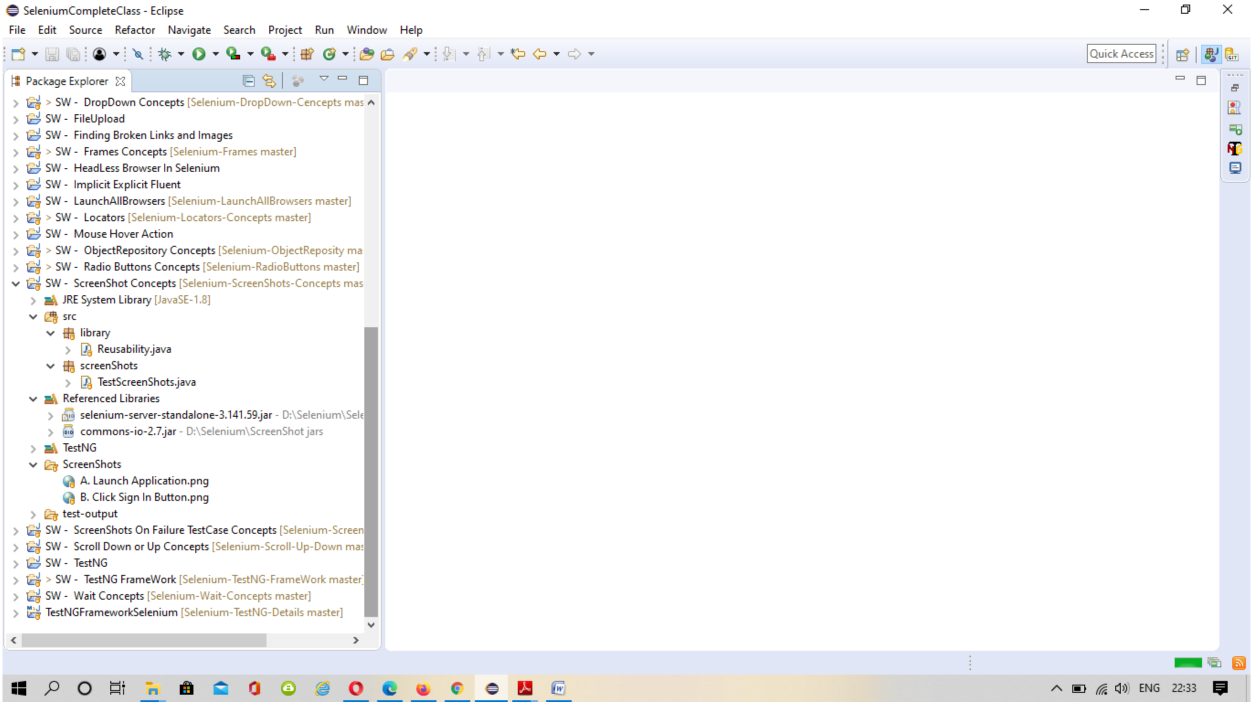
ScreenShot in Selenium WebDriver

- **Why ScreenShot required in Automation?**
- **How to capture screenshot in selenium?**
- **How to create library to utilize the code?**

- **Why ScreenShot required in Automation?**
  1. Screenshot helps us the understand the flow of application weather application behaving correctly or not.
  2. It helps as while doing cross browser testing.
  3. If you are working with headless browser then you can track the execution of test. (Headless Means code will run in back end by show only gui. Ex. PhantomJS, HTMLUnt…)
  4. Screenshot on failure.

- **How to capture screenshot in selenium?**


SeleniumCompleteClass - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
> SW - DropDown Concepts [Selenium-DropDown-Cencepts mas
SW - FileUpload
SW - Finding Broken Links and Images
> SW - Frames Concepts [Selenium-Frames master]
SW - HeadLess Browser In Selenium
SW - Implicit Explicit Fluent
SW - LaunchAllBrowsers [Selenium-LaunchAllBrowsers master]
> SW - Locators [Selenium-Locators-Concepts master]
SW - Mouse Hover Action
> SW - ObjectRepository Concepts [Selenium-ObjectReposity ma
> SW - Radio Buttons Concepts [Selenium-RadioButtons master]
SW - ScreenShot Concepts [Selenium-ScreenShots-Concepts mas
JRE System Library [JavaSE-1.8]
src
library
Reusability.java
screenShots
TestScreenShots.java
Referenced Libraries
selenium-server-standalone-3.141.59.jar - D:\Selenium\Sele
commons-io-2.7.jar - D:\Selenium\ScreenShot jars
TestNG
ScreenShots
A. Launch Application.png
B. Click Sign In Button.png
test-output
SW - ScreenShots On Failure TestCase Concepts [Selenium-Screen
SW - Scroll Down or Up Concepts [Selenium-Scroll-Up-Down mas
SW - TestNG
> SW - TestNG FrameWork [Selenium-TestNG-FrameWork master]
SW - Wait Concepts [Selenium-Wait-Concepts master]
TestNGFrameworkSelenium [Selenium-TestNG-Details master]
ENG 22:33

```java
package screenShots;

//import java.io.File;
//import org.apache.commons.io.FileUtils;  // Import Jar file from MVNRepository and
To Project
import org.openqa.selenium.By;
//import org.openqa.selenium.OutputType;
//import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

import library.Reusability;

public class TestScreenShots {
    @Test
    public void Browser() throws Exception {
        System.setProperty("webdriver.chrome.driver",
                "D:\\Selenium\\Selenium Browsers Jars\\Chrome
83\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.easycalculation.com/");
        driver.manage().window().maximize();

        Reusability.capturedScreenShot(driver, "A. Launch Application");

        driver.findElement(By.xpath("(//span[contains(.,'Sign
in')])[2]")).click();

        Reusability.capturedScreenShot(driver, "B. Click Sign In Button");

        driver.quit();
    }
}

/*      TakesScreenshot ts = (TakesScreenshot) driver;
File source = ts.getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(source, new File("./ScreenShots/EasyCalculation.png"));
System.out.println("Captured ScreenShot");*/
```

```java
package library;

import java.io.File;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

public class Reusability {

    public static void capturedScreenShot(WebDriver driver, String screenShotName)
{
        try {
            TakesScreenshot ts = (TakesScreenshot) driver;
            File source = ts.getScreenshotAs(OutputType.FILE);
            FileUtils.copyFile(source, new File("./ScreenShots/" +
screenShotName + ".png"));
            System.out.println("Captured ScreenShot - By Selenium
WebDriver");

        } catch (Exception e) {
            System.out.println("Exception While Taking Screen Shot" +
e.getMessage());
        }
    }
}
```

First Program 1:

```
package screenShots;

import java.io.File;
import org.apache.commons.io.FileUtils;  // Import Jar file from MVNRepository and To Project
import org.openqa.selenium.By;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;


public class TestScreenShots {
    @Test
    public void Browser () throws Exception {
        System.setProperty("webdriver.chrome.driver",
                "D:\\Selenium\\Selenium Browsers Jars\\Chrome 84\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.easycalculation.com/");
        driver.manage().window().maximize();
        driver.findElement(By.xpath("(//span[contains(.,'Sign in')])[2]")).click();

        TakesScreenshot ts = (TakesScreenshot) driver;
        File source = ts.getScreenshotAs(OutputType.FILE);
        FileUtils.copyFile(source, new File("./ScreenShots/EasyCalculation.png"));
        System.out.println("Captured ScreenShot");
        driver.quit();
    }
}
```

- **How to create library to utilize the code?**

Step 1. Create Library package and create class Reusability
Step 2. Copy Above code

```
        TakesScreenshot ts = (TakesScreenshot) driver;
        File source = ts.getScreenshotAs(OutputType.FILE);
        FileUtils.copyFile(source, new File("./ScreenShots/EasyCalculation.png"));
        System.out.println("Captured ScreenShot");
```

Step 3. Surround with Try and catch method Code looks like below:

```
 package library;

import java.io.File;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;

public class Reusability {

public static void capturedScreenShot(WebDriver driver, String screenShotName) {

        try {
                TakesScreenshot ts = (TakesScreenshot) driver;
                File source = ts.getScreenshotAs(OutputType.FILE);
                FileUtils.copyFile(source, new File("./ScreenShots/" + screenShotName + ".png"));
                System.out.println("Captured ScreenShot - By Selenium WebDriver");

        } catch (Exception e) {
                System.out.println("Exception While Taking Screen Shot" + e.getMessage());
                }
        }
}
```

Step 4: Alter the above program

```
package screenShots;

//import java.io.File;
//import org.apache.commons.io.FileUtils;  // Import Jar file from MVNRepository and To Project
import org.openqa.selenium.By;
//import org.openqa.selenium.OutputType;
//import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

import library.Reusability;

public class TestScreenShots {
        @Test
        public void Browser() throws Exception {
                System.setProperty("webdriver.chrome.driver",
                                "D:\\Selenium\\Selenium Browsers Jars\\Chrome 84\\chromedriver.exe");
                WebDriver driver = new ChromeDriver();
                driver.get("https://www.easycalculation.com/");
                driver.manage().window().maximize();

                Reusability.capturedScreenShot(driver, "A. Launch Application");
                driver.findElement(By.xpath("(//span[contains(.,'Sign in')])[2]")).click();
                Reusability.capturedScreenShot(driver, "B. Click Sign In Button");

                driver.quit();
        }
}
```

How to take screen shot on failed test case in selenium webdriver

- ITestResult
- @AfterMethod
- ScreenShot on Failure

ITestResult

https://www.javadoc.io/doc/org.testng/testng/6.8.21/org/testng/ITestResult.html

It is an interface which keeps all information about the test case which we executed

We will capture some information from this like

TestCase Execution Status, TestCase Name ....etc.

@AfterMethod

Its an annotation of TestNG and this will execute after Every Test.