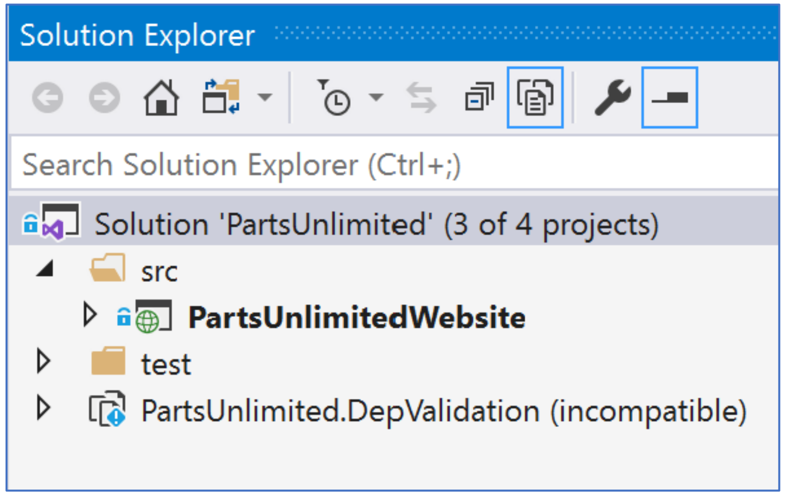
Lab: User Interface Testing - Selenium

May 2020

Let's create a User Interface project

1. **Note:** to run this lab Firefox needs to be installed in the local environment.
2. Open **PartsUnlimited** solution in Visual Studio 2019 (the project used for manual testing lab). See **image 1.**
3. In Solution Explorer, right click **PartsUnlimited > Add New Project…**
4. Select **MsTest Test Project C# (.net core).** See **image 2.**
5. Enter project name **PartsUnlimited.SeleniumTests.** Click **Create**
6. In solution explorer, remove the file UnitTest1.cs

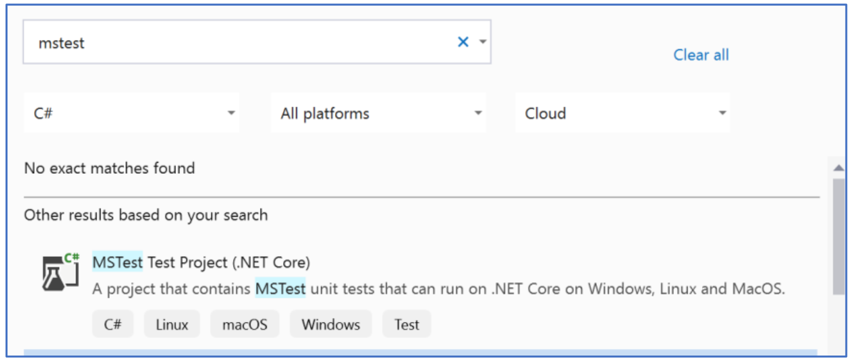


Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'PartsUnlimited' (3 of 4 projects)
src
PartsUnlimitedWebsite
test
PartsUnlimited.DepValidation (incompatible)

Image 1



mstest

Clear all

C#
All platforms
Cloud

No exact matches found

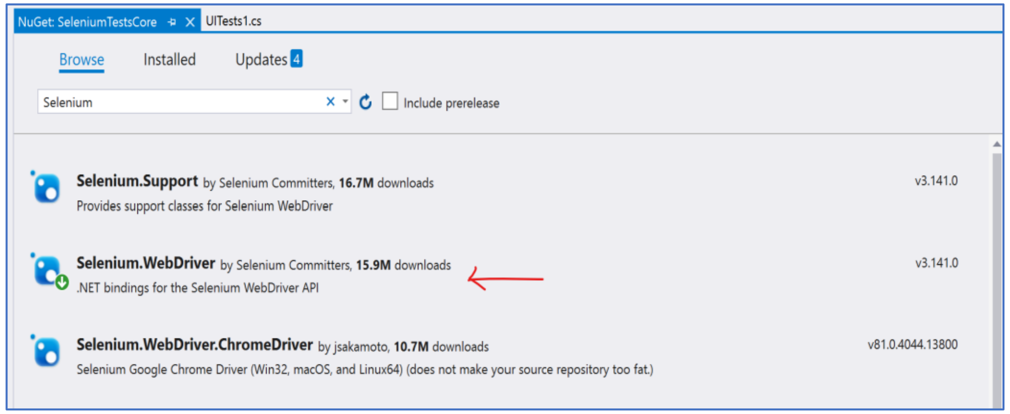
Other results based on your search

MSTest Test Project (.NET Core)
A project that contains MSTest unit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Test

Image 2

Let’s add Selenium nuget package

7. Under **PartsUnlimited.SeleniumTests,** right click **Dependencies.** Select **Manage Nuget Packages…**
8. In **Nuget Package Manager**, select **Browse** tab. Search for **Selenium.WebDriver.**
9. Click **Selenium.WebDriver** item. Click **Install** button to add package to the project.



Selenium

Selenium.Support by Selenium Committers, **16.7M** downloads v3.141.0
Provides support classes for Selenium WebDriver

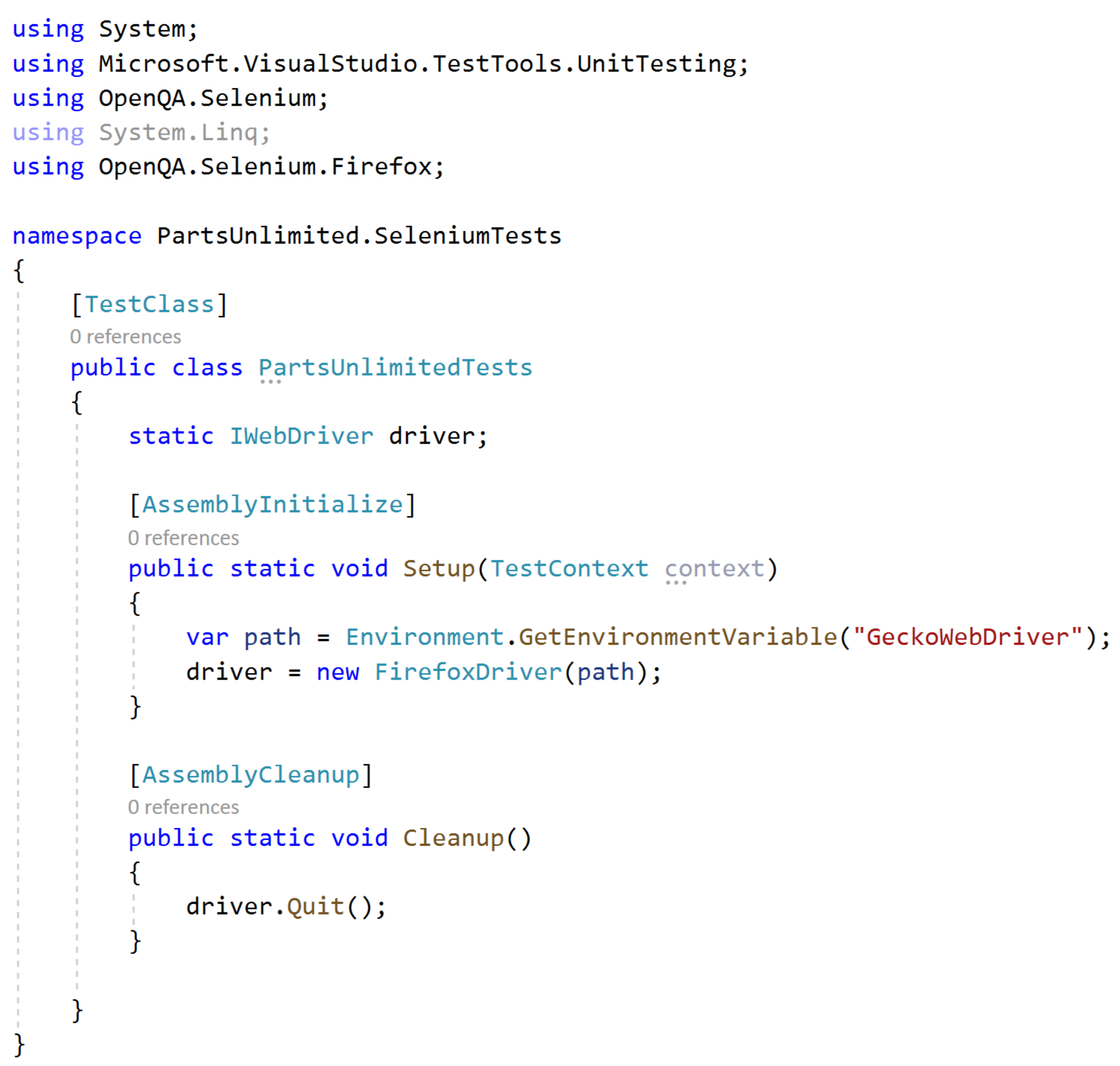
Selenium.WebDriver by Selenium Committers, **15.9M** downloads v3.141.0
.NET bindings for the Selenium WebDriver API

Selenium.WebDriver.ChromeDriver by jsakamoto, **10.7M** downloads v81.0.4044.13800
Selenium Google Chrome Driver (Win32, macOS, and Linux64) (does not make your source repository too fat.)

Image 3

Let’s add user interface tests

10. In solution explorer, right click **PartsUnlimited.SeleniumTests.** Select **Add > Class…** Name **file PartsUnlimitedTests.cs.**
11. Replace the content of the file with snippet 1 (snippets are located at workshop-testing\src\Snippets\selenium)
12. Look at Setup method. Firefox driver path is passed as a parameter for Firefox driver. The path is stored in an environment variable.

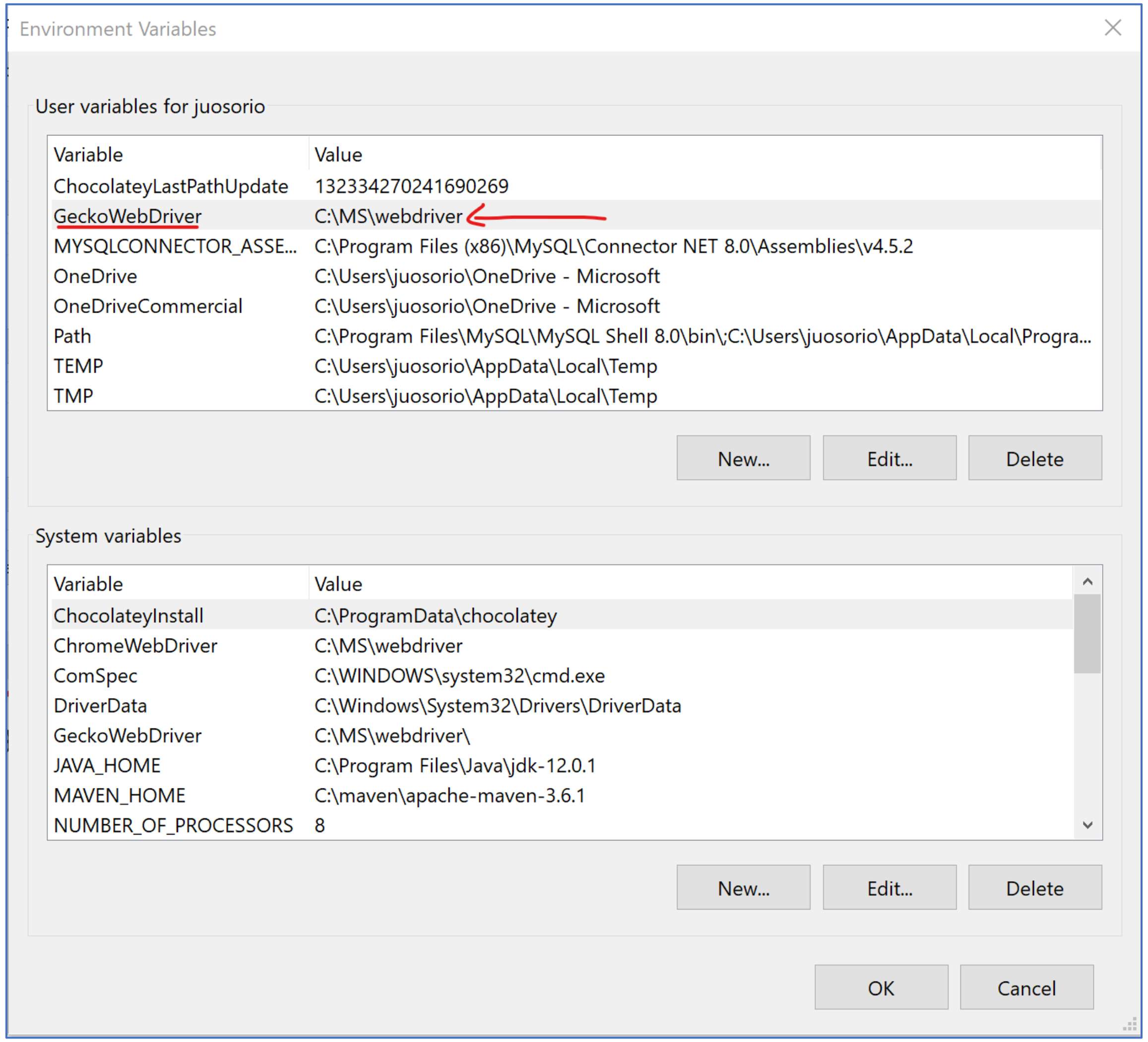


```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OpenQA.Selenium;
using System.Linq;
using OpenQA.Selenium.Firefox;

namespace PartsUnlimited.SeleniumTests
{
    [TestClass]
    0 references
    public class PartsUnlimitedTests
    {
        static IWebDriver driver;

        [AssemblyInitialize]
        0 references
        public static void Setup(TestContext context)
        {
            var path = Environment.GetEnvironmentVariable("GeckoWebDriver");
            driver = new FirefoxDriver(path);
        }

        [AssemblyCleanup]
        0 references
        public static void Cleanup()
        {
            driver.Quit();
        }

    }
}
```

snippet 1

Let’s add environment variable

13. Create an environment variable with the name **GeckoWebDriver**. The path correspond to a folder where the web driver for Firefox exists in the system.
14. To add the driver to your local system, use the file **geckodriver.exe** in workshop-testing\src\driver and copy it to the path used for the environment variable.
15. See image 4



User variables for juosorio

Variable	Value
ChocolateyLastPathUpdate	132334270241690269
GeckoWebDriver	C:\MS\webdriver
MYSQLCONNECTOR_ASSE...	C:\Program Files (x86)\MySQL\Connector NET 8.0\Assemblies\v4.5.2
OneDrive	C:\Users\juosorio\OneDrive - Microsoft
OneDriveCommercial	C:\Users\juosorio\OneDrive - Microsoft
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\juosorio\AppData\Local\Progra...
TEMP	C:\Users\juosorio\AppData\Local\Temp
TMP	C:\Users\juosorio\AppData\Local\Temp

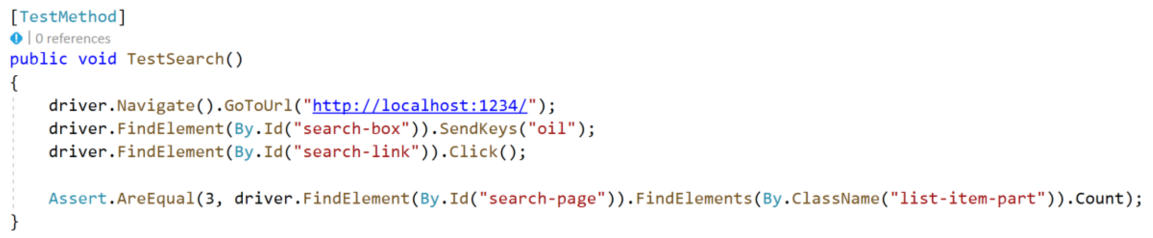
System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ChromeWebDriver	C:\MS\webdriver
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
GeckoWebDriver	C:\MS\webdriver\
JAVA_HOME	C:\Program Files\Java\jdk-12.0.1
MAVEN_HOME	C:\maven\apache-maven-3.6.1
NUMBER_OF_PROCESSORS	8

Image 4

Let’s add user interface tests

16. In Visual Studio, open file **PartsUnlimitedTests.cs.** Add a method to the class using snippet 2.
17. This test method navigates to the local instance of PartsUnlimited web site, enters oil for search term and clicks to run a search. It verifies that 3 elements are found.
18. Replace http://localhost:1234/ with the address of your local instance for the project PartsUnlimited and save.
19. Build the solution in Visual Studio.
20. Open Test Explorer. Select the test **TestSearch** and run it. Make sure Partsunlimited website is running before executing the test.
21. Verify TestSearch is successful
22. To see the execution step by step, add a break point at the beginning of the method. Right click inside the method, select **Debug Test.** Move step by step with F11 key.

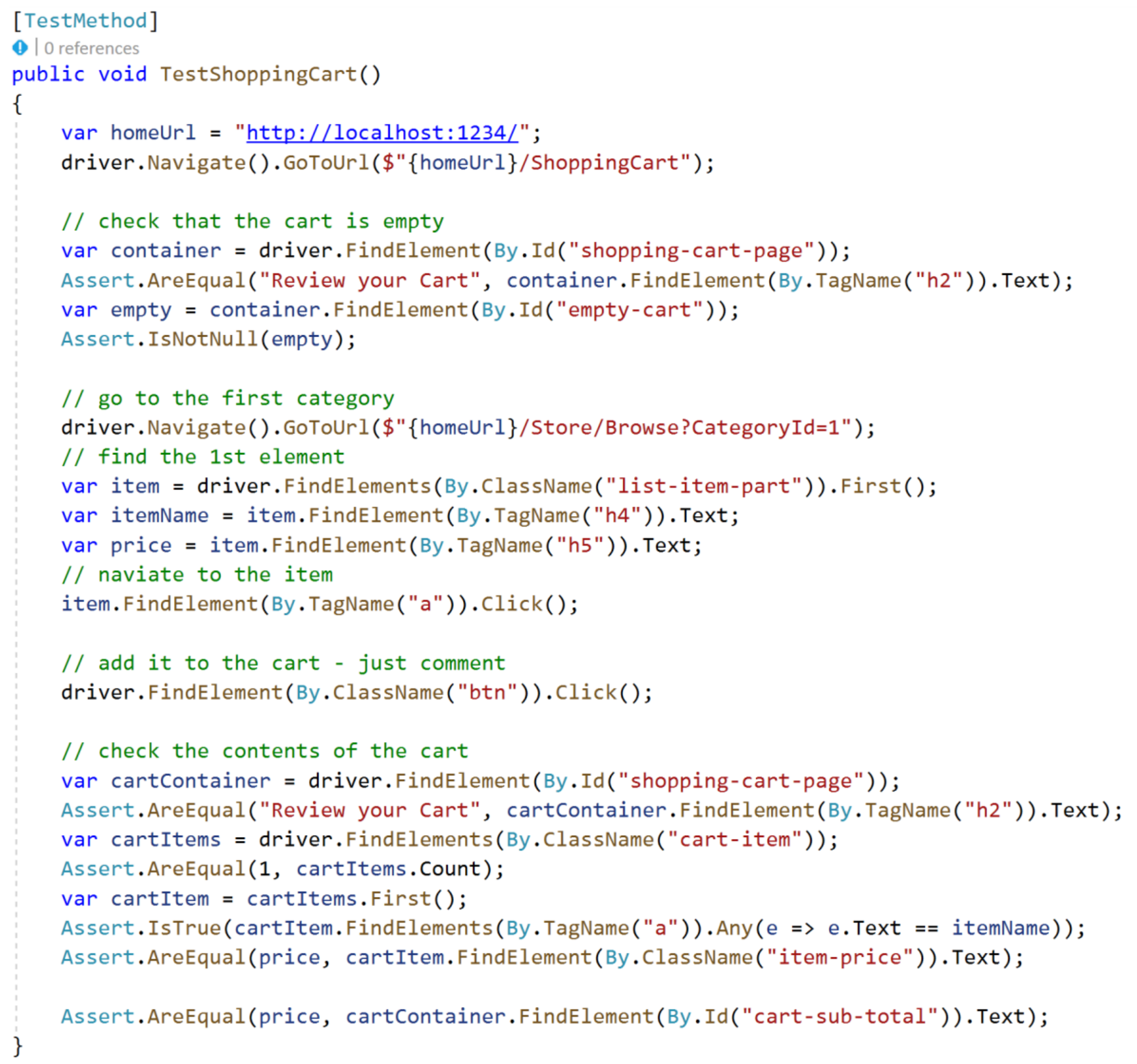


```
[TestMethod]
| 0 references
public void TestSearch()
{
    driver.Navigate().GoToUrl("http://localhost:1234/");
    driver.FindElement(By.Id("search-box")).SendKeys("oil");
    driver.FindElement(By.Id("search-link")).Click();

    Assert.AreEqual(3, driver.FindElement(By.Id("search-page")).FindElements(By.ClassName("list-item-part")).Count);
}
```

Snippet 2

Let’s add more user interface tests

23. Add another method to the class using snippet 3.
24. This new method test the shopping cart functionality.
25. Build the solution. Verify the new test shows up in **Test Explorer.**
26. Run the test step by step to see the execution in Firefox.



```
[TestMethod]
0 references
public void TestShoppingCart()
{
    var homeUrl = "http://localhost:1234/";
    driver.Navigate().GoToUrl($"{homeUrl}/ShoppingCart");

    // check that the cart is empty
    var container = driver.FindElement(By.Id("shopping-cart-page"));
    Assert.AreEqual("Review your Cart", container.FindElement(By.TagName("h2")).Text);
    var empty = container.FindElement(By.Id("empty-cart"));
    Assert.IsNotNull(empty);

    // go to the first category
    driver.Navigate().GoToUrl($"{homeUrl}/Store/Browse?CategoryId=1");
    // find the 1st element
    var item = driver.FindElements(By.ClassName("list-item-part")).First();
    var itemName = item.FindElement(By.TagName("h4")).Text;
    var price = item.FindElement(By.TagName("h5")).Text;
    // naviate to the item
    item.FindElement(By.TagName("a")).Click();

    // add it to the cart - just comment
    driver.FindElement(By.ClassName("btn")).Click();

    // check the contents of the cart
    var cartContainer = driver.FindElement(By.Id("shopping-cart-page"));
    Assert.AreEqual("Review your Cart", cartContainer.FindElement(By.TagName("h2")).Text);
    var cartItems = driver.FindElements(By.ClassName("cart-item"));
    Assert.AreEqual(1, cartItems.Count);
    var cartItem = cartItems.First();
    Assert.IsTrue(cartItem.FindElements(By.TagName("a")).Any(e => e.Text == itemName));
    Assert.AreEqual(price, cartItem.FindElement(By.ClassName("item-price")).Text);

    Assert.AreEqual(price, cartContainer.FindElement(By.Id("cart-sub-total")).Text);
}
```

Snippet 2