thread.sleep(1000);

driver.findElement(By.cssSelector("#inputUsername")).sendKeys("rahul");

driver.findElement(By.cssSelector("input[type\*='pass']")).sendKeys("rahulshettyacademy");

driver.findElement(By.id("chkboxOne")).click();

driver.findElement(By.xpath("//button[contains(@class,'submit')]")).click();

}

}

Section6: 41. We are learned assertions, tagname element, experted and actual value assertions & only p tag is there then.

41.what is assertion  ?

ssertion expected versus actual value validation should be handled automatically in

the code only. So that's why if you remember initially when we are downloading the dependencies, we have used this testNG. testNG framework which will give us a lot of assertions for validate. So make sure you have this dependency.

Now from this test, I will introduce one assertion.So here do remember that you remove scope.So that's not required. If you put it that, then you will not see that test related jars in your actual program files.

what does this assertion do?

It will get the text from the screen and the text whatever it retrieved.It will compare with this.If both matches test will pass. If any one of the mismatch happened here, then test will fail. So that's how we use assertions.

if you don't have that, wait here.

What happens now?If you ask to get me the tag name P within milliseconds, selenium will go from this step to this step.After clicking on the sign in button selenium within no time.Not even one millisecond.It will immediately go to step number 22 and it will try to find the tag name p dot get text. So here it it takes at least one second to change the state of the page.

42. //button[text()='Log Out'] - this is we are using only in x path if you now the text of the button.

In the syntax if instead of button you can mention \* also but it take all the tags where the name is present in logout

driver.findElement(By.*xpath*("//button[text()='Log Out']")).click();

driver.close();

43. how to remove this hard coding the password ? get the password dynamically from the page

Instead of writing  everything in this one main block, I'll create one new method for giving the password.

you can create methods in your class also, but that methods should go outside of the main execution block

 if you want to call any method in the class into main block.

There are two ways either you can create the object of the class and call that method with object.method.

If you move your cursor, it will tell you that firstly, you have to pass the driver.

OK, so this method accepts driver argument.See the driver what you created.

Now you will pass into to this method argument. And now if you move your cursor thing that changes to static, like when you change this method to static,

then you can directly call this method without creating the object.

how you dynamically send the password by extracting and passing your password string.

**import** java.time.Duration;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.edge.EdgeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.testng.Assert;

**public** **class** Locators2 {

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

// **TODO** Auto-generated method stub

String name = "rahul";

System.*setProperty*("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

WebDriver driver = **new** ChromeDriver();

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

// WebDriver driver = new EdgeDriver();

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(5));

String password = *getPassword*(driver);

driver.get("https://rahulshettyacademy.com/locatorspractice/");

driver.findElement(By.*id*("inputUsername")).sendKeys(name);

driver.findElement(By.*name*("inputPassword")).sendKeys(password);

driver.findElement(By.*className*("signInBtn")).click();

Thread.*sleep*(2000);

System.***out***.println(driver.findElement(By.*tagName*("p")).getText());

Assert.*assertEquals*(driver.findElement(By.*tagName*("p")).getText(), "You are successfully logged in.");

Assert.*assertEquals*(driver.findElement(By.*cssSelector*("div[class='login-container'] h2")).getText(),"Hello "+name+",");

driver.findElement(By.*xpath*("//\*[text()='Log Out']")).click();

driver.close();

}

**public** **static** String getPassword(WebDriver driver) **throws** InterruptedException

{

driver.get("https://rahulshettyacademy.com/locatorspractice/");

driver.findElement(By.*linkText*("Forgot your password?")).click();

Thread.*sleep*(1000);

driver.findElement(By.*cssSelector*(".reset-pwd-btn")).click();

String passwordText =driver.findElement(By.*cssSelector*("form p")).getText();

//Please use temporary password 'rahulshettyacademy' to Login.

String[] passwordArray = passwordText.split("'");

// String[] passwordArray2 = passwordArray[1].split("'");

// passwordArray2[0]

String password = passwordArray[1].split("'")[0];

**return** password;

//0th index - Please use temporary password

//1st index - rahulshettyacademy' to Login.

//0th index - rahulshettyacademy

//1st index - to Login

}

}

46. Identify locators using Siblings with Xpath traverse

//header/div/button[1] - parent to child

/following/sibiling::tagname[1] - sibling to another sibling

WebDriver driver = **new** ChromeDriver();

// Sibling - Child to parent traverse,parent to child, sibiling to sibiling

//header/div/button[1]/following-sibling::button[1]

driver.get("https://rahulshettyacademy.com/AutomationPractice/");

System.***out***.println(driver.findElement(By.*xpath*("//header/div/button[1]/following-sibling::button[1]")).getText());

System.***out***.println(driver.findElement(By.*xpath*("//header/div/button[1]/parent::div/button[2]")).getText());

when you say absolute xpath, so that means you will start from the top. the main parent.

This is the route object right from the root object.You will start travelling to your child element if you do in that way. That means you are right being an absolute xpath.

Absolute x path starts from single slash. Single slash represents that you are starting from the root of this html.

Relative xpath - you are jumping into the middle, your writing the path from the middle. Relative path use double slash.

And do remember that these kind of travel thing back from child to parent is not possible in CSS.

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class WindowActivities {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "/Users/rahulshetty/Documents/chromedriver");

WebDriver driver = new ChromeDriver();

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

driver.get("http://google.com");

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

driver.navigate().back();

driver.navigate().forward();

}

}

Interview Question -

**Core Java Basics needed to Get started with Automation –**

* **Variables & Data types in Java**
* **Working with Arrays**
* **Loops & Conditions**
* **Strings and its functions**
* **Importance of Array Lists**
* **Array list operations and conversion of Array to List**
* **Declaring Methods**
* **Accessing Methods in class & Static keyword**

**Note: Rest all topics are present in the last 2 sections of this course.**