



Selenium Easy

For Complete Web Automation Testing Tutorials using Selenium Webdriver

[Home](#) >> [Selenium Tutorials](#) >> [Handling Authentication Window with WebDriver \(In Firefox, Chrome and IE\)](#)

Handling Authentication Window with WebDriver (In Firefox, Chrome and IE)

When you are working in a test environment, Stage or Pre Production, there are cases where you may need to work with applications which are secured with Authentication (Basic Auth).

When ever you enter the URL, it will prompt you to enter the User name and the password and It will not allow to perform any further operations until you provide username and password. And this Authentication pop-up is not a JavaScript pop-up, it is a Browser dialog window which selenium cannot handle simply using sendKeys method which we do for normal JavaScript pop-ups..

To work with Basic Authentication pop-up (which is a browser dialogue window), you just need to send the user name and password along with the application URL.

Syntax:

```
driver.get("http://admin:admin@yoururl.com");
```

Check out the example below to execute in Firefox browser:

```
@Test
    public void testBasicAuth_Firefox() {

        WebDriver driver = new FirefoxDriver();
        driver.manage().window().maximize();
        driver.get("http://admin:admin@yoururl.com");
        //To check if we have landed in the correct place
        String text = driver.findElement(By.className("home")).getText();
        Assert.assertTrue(text.contains("Welcome"), "Basic Authentication failed");
    }
```

Check out the example below to work with Chrome browser:

```

@Test
    public void testBasicAuth_Chrome() {

        System.setProperty("webdriver.chrome.driver", "G:/Jars/chromedriver.exe");

        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://admin:admin@yoururl.com");
        //To check if we have landed in the correct place
        String text = driver.findElement(By.className("home")).getText();
        Assert.assertTrue(text.contains("Welcome"), "Basic Authentication failed");
    }

```

The best way to make it work with IE is using Autolt tool. If not, you may need to change the stuff in registry, To change for the current user, you need to edit in 'HKEY_CURRENT_USER...' and if you want to do that for all users, you can set the value of register keys as 'HKEY_LOCAL_MACHINE...' etc.

Once you open the URL in IE it will look like the below screen shot: -



Check out the example to work with IE using Autolt tool.

First create Autolt script as below and save it as basicauth.au3

```

; To pass user name and password
WinWaitActive("Windows Security")
Send("admin")
Send("{TAB}")
Send("admin")
Send("{ENTER}")

```

After creating the above AutoIT script, compile the script and take the location of the script exe file. Now the selenium code should look like below :

```
@Test
    public void testBasicAuth_IE() {
        DesiredCapabilities caps = DesiredCapabilities.internetExplorer();
        caps.setCapability(InternetExplorerDriver.INITIAL_BROWSER_URL, "");

        System.setProperty("webdriver.ie.driver", "G:/Jars/IEDriverServer.exe");

        WebDriver driver=new InternetExplorerDriver(caps);
        driver.manage().window().maximize();
        driver.get("http://yoururl.com");
        try {
            Runtime.getRuntime().exec("G:/basicauth.exe");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

Hope the above solutions works.

◀ Optional annotation in TestNG [node:title] ▶

Selenium Tutorials:

Selenium Tutorials

Selenium Tutorials

- › Introduction to selenium
- › Basic HTML concepts required for Selenium
- › Locators for Selenium
- › How to Run Webdriver in chrome browser?
- › How to run webdriver in IE browser?
- › Selenium webdriver using Eclipse
- › Synchronization in Selenium Webdriver
- › XPath tutorial for Selenium
- › CSS selectors for Selenium with example
- › Firefox Not Connected Exception in Selenium
- › Handle windows popups using Selenium Webdriver
- › Working with Frames in Selenium Webdriver
- › How to handle javascript alerts, confirmation and prompts?
- › Defining Firefox Profile preferences using Selenium Webdriver
- › Why Automation Testing is required?

- › Importance of Automation Testing
- › Mouse Hover Actions in Selenium Webdriver
- › StaleElementReferenceExceptions in Selenium Webdriver
- › Difference between Webdriver get() and Webdriver navigate()
- › Navigation Methods in Webdriver with Examples
- › Handling Cookies in Webdriver
- › How to addcookie with Selenium Webdriver
- › How to delete Cookies in Selenium Webdriver
- › WebDriver Methods
- › Automation Test Frameworks
- › Taking Screenshot using Webdriver
- › Webdriver SELECT Methods to work with Dropdowns
- › Webdriver Select with Multiple Attribute
- › Handling keyboard events and mouse hover events using Webdriver
- › Set browser width and height in Selenium Webdriver
- › Selenium Automation Framework Example
- › UI Map or Object Repository using Properties File
- › Test Data in automation framework
- › Reporting Structure in Selenium Automation Framework
- › Test Configuration File / Executable File
- › Uploading a file with Selenium Webdriver
- › Upload file using AutoIT
- › Uploading a file with sendKeys method
- › Read data from Properties file using Java Selenium
- › Scrolling web page with Selenium Webdriver using java
- › Keyword Driven Framework Example
- › Install AutoIT in Windows
- › Examples with xpath and Css (ID, Name, Text and Links)
- › Testing in multiple browsers using selenium and testng
- › Working with Checkbox using Resuable Methods
- › Working with Select examples
- › File upload using Robots
- › Download file using selenium webdriver
- › Perform operations on new window
- › Introduction to Page Object Model Framework
- › Mouse hover using javaScript Executor

- › Working with Multiple Checkboxes with Safecheck
- › Click element using JavaScriptExecutor
- › Drag and Drop using Webdriver Action Class
- › WebDriver Waits Examples
- › Simple Page Object Model example
- › Double Click on element using Webdriver
- › Right Click Context Menu example
- › Working with AutoComplete Text box
- › Working with Date Picker Calendar
- › Working with Ajax controls using Webdriver
- › Get Attribute values using Webdriver
- › Get Css values using Webdriver
- › Resizing a web element using movebyoffset
- › Validate Downloaded file after clicking on downloaded button/ link
- › How to verify entries in Exported CSV file
- › Element is not clickable at point
SeleniumWebdriverException
- › Extract PDF text And Verify Text Present in PDF using WebDriver
- › Handling Authentication Window with WebDriver (In Firefox, Chrome and IE)

Recent Post

Handling Authentication Window with WebDriver
(In Firefox, Chrome and IE)

Selenium Webdriver Tutorials

Locators for Selenium

Optional annotation in TestNG

Parameterization in TestNG using testng.xml

Automation Framework

- › Framework Structure
- › Object Repository
- › Test Data
- › Test Scripts
- › Reports
- › Configuration and Execution

Tags Cloud

Asserts(1) AutoComplete(1) Autolt(2) Automation
(3) Build.xml(1) Checkboxes(2) Cookies(3) Css(2)
Css Examples(2) DataProvider(1) Date Picker(1)
Drag and Drop(1) Exceptions(4) File Download(3)
Firefox Profile(2) Framework Frameworks(5) iFrames
(1) Interview Questions(1) JavascriptExecutor(3)
JUnit(2) Keyword Driven(1) Listeners(3)
Manual Testing(1) Maven(5) Mouse Hover(2)
Page Object Model(2) Page Scroll(1) Parallel Test

Home

- » [Home](#)
- » [Tutorials](#)
- » [About](#)
- » [Contact](#)
- » [Privacy Policy](#)
- » [Site Map](#)

Tutorial Menu

- » [Selenium Tutorials](#)
- » [TestNG Tutorial](#)
- » [JUnit Tutorial](#)
- » [JXL Tutorial](#)
- » [Apache POI](#)
- » [ANT Tutorial](#)
- » [Maven Tutorial](#)

Selenium Tutorials

- » [Basic Webdriver Tutorials](#)
- » [Advanced Webdriver Tutorials](#)
- » [Selenium Framework Examples](#)
- » [JavascriptExecutor Examples](#)

Interview Questions

- » [Testing Concepts](#)
- » [Quality Standards](#)
- » [General Questions](#)
- » [Roles & Responsibilities](#)
- » [Technology & Environment](#)

We welcome your questions and requests