

MS Visual Studio Setup -1

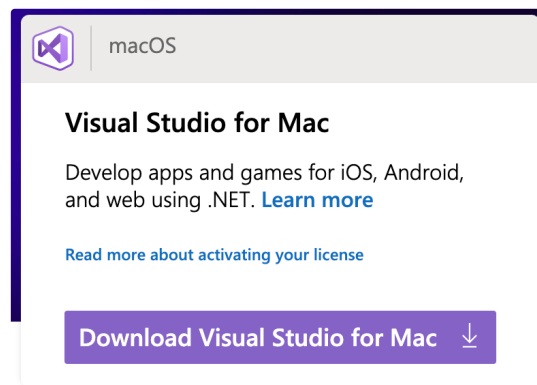
- [Getting Started](#)
- [What are the tools required to install?](#)
- [Setup Guides](#)

Getting Started

Tech Stacks	Description
Selenium	Main Test Driver
NUnit	C# Test Framework
DotNet	C# Platform

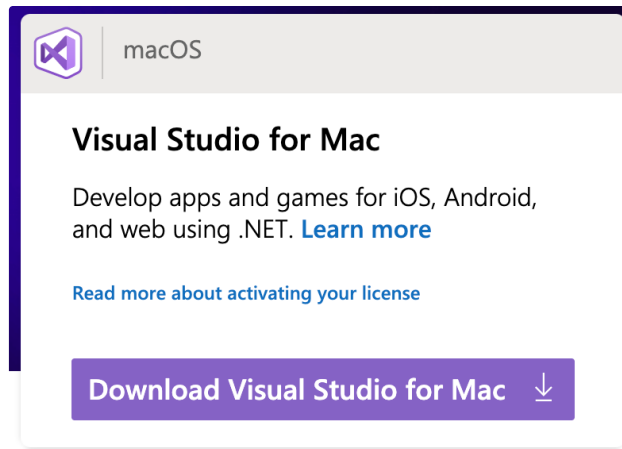
What are the tools required to install?

We need to install only Microsoft Visual Studio to develop the test script using C# with NUnit Test Framework and Selenium WebDriver. These packages are called NuGet Packages and can be installed directly as Visual Studio has the built in package managers. Download the Visual Studio installer (Community version) for MacOS/Windows here : [Visual Studio: IDE and Code Editor for Software Developers and Teams](#)

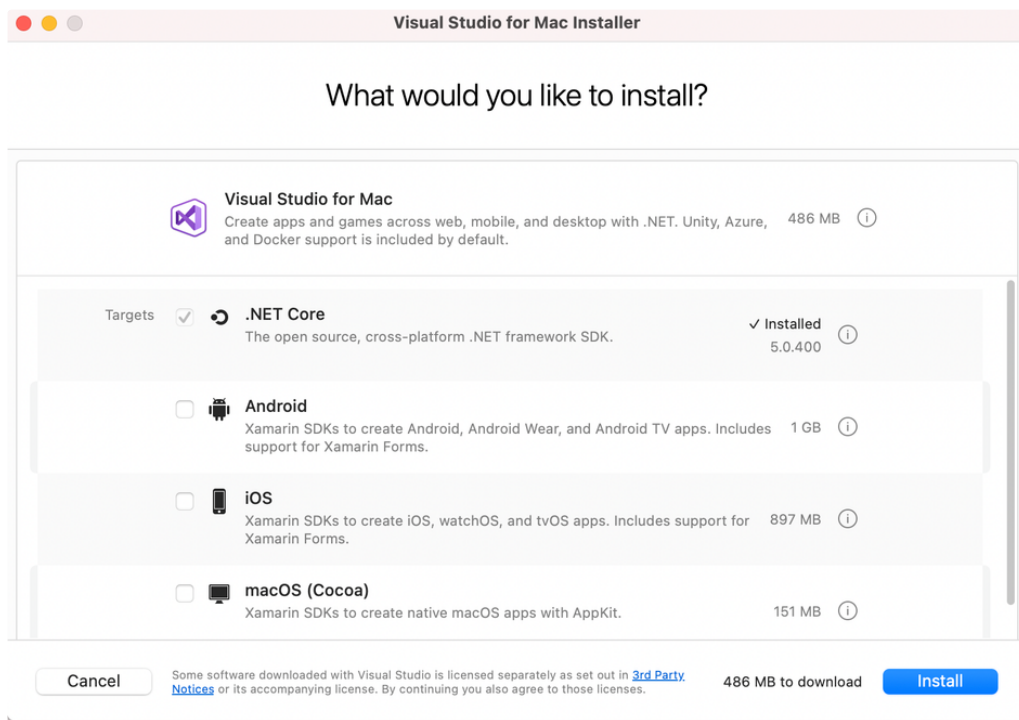


Setup Guides

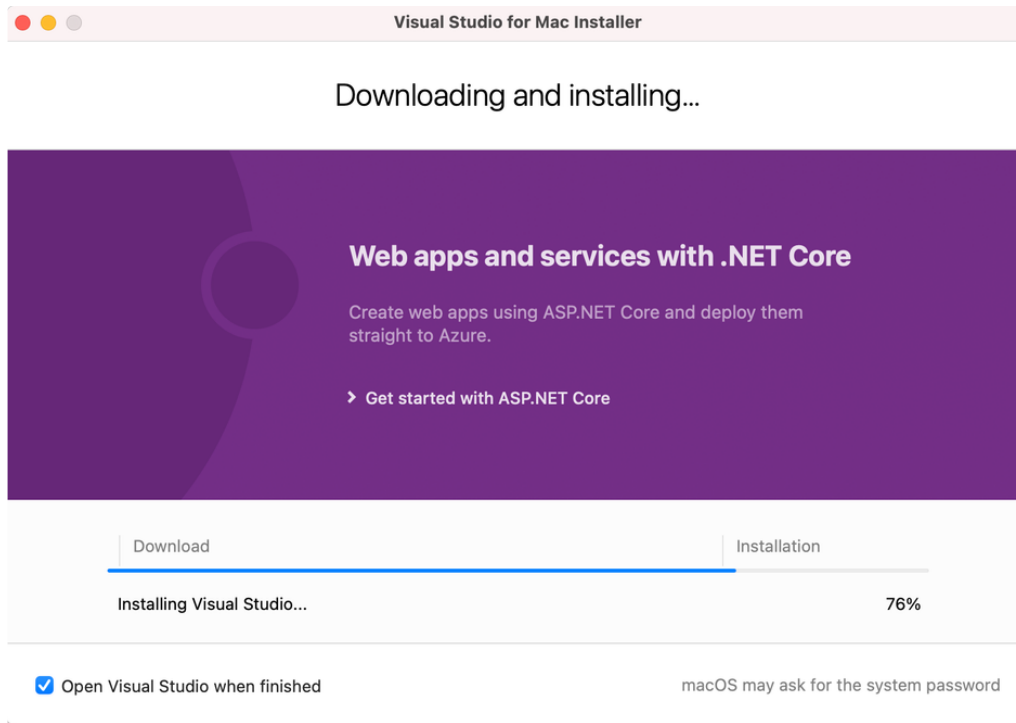
1. Run the Visual Studio installer. Click install.



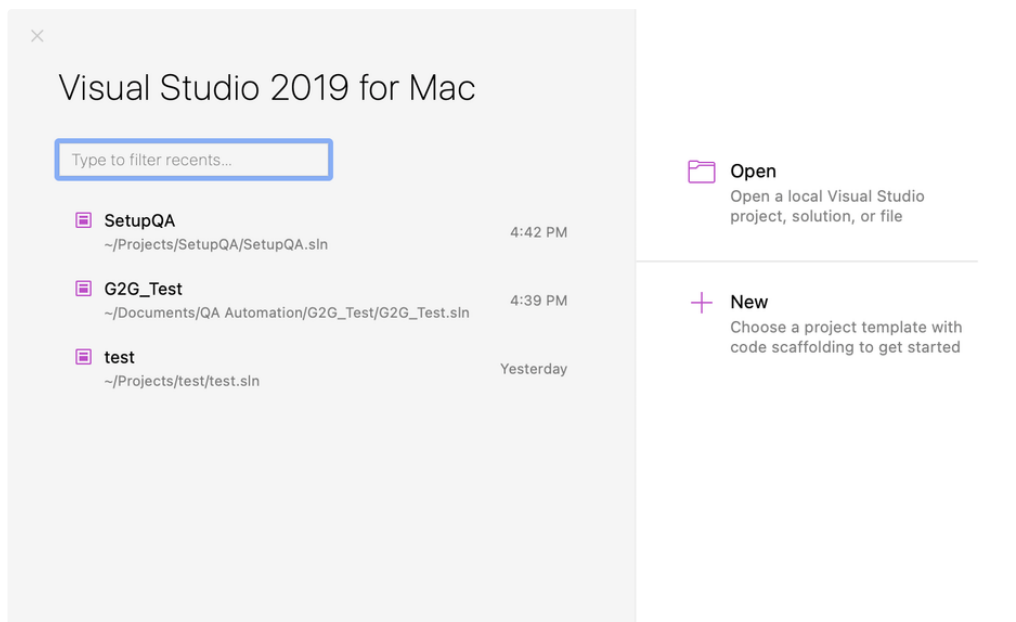
2. Choose .Net Core option, as the framework to be used.



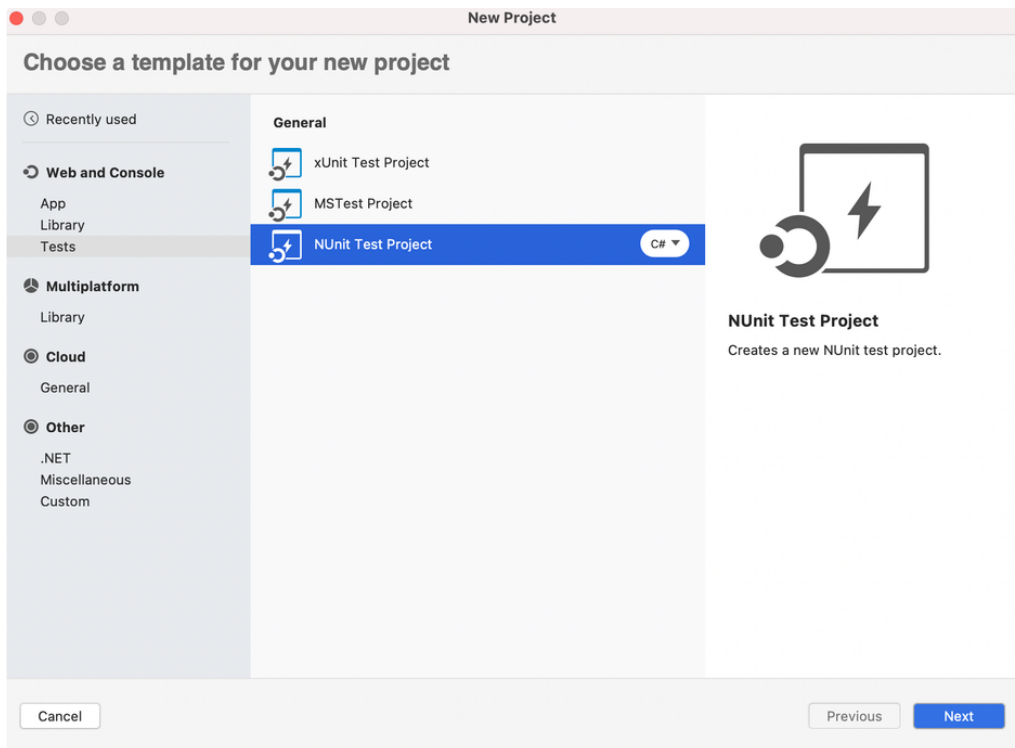
3. The installation will start and allow any prompt along the way that is asking for permission.



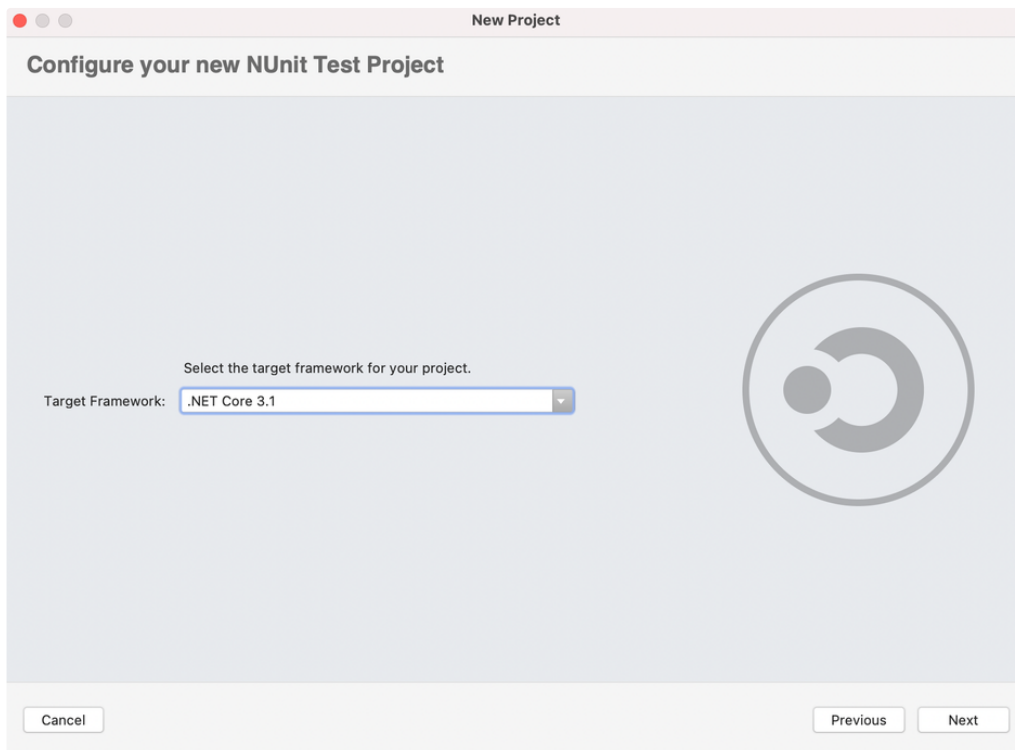
4. Once installation finished, you may create a new C# project. Please select the type of project as 'NUnit Test Project', since we will be developing Selenium test script bind with C# and NUnit Test Framework.



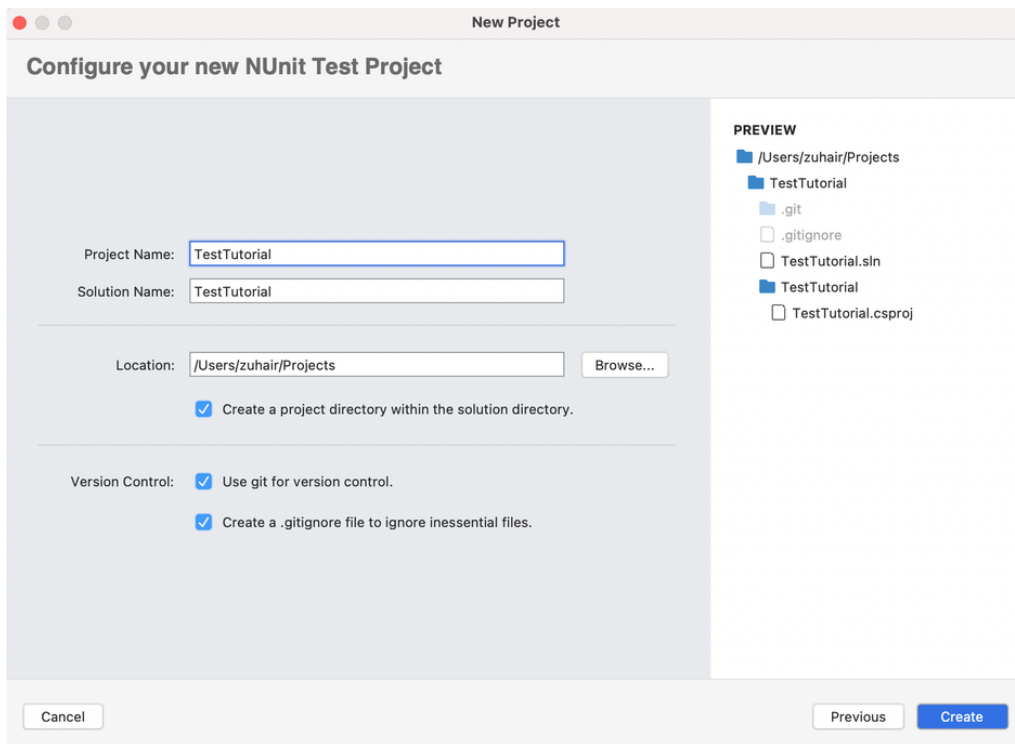
Click new to start a project



Select NUnit Test Project

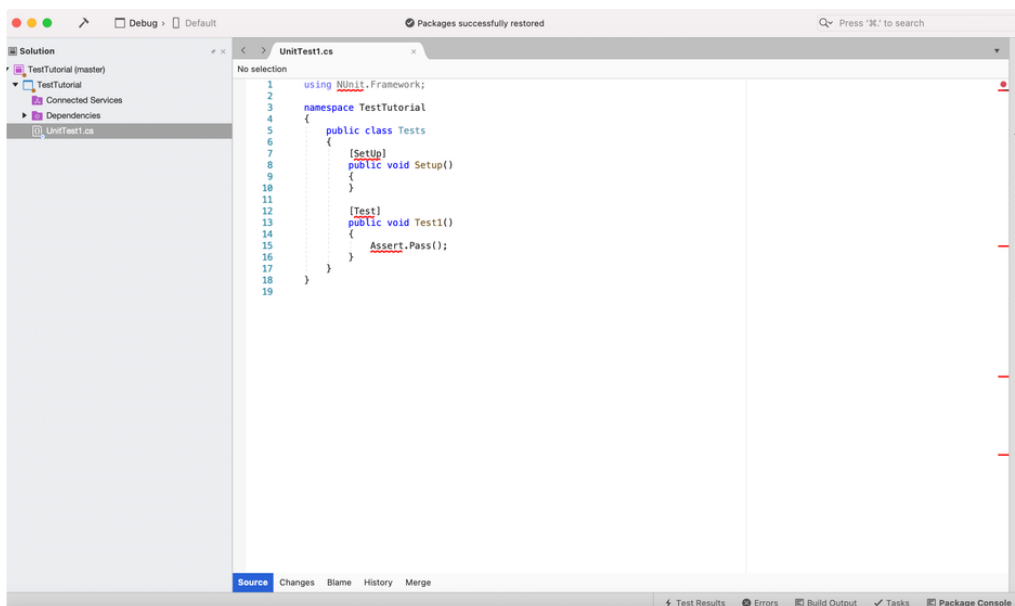


Select .Net Core 3.1

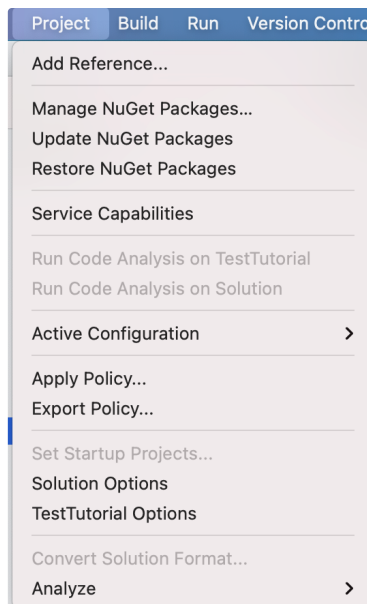


Enter Project name

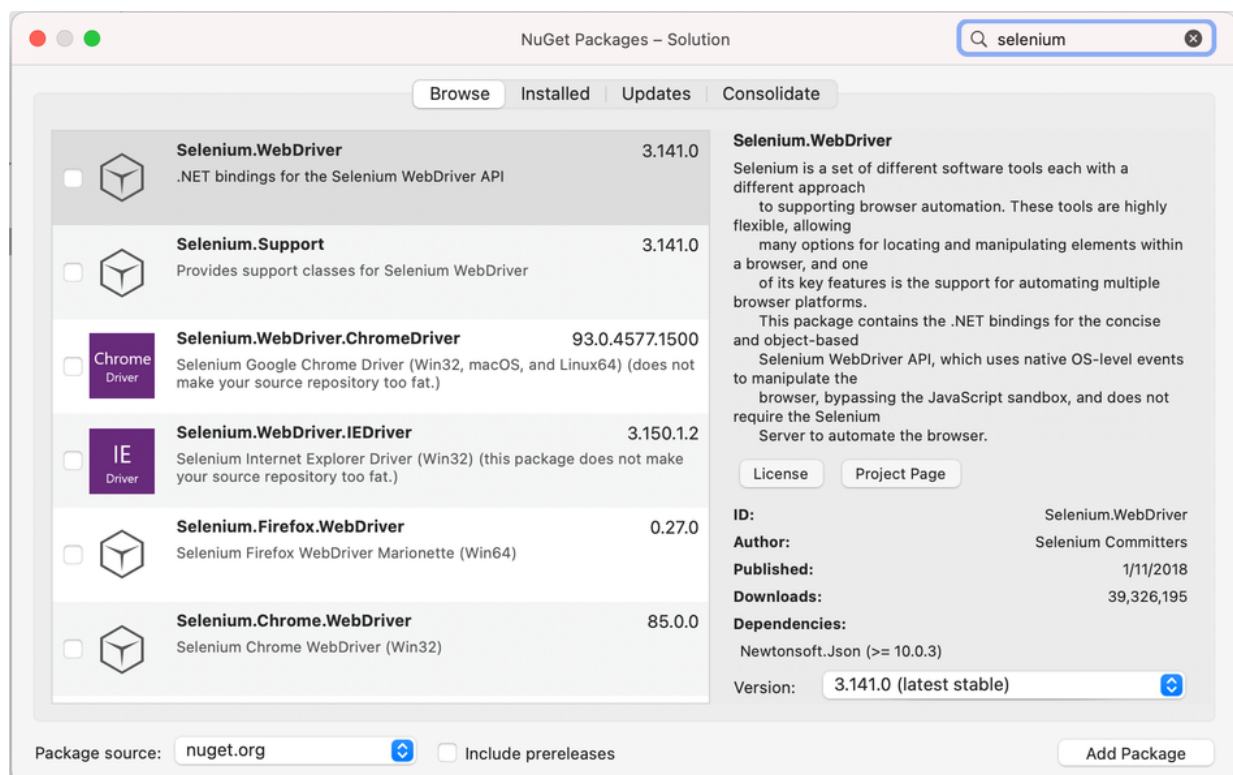
5. A new project will be created, and you are good to write the test script. If you already have an existing project, you may open it directly.



6. Now we need to add the required packages to run the Selenium test script. Go to 'Project' and select 'Manage NuGet Packages' to add the required packages.



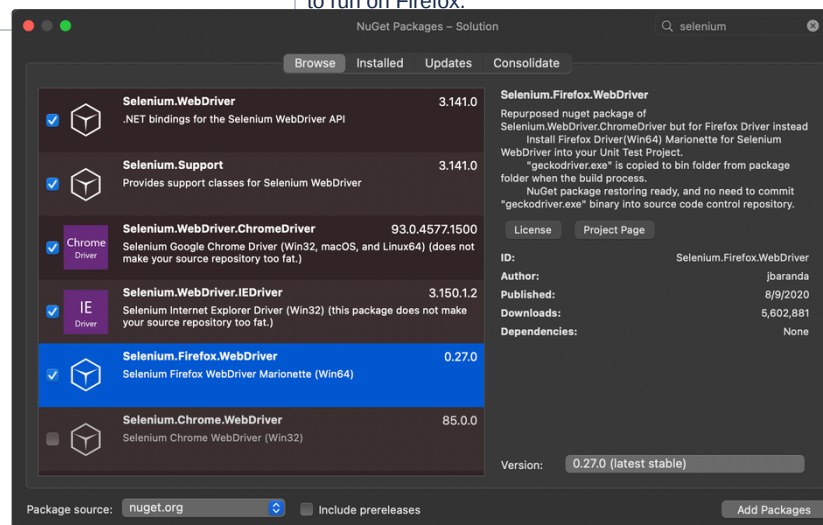
Select 'Manage NuGet Packages'



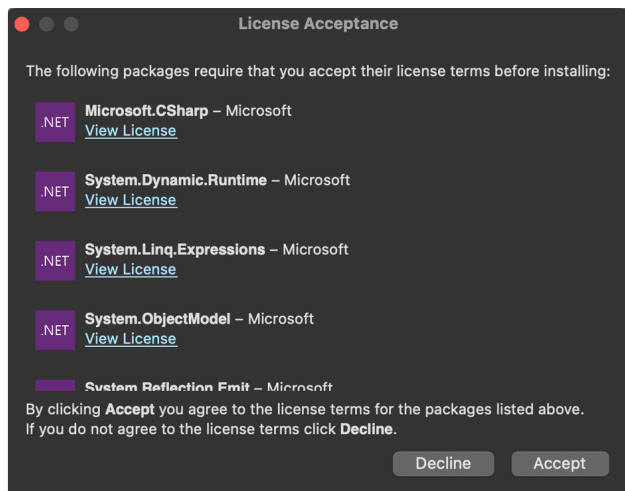
Search Selenium keyword in the search input.

NuGet Packages	Description
Selenium.WebDriver	Contains the .NET bindings for the concise and object-based Selenium WebDriver API, which uses native OS-level events to manipulate the browser, bypassing the JavaScript sandbox, and does not require the Selenium Server to automate the browser
Selenium.Support	Contains .NET support utilities and classes that users may find useful in using Selenium WebDriver. These support classes are mainly intended to spark ideas of

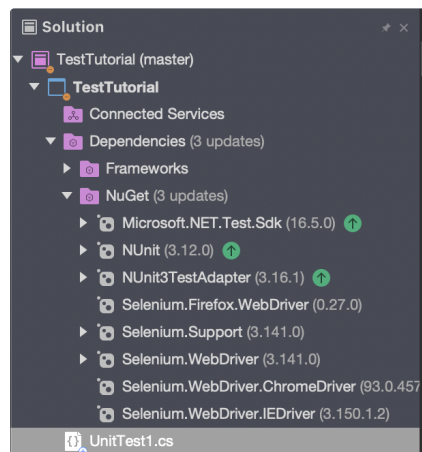
	what is possible with Selenium WebDriver, and may not be entirely appropriate for production use..
Selenium.WebDriver.ChromeDriver	Install Chrome Driver (Win32, macOS, and Linux64) for Selenium WebDriver into your Unit Test Project. "chromedriver(.exe)" is copied to the bin folder from the package folder when the build process. NuGet package restoring ready, and no need to commit "chromedriver(.exe)" binary into source code control repository. Adding these package will automatically add the "chromedriver(.exe)" needed to launch Chrome used for testing. Required for test to run on Google Chrome.
Selenium.WebDriver.IEDriver	Install IE Driver(x86) for Selenium WebDriver into your Unit Test Project. "IEDriverServer.exe" is copied to bin folder from package folder when the build process. NuGet package restoring ready, and no need to commit "IEDriverServer.exe" binary into source code control repository. Adding these package will automatically add the "IEDriverServer.exe" needed to launch IE used for testing. Required for test to run on Internet Explorer.
Selenium.FirefoxDriver.WebDriver	Repurposed nuget package of Selenium.WebDriver.ChromeDriver but for Firefox Driver instead Install Firefox Driver(Win64) Marionette for Selenium WebDriver into your Unit Test Project. "geckodriver.exe" is copied to bin folder from package folder when the build process. NuGet package restoring ready, and no need to commit "geckodriver.exe" binary into source code control repository. Adding these package will automatically add the "geckodriver.exe" needed to launch Firefox used for testing. Required for test to run on Firefox.



Select 'Selenium.WebDriver', 'Selenium.Support', 'Selenium.WebDriver.ChromeDriver', 'Selenium.WebDriver.IEDriver', 'Selenium.Firefox.WebDriver' packages and click add packages



Accept the license acceptance display shown



Once the packages have been added, you can see the packages listed under the NuGet directory in the project explorer

7. To use the added packages in the script, you need to import the packages into the script.

```
1 using System;
2 using OpenQA.Selenium;
3 using OpenQA.Selenium.Chrome;
4 using OpenQA.Selenium.Safari;
5 using OpenQA.Selenium.IE;
6 using OpenQA.Selenium.Firefox;
7 using OpenQA.Selenium.Edge;
8 using OpenQA.Selenium.Opera;
9 using OpenQA.Selenium.Support;
10 using System.Xml;
11 using System.IO;
12 using NUnit;
13 using NUnit.Framework;
```



```
UnitTest1.cs
No selection

1  using System;
2  using OpenQA.Selenium;
3  using OpenQA.Selenium.Chrome;
4  using OpenQA.Selenium.Safari;
5  using OpenQA.Selenium.IE;
6  using OpenQA.Selenium.Firefox;
7  using OpenQA.Selenium.Edge;
8  using OpenQA.Selenium.Opera;
9  using OpenQA.Selenium.Support;
10 using System.Xml;
11 using System.IO;
12 using NUnit;
13 using NUnit.Framework;
14
15 namespace Test
16 {
17     public class Tests
18     {
19         public static IWebDriver driverChrome = new ChromeDriver();
20         public static IWebDriver driverSafari = new SafariDriver();
21         public static IWebDriver driverIE = new InternetExplorerDriver();
22         public static IWebDriver driverFirefox = new FirefoxDriver();
23         public static IWebDriver driverEdge = new EdgeDriver();
24         public static IWebDriver driverOpera = new OperaDriver();
25         [SetUp]
26         public void Setup()
27         {
28         }
29
30         [Test]
31         public void Test1()
32         {
33             Assert.Pass();
34         }
35     }
36 }
37
38
39
```

Importing 'Selenium.Chrome', up until 'Selenium.Opera', from line 3-8 in the image above depends on your test browser requirement. If only certain browser required for test run, you did not to import all the browser driver packages. Opera and Safari drivers are embedded together in the main Selenium.WebDriver package, so there is no additional packages required to be add for both.