

Selenium

Headless Browser Testing

In this session, we are going to look into headless browser testing using Selenium. It was a request from a YouTube subscriber and a connection from LinkedIn. A headless browser is a browser that does not have a User Interface. Therefore, we will not see our Test Script execute in the browser UI. However, the benefit is Test Scripts execute much faster in a headless browser.

In our Test Script, I'm going to show you how to perform this process using 3 drivers: HTMLUnitDriver, ChromeDriver, and FirefoxDriver. We can also use EdgeDriver which is similar to ChromeDriver and FirefoxDriver. PhantomJS has been discontinued as a way to perform headless browser testing.

For HTMLUnitDriver, we must get the dependency from [Maven's Repository](#) or download the jars. I prefer to use the dependency so let's search for SeleniumHTMLUnitDriver. The version is 2.52, copy the dependency, and go to our pom.xml file to paste the dependency.

```
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-htmlunit-driver</artifactId>
  <version>2.52.0</version>
</dependency>
```

Reload all Maven Projects. Our Test Script starts with @Test public void htmlHeadlessBrowser () { }
HtmlUnitDriver driver = new HtmlUnitDriver(); We can also use WebDriver as the type and not HtmlUnitDriver. By default, HtmlUnitDriver runs without a browser. Therefore, the next step is to load the browser. Before I load the browser, Let's go to the AUT which will be OrangeHRM. The Test Script will enter Username, Password, then click the Login button. Admin is the Username. admin123 is the Password.



Inspect the Username and we see an id value of txtUsername, Password has an id value of txtPassword, the Login button has an id value of btnLogin. Copy the [URL](https://opensource-demo.orangehrmlive.com/) and go back to our Test Script.

Now, we write `driver.get("https://opensource-demo.orangehrmlive.com/");` Next step is to find all of the elements `driver.findElement(By.id("txtUsername")).sendKeys("Admin");`
`driver.findElement(By.id("txtPassword")).sendKeys("admin123");` Next is to click the button `driver.findElement(By.id("btnLogin")).click();` Last step is to print the Page Title. This will help us know the Test Script executed as expected because we are not going to see the browser. `sout("HTML Title: " + driver.getTitle());`

```
@Test
public void htmlHeadlessBrowser () {
    WebDriver driver = new HtmlUnitDriver();
    driver.get("https://opensource-demo.orangehrmlive.com/");
    driver.findElement(By.id("txtUsername")).sendKeys("Admin");
    driver.findElement(By.id("txtPassword")).sendKeys("admin123");
    driver.findElement(By.id("btnLogin")).click();
    System.out.println("HTML Title: " + driver.getTitle());
}
```

In order to run the test for HTMLUnitDriver, we must remove our dependency for selenium-java. If not, there will be a SessionNotFoundException. In the pom.xml file, I'm going to comment out the dependency for selenium-java <!-- --> then Reload our Maven Project.

```
<!--
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.0.0-alpha-6</version>
    </dependency>
-->
```

Now, let's run the HTMLUnitDriver Test Script. It passed, there was no browser but the console shows HTML Title: OrangeHRM.

There are 2 ways to perform headless browser testing using ChromeDriver and FirefoxDriver. I'm going to show you both ways starting with ChromeDriver. First, let me remove the comment for selenium-java, reload the project and start with Chrome. @Test public void chromeHeadlessBrowserOne () { } One is the first way, I'm going to show you and first is to setup the chromedriver.

WebDriverManager.chromedriver().setup(); ChromeOptions options = new ChromeOptions(); ChromeOptions is a class for managing options strictly for Chrome. At this point, our options. is to set the browser to headless with a value of true. Pass in the object reference is the last step. Now, we are going to copy and paste remaining Test Script to load the login page. When I say pass in the object reference, WebDriver driver = new ChromeDriver(options); This is where I pass in options. It is the object reference that I pass in. I'm going to put a space right there. Now, I'm finished with ChromeDriver the first way. I'm going to change the title to Chrome Title 1 and that's all we do.

```
@Test
public void chromeHeadlessBrowserOne () {
    WebDriverManager.chromedriver().setup();
    ChromeOptions options = new ChromeOptions();
    options.setHeadless(true);
    WebDriver driver = new ChromeDriver(options);

    driver.get("https://opensource-demo.orangehrmlive.com/");
    driver.findElement(By.id("txtUsername")).sendKeys("Admin");
    driver.findElement(By.id("txtPassword")).sendKeys("admin123");
    driver.findElement(By.id("btnLogin")).click();
    System.out.println("Chrome Title 1: " + driver.getTitle());
}
```

With the 2nd way, we only change setHeadless to addArguments. Copy and Paste and change One to Two. options.setHeadless go to addArguments("--headless") This method let's our program know we want to set Chrome to be headless. Change Chrome Title 1 to 2. That's it.

```
@Test
public void chromeHeadlessBrowserTwo () {
    WebDriverManager.chromedriver().setup();
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--headless");
    WebDriver driver = new ChromeDriver(options);

    driver.get("https://opensource-demo.orangehrmlive.com/");
    driver.findElement(By.id("txtUsername")).sendKeys(...keysToSend: "Admin");
    driver.findElement(By.id("txtPassword")).sendKeys(...keysToSend: "admin123");
    driver.findElement(By.id("btnLogin")).click();
    System.out.println("Chrome Title 2: " + driver.getTitle());
}
```

Those are the 2 ways to perform headless browser testing in Chrome. It's the same steps for FirefoxDriver so I'm going to copy and paste these steps and change Chrome to Firefox.

```
@Test
public void firefoxHeadlessBrowserOne () {
    WebDriverManager.firefoxdriver().setup();
    FirefoxOptions options = new FirefoxOptions();
    options.setHeadless(true);
    WebDriver driver = new FirefoxDriver(options);

    driver.get("https://opensource-demo.orangehrmlive.com/");
    driver.findElement(By.id("txtUsername")).sendKeys(...keysToSend: "Admin");
    driver.findElement(By.id("txtPassword")).sendKeys(...keysToSend: "admin123");
    driver.findElement(By.id("btnLogin")).click();
    System.out.println("Firefox Title 1: " + driver.getTitle());
}
```

```
@Test
public void firefoxHeadlessBrowserTwo () {
    WebDriverManager.firefoxdriver().setup();
    FirefoxOptions options= new FirefoxOptions();
    options.addArguments("--headless");
    WebDriver driver = new FirefoxDriver(options);

    driver.get("https://opensource-demo.orangehrmlive.com/");
    driver.findElement(By.id("txtUsername")).sendKeys(...keysToSend: "Admin");
    driver.findElement(By.id("txtPassword")).sendKeys(...keysToSend: "admin123");
    driver.findElement(By.id("btnLogin")).click();
    System.out.println("Firefox Title 2: " + driver.getTitle());
}
```

Let's Run for all of the methods. No browser for Chrome or Firefox but we see Chrome Title 1: OrangeHRM, Chrome Title 2: OrangeHRM, Firefox Title 1: OrangeHRM, Firefox Title 2: OrangeHRM and we see the exception for HtmlUnitDriver SessionNotFoundException. That's it and Thanks for watching and I'll see you in the next video. If you are interested in more videos, feel free to subscribe to my [YouTube channel](#) and click the bell icon. You can also follow me on [Twitter](#), connect with me on [LinkedIn](#) and [Facebook](#). I'll make sure to place the transcript and code on [GitHub](#).

Contact

- ✓ YouTube <https://www.youtube.com/c/RexJonesII/videos>
- ✓ Facebook <https://facebook.com/JonesRexII>
- ✓ Twitter <https://twitter.com/RexJonesII>
- ✓ GitHub <https://github.com/RexJonesII/Free-Videos>
- ✓ LinkedIn <https://www.linkedin.com/in/rexjones34/>