**Selenium**

**Q. What is Selenium?**

Selenium is a set of web drivers which will help to automate web testing.

**Q. How will you find an element using Selenium?**

In Selenium, every object or control in a web page is referred as an element, there are different ways to find an element in a web page they are

* By.id()
* By.name()
* By.tagName()
* By.className()
* By.linkText()
* By.partialLinkText()
* By.xpath()
* By.cssSelector()

**Example:**

* IWebElement GetByClassName = driver.FindElement(By.ClassName(“ClassElementName”));
* IWebElement GetByElementID = driver.FindElement(By.Id(“ElementID”));

**Q. What is the use of XPath in Selenium?**

In Selenium, XPath can be used to find element on the page. It is also useful in finding dynamic elements.

**Q. Explain the difference between single and double slash in X-path?**

Single slash ‘/ ’

* Single slash ( / ) start selection from the document node
* It allows you to create ‘absolute’ path expressions

Double Slash ‘// ’

* Double slash ( // ) start selection matching anywhere in the document
* It enables to create ‘relative’ path expressions

**Q. What is the difference between verify and assert commands?**

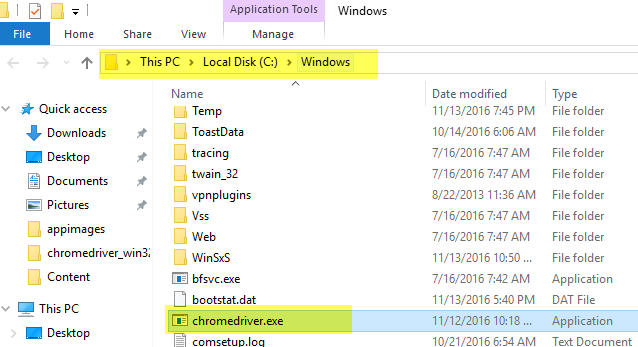
* **Assert:**  Assert allows to check whether an element is on the page or not. The **test will stop on the step failed**, if the asserted element is not available. In other words, the test will terminated at the point where check fails.
* **Verify:** Verify command will check whether the element is on the page, the test will carry on executing, even when element was not found or test failed.  In verification, all the commands are going to run guaranteed even if any of test fails.

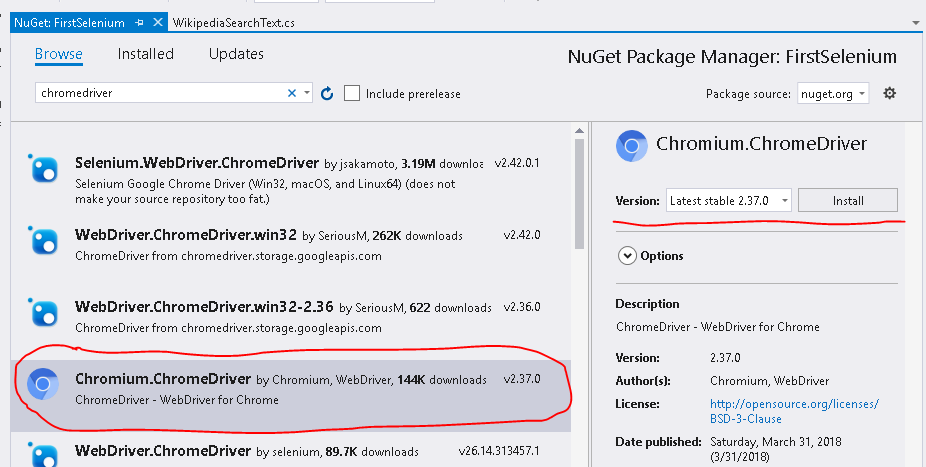
**Practical Explanation of how things works**

Selenium needs browser specific drivers. In order to have them installed, search for “Selenium driver chrome”. It will take you to: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

**Step 1:**  Download the latest chrome driver for selenium as instructed above. <https://chromedriver.storage.googleapis.com/index.html?path=2.25/> Choose chrome driver and download. Example: [chromedriver\_win32.zip](https://chromedriver.storage.googleapis.com/2.25/chromedriver_win32.zip)

**Step 2:** Unzip the chromedriver zip file and copy and paste the content into c:\windows\

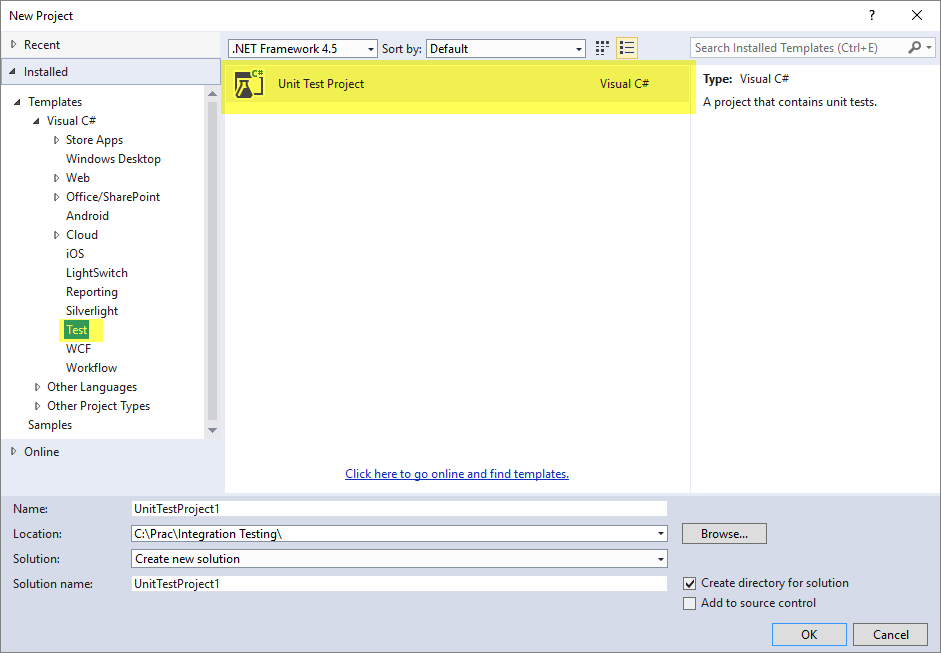




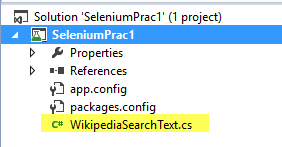
**Install Nuget Packages:**

1. **Selenium.WebDriver**
2. **Chromium.ChromeDriver**

**Step 3:** Create new Unit Test Project

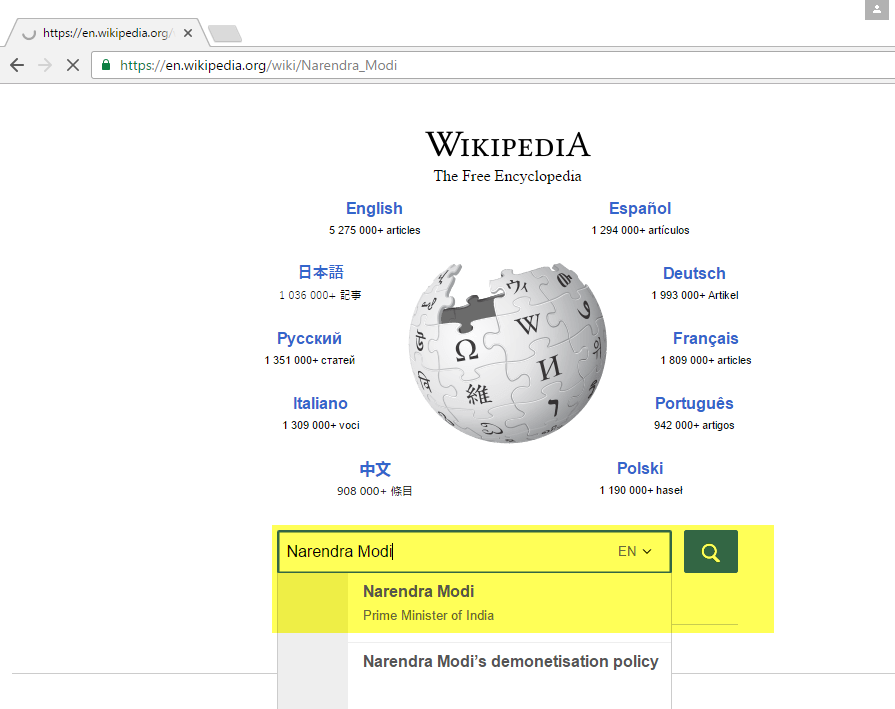


**Step 4:** Create unit test class file WikipediaSearchText.cs



**Step 5:** Write Selenium Code

**Step 6:** VS.Net Menu: Test > Windows > Test Explorer and Run all test





**Whole Code for better understanding:**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

namespace SeleniumPrac1

{

[TestClass]

public class WikipediaSearchText

{

private IWebDriver driver;

//Run 1 time BEFORE each test run

[TestInitialize]

public void TestInit()

{

driver = new ChromeDriver();

driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(5);

driver.Navigate().GoToUrl("https://www.wikipedia.org/");

}

//Run 1 time AFTER each test run

[TestCleanup]

public void TestCleanUp()

{

driver.Quit();

}

[TestMethod]

public void Search\_Find\_ResultPageIsShown()

{

var searchTerm = "Narendra Modi";

SearchFor(searchTerm);

IWebElement firstHeading = GetByElement("firstHeading");

var expected = searchTerm;

var actual = firstHeading.Text;

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void Search\_Find\_ResultPageIsNotFound()

{

SearchFor("abcxyz");

IWebElement createLinkMessage = GetByClassName("mw-search-createlink");

var expected = "The page \"Abcxyz\" does not exist";

//var actual = createLinkMessage.Text;

Assert.IsTrue(createLinkMessage.Text.Contains(expected));

}

#region Helper Methods

private IWebElement GetByClassName(string className)

{

return driver.FindElement(By.ClassName(className));

}

private IWebElement GetByElement(string elementID)

{

return driver.FindElement(By.Id(elementID));

}

private void SearchFor(string searchTerm)

{

IWebElement searchInput = driver.FindElement(By.Id("searchInput"));

searchInput.SendKeys(searchTerm);

searchInput.SendKeys(Keys.Enter);

}

#endregion

}

}