Developer Tips: C# Selenium with MSTest Basics

Selenium is a great tool for testing your UI. There are plenty of great tutorials on the web that I encourage you to review. This article is going to cover some basic setup steps, a simple .NET Core 2.1 code sample, and promote additional resources such as an intermediate session you may attend at Dog Food Con 2019 to learn how to incorporate your Selenium tests into your DevOps pipeline.

# Browser settings

1. Assumptions: Chrome/Firefox (64-bit)/IE11/Edge (Win10 or higher).
   1. For this article I'll test against Chrome.
2. Most of these settings have to be done for IE11 as the modern browsers do this by default or the alternative usually still works.
   1. Always open pop-ups in a new tab.
   2. Turn off pop-up blockers.
   3. IE11: Enable Protected Mode for all security zones.
   4. Disable save password prompts.
   5. When prompted to AutoComplete, click "No".
   6. Set zoom to 100%.
3. Restart the browsers

# Windows settings

Links and instructions target Windows 10 (Win10), but most steps work with minor adjustments based on Windows version. Notes are provided if there are any version specific limitations.

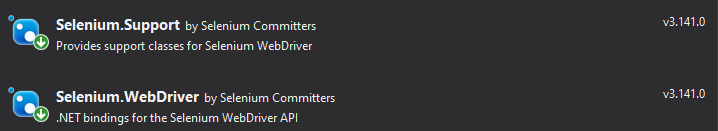
1. [Disable the logon screen saver](https://support.microsoft.com/en-us/help/185348/how-to-change-the-logon-screen-saver-in-windows) so while you're sitting back watching your automated tests run the screen saver does not kick on ruining your test.
2. Restart the computer

# Setup the WebDrivers

1. IE 11
   1. <https://www.seleniumhq.org/download/>
   2. Under "The Internet Explorer Driver Server" section > click "32 bit Windows IE"
      1. 64 bit should also work, but some consultants I worked with recommended the 32 bit over 64 bit as of 12/2018
   3. Extract "IEDriverServer.exe" from the zip to c:\Selenium.WebDrivers
2. Microsoft Edge (EdgeHtml)
   1. <https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>
   2. Edge version 18 or great, then run the following in command prompt as an administrator
      1. DISM.exe /Online /Add-Capability /CapabilityName:Microsoft.WebDriver~~~~0.0.1.0
   3. Edge version less than 18, then do the following
      1. Under "Downloads" > Microsoft Edge (EdgeHtml) > click the top Release #####
      2. Save "MicrosoftWebDriver.exe" to c:\Selenium.WebDrivers
3. Microsoft Edge (Chromium)
   1. Since this version is in Preview I did not download and test but here are the steps
   2. <https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>
   3. Under "Downloads" > Microsoft Edge (Chromium) > for the top Release ##### click x64
   4. Extract "msedgedriver.exe " from the zip to c:\Selenium.WebDrivers
4. Chrome
   1. <https://sites.google.com/a/chromium.org/chromedriver/>
   2. Under "All versions available in Downloads" next to the "Latest stable release" click the ChromeDriver #.## link > Click "chromedriver\_win32.zip"
   3. Extract "chromedriver.exe" from the zip to c:\Selenium.WebDrivers
5. Firefox
   1. <https://github.com/mozilla/geckodriver/releases>
   2. Under the latest release v#.##.# under "Assets" click the geckodriver-\*-win64.zip
   3. Extract "geckodriver.exe" from the zip to c:\Selenium.WebDrivers

# Create the Application

1. Create a new project > MSTest Test Project (.NET Core)
2. Install the following NuGet packages
   1. Selenium.Support by Selenium Committers
   2. Selenium.WebDriver by Selenium Committers



# Resources

1. [Introduction To Selenium Webdriver With C# In Visual Studio 2015](https://www.c-sharpcorner.com/UploadFile/093731/introduction-to-selenium-webdriver-with-C-Sharp-in-visual-studio/)
2. [Migrating A Selenium Project From .NET Framework To .NET Core](https://medium.com/maestral-solutions/migrating-selenium-project-from-net-framework-to-net-core-32a56589fe7c)

# Dog Food Conference

# Conclusion