# Testing using Selenium WebDriver and NUnit in C#

- Overview
- Description
- TipsReferences
- References

### Overview

In this tutorial, we will create some UI tests using NUnit and Selenium WebDriver for browser automation.

# Source-code for this tutorial Code is available in GiHub; the repo contains some additional tests beyond the scope of this tutorial and some auxiliary scripts.

## Description

Our target application is a simple website providing a login page that we aim to test using positive and negative test scenarios.

$\leftarrow  ightarrow \mathbf{C}$ $\triangleq$ robotwebdemo.herokuapp.com $\mathbf{e}_{\mathbf{r}}$	
Login Page         Please input your user name and password and click the login button.         User Name: demo         Password:         Image: Login	

We start by creating a #C .NET project, with NUnit and Selenium WebDriver dependencies.

#### nunit\_webdriver\_tests.csproj

```
<Project Sdk="Microsoft.NET.Sdk">
 <PropertyGroup>
   <TargetFramework>net5.0</TargetFramework>
   <IsPackable>false</IsPackable>
 </PropertyGroup>
 <ItemGroup>
   <PackageReference Include="DotNetSeleniumExtras.PageObjects" Version="3.11.0" />
   <PackageReference Include="DotNetSeleniumExtras.PageObjects.Core" Version="3.12.0" />
   <PackageReference Include="NUnit" Version="3.13.1" />
   <PackageReference Include="NUnit.Console" Version="3.12.0" />
   <PackageReference Include="NUnit3TestAdapter" Version="3.16.1">
     <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
     <PrivateAssets>all</PrivateAssets>
   </PackageReference>
   <PackageReference Include="Microsoft.NET.Test.Sdk" Version="16.5.0" />
   <PackageReference Include="Selenium.WebDriver" Version="3.141.0" />
 </ItemGroup>
</Project>
```

Before implementing the tests, we need to choose an approach for abstracting our website.

Interaction with the login page is abstracted using the page objects model. There's a object for the login page itself and another for the page containing the result.

Tests may be annotated with the "Requirement" property, if you wish to link the corresponding Test issue to an existing requirement/story issue in Jira.

The following code snippet shows two test scenarios: one for a successful login and another for an invalid login attempt due to incorrect credentials.

#### WebdemoTests.cs

```
using System;
using NUnit.Framework;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using Webdemo.PageObjects;
namespace SeleniumWebdriver
{
    [TestFixture]
   public class WebdemoTests
    {
        private IWebDriver driver;
        [SetUp]
        public void SetupTest()
        {
            ChromeOptions options = new ChromeOptions();
            options.AddArgument("--no-sandbox"); // Bypass OS security model, to run in Docker
            options.AddArgument("--headless");
            driver = new ChromeDriver(options);
        }
        [TearDown]
        public void TeardownTest()
        {
            try
            {
                driver.Quit();
            }
            catch (Exception)
            {
                \ensuremath{{\prime}}\xspace // Ignore errors if unable to close the browser
            }
        }
        [Test, Property("Requirement", "CALC-2")]
        public void ValidLogin()
        {
            LoginPage loginPage = new LoginPage(driver).Open();
            LoginResultsPage loginResultsPage = loginPage.Login("demo", "mode");
            Assert.AreEqual(loginResultsPage.Title, "Welcome Page");
            Assert.IsTrue(loginResultsPage.Contains("Login succeeded"));
        }
        [Test, Property("Requirement", "CALC-2")]
        public void InvalidLogin()
        {
            LoginPage loginPage = new LoginPage(driver).Open();
            LoginResultsPage loginResultsPage = loginPage.Login("demo", "invalid");
            Assert.AreEqual(loginResultsPage.Title, "Error Page");
            Assert.IsTrue(loginResultsPage.Contains("Login failed"));
        }
    }
}
```

Running the tests can be done using dotnet utility.

example of a Bash script to run the tests
dotnet test -s nunit.runsettings --filter WebdemoTests

We can specify a configuration file to fine-tune NUnit behaviour, such as the output directory for the automation results report.

```
nunit.runsettings

</pre
```

After successfully running the Test Case and generating the NUnit XML report (e.g., nunit\_webdriver\_tests.xml), it can be imported to Xray via a CI tool (e. g. Jenkins), or the REST API, or by using the **Import Execution Results** action within the Test Execution.





A Test Execution issue will be created in this case. The first time results are imported, Test issues are autoprovisioned; one per each NUnit test; in subsequent imports, results (i.e. Test Runs) are created for the already existing Tests so that these are reused and not duplicated.

Each NUnit's test is mapped to a Generic Test in Jira, and the Generic Test Definition field contains the name of the namespace, class, and the method name that implements the test.

In the details of execution screen, we can see the Test Run, showing the overall result and also the duration of the Test.

CALC / Test Execution: CALC-14 / Test: CALC-10 ValidLogin		Export Test a	Return to Test Execution	Execute with Exploratory Ap	Previous
Execution Status PASS er Finished On: 25/May/2110:00 AM			Assig Executed T environme	nee: Xpand IT Admin I By: Xpand IT Admin ests - nts:	Versions: - Revision: -
Comment Preview Comment 🗸	Execution Defects (0) Create Defect Create Sub	-Task   Add Defects   🗸	Execution Evidence (0)		Add Evidence 🗸
Execution Details					
Test Description					^
None					
Test Issue Links (1)					^
tests CALC-2 As a user, I can login the website				0	TO DO
Custom Fields					~
There are no Test Run Custom Fields defined.					
Test Details					^
Test Type: Generic Definition: SeleniumWebdriver.WebdemoTests.ValidLogin					
Results					^
Context	Output			Duration Sta	itus
TestCase 0-1004 - ValidLogin	-			795.851 ms PA	ISS

In this case, we can also see that the Test was linked automatically to the existing user story (i.e. CALC-2).

Therefore, in the Story issue screen we can track the impacts of the test results on the calculated coverage, which in this case shows our Story as being "OK" due to the passing tests.

0	As a user, I can login the website											
<b>/</b> E	Edit	Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin	~			
∽ De	etails											
Ту	rpe:		Story						Status:	TO DO (View Workflow)		
Pri	iority:		O Trivial						Resolution:	Unresolved		
La	ideis: equirem	ent Status:	None	ж								
	quirei											
∽ De	escript	ion										
Cli	ick to a	dd descriptior	n									
✓ Te	st Cov	erage										
										Create Test	Create Sub-Test Execution	+ Link Y
1	TEST CO	VERAGE FOR THE	FOLLOWING AN	ALYSIS SCOPI	E							
	Scop	e: Version; Ve	rsion: None - I	atest execu	ution; Env	vironment: All	Environm	ients 👻			-	ок
	Ŧ	Filter(s)										
	<b>·</b>										Show 10 ✔ entries	Columns 🗸
	÷	P 🔶 S	tatus	🔶 Reso	olution		🔺 Key		Summary	Test Runs	🔶 Test Status	
(		) тс	DO	Unres	olved		CALC-1	0	ValidLogin	≣0	PASS	
(		) TC	DO	Unres	olved		CALC-1	2	InvalidLogin	≣0	PASS	
Sh	nowing '	to 2 of 2 entrie	es								First Previous	1 Next Last

## Tips

If you're using Visual Studio as your IDE, you need to have some dependencies/packages installed.

- NUnit
- NUnit3TestAdapter

These can be installed from Tools>NuGet Package Manager (using the console or the manager's UI).

🕅 File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) 🔑 TestProject1			Sign in 🗛 — 🗇
🗴 💿 - 💿 🛛 - 🏩 📲 🥙 - 🥂 - 🛛 Debug - Any CPU 🕨 TestProject 1 - 🗮 🔯 -			لاً Live Share الأ
👸 NuGet - Solution + 🗙 nunltrunsettings UnitTestLos		- 0	Solution Explorer 🔷 👻
Browse Installed Updates 2 Consolidate		Manage Packages for Solution	000 # 6-50 <b>% -</b>
			Search Solution Explorer (Ctrl+;) P -
Search (Ctri+L) D C include prerelease		Package source: nuget.org • 😡	Solution 'TestProject1' (1 of 1 project)
Microsoft .NTL First S& by Moreon     The Microsoft Ser building NT test projects.     Work by Could Project Net In Normal     Microsoft is a unit determined from AMM the AMM strengt strengt and and any STO Focus.     Microsoft is a unit determined from the Normal Focus Normal Section Sec	2 2003 2013 2013 2013 2013 2017 2017 2017 2017 2017 2017 2017 2017		Comparison     C
			TesResults     C UnitTestLos

Then you can configure the Test Explorer to run the NUnit tests while at the same time producing a NUnit XML report.

#### nunit.runsettings

```
<?xml version="1.0" encoding="utf-8"?>
<RunSettings>
   <NUnit>
     <TestOutputXml>C:\TestResults</TestOutputXml>
</NUnit>
</RunSettings>
```

The fak View Gk Project Build Debug Test Analyse Tools Learnions Window Help Servit/Cat-C2 /P Techniquet Sprin A - 0 ×								
💿 - 이 🏽 - 🏩 🔐 🦻 - 연 - 🛛 Debug - 🖌 Any CPU - 🕨 TestPri	roject1 - 🙀 🖸 🖉				남 Live Share R			
nunitrunsettings a × UnitTestLes			Close (Shift+Esc)	- 0	Solution Explorer - 0 × 2			
<pre>chul version="1.0" encoding="utf+8"?&gt;</pre>				÷	00000 0-5200 /-			
CRunSettings>					Search Solution Explorer (Ctrl+-)			
<testoutputxml>C:\Users\IEUser\source\repos\TestProject1\Tes</testoutputxml>	stProject1\TestResults <td>&gt;</td> <td></td> <td></td> <td>Solution TestProject1' (1 of 1 project)</td>	>			Solution TestProject1' (1 of 1 project)			
<pre></pre>				· · · · · · · · · · · · · · · · · · ·	<ul> <li>TestProject1</li> </ul>			
				1.7	Dependencies			
				-				
					High Microsoft NETCore App			
					Packages			
					<ul> <li>Microsoft Pet 1 (ent.5dx (16.9.1)</li> <li>M NUmit (3.12.0)</li> </ul>			
					<ul> <li>Wunk3TestAdapter (3.16.1)</li> </ul>			
Text Fundame					TestResults			
lett toporer					p C* Unitiestics			
1 · · · · ·	0	Jearch Test Exp	···· ·					
Test	Duration	Due Teste Aller Build	Auto Detect nunsettings Files					
<ul> <li>C lestrop</li> <li>C Totast</li> </ul>	4Project1 (1) 26 m	Processory Auchitecture for Amy CBU Doole to	SP belect Solution Wide runsettings hie					
	101	Pun Tests in Pavallel	C/Users/EUser/source/repos/TestProject//nunit.runsettings					
		Coloma	Outcomes.					
		When Test in Summers Dans	Q 1 Passed					
		may not al summary vane						
	0	Options						
			4 b					
100 % · O No issues found			> In:1	Ch 1 THES CRUE				
Output	and the second se			+ # × !				
Show output from Tests	5 SI Fa							
Setting Description								
Setting: PrivateoinPath = Setting: TestOutputXml = C:\Users\IEUser\source\repos\TestProjecti\1	TestProject1\TestResults							
Setting: RandomSeed =								
Setting: DefaultTestNamePattern =								
Setting: ShowInternalProperties = False Setting: UseParentFONForParametrizedTests = False								
Setting: UseMUnitIdforTestCaseId = False								
Setting: DumpXmlTestDiscovery = False Setting: DumpXmlTestResults = False								
Setting: Prefilter = False								
Setting: VsiestCategoryType = RunTests by IEnumerable <testcase></testcase>								
Whit Adapter 5.16.1.8: Test execution started								
UseVsKeepEngineRunning: False								
EnableShutdown: True Region calacted tests in Cultimers/TRises).comman	TartBrolact1 bio Dabus patr o Toron	internal dia						
Test Output folder checked/created : C:\Users\IEUser\source\repos\	TestProject1\TestProject1\TestResult	3						
NUnit3TestExecutor converted 1 of 1 NUnit test cases Test results written to Civilians/TElian/Source) result and Products	1) Test@voiect1) TestBesults) TestBoofer	t1.v=)						
I I I I I I I I I I I I I I I I I I I								
Package Manager Console Error List Output					Solution Explorer Git Changes			

### References

- GitHub repository for this tutorial
  https://github.com/nunit/docs/wiki
  http://www.seleniumhq.org/docs/03\_webdriver.jsp
  http://www.dotnetcatch.com/2016/11/23/selenium-with-net-core/
  http://toolsqa.wpengine.com/selenium-webdriver/c-sharp/iwebdriver-browser-commands-in-c-sharp/
  Configure unit tests by using a .runsettings file