Setup testing environments

By: Hieu Nguyen

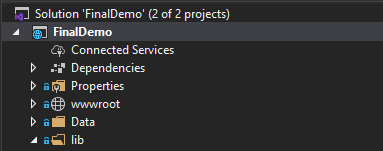
Purpose:

Setting up the testing environment that is compatible with Blazor app.

# Setup

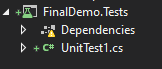
## xUnit

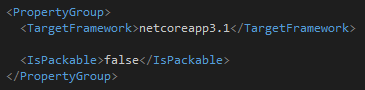
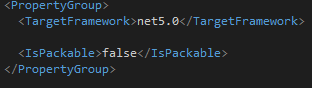
1. Right click on the project Solution -> Add -> New Project
2. Select xUnitTestProject(.NET Core)
3. Select the Project and copy TargetFramework





1. Select the xUnitTest project and replace the TargetFramework to match with your project

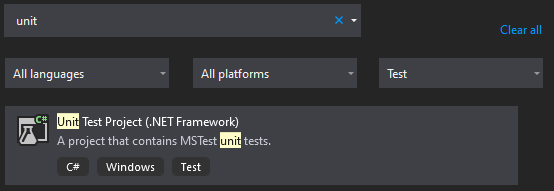




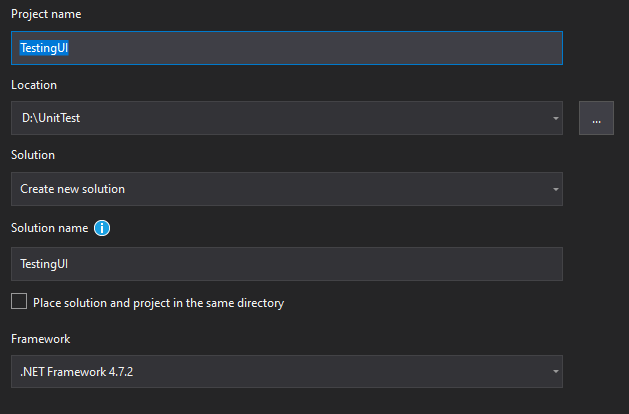
1. Right click xUnitTest project’s Dependencies -> Add Project Reference

## Selenium

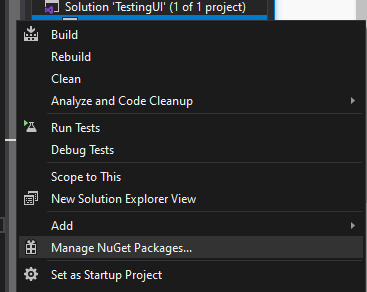
1. Right click on the project Solution -> Add -> New Project



1. Calling it TestingUI



1. Add Nuget packages



In the Browse Tab, search for Selenium, install:

Selenium.WebDriver

Selenium.WebDriver.ChromeDriver

Selenium.WebDriver.IEDriver

Selenium.FireFox.webDriver

1. Create the test

using System;

using System.Text;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Chrome;

using OpenQA.Selenium.IE;

namespace TestingUI

{

[TestClass]

public class MySeleniumTests

{

private Microsoft.VisualStudio.TestTools.UnitTesting.TestContext testContextInstance;

private IWebDriver driver;

private string appURL;

public MySeleniumTests()

{

}

[TestMethod]

[TestCategory("Chrome")]

public void TheBingSearchTest()

{

driver.Navigate().GoToUrl(appURL);

//driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(10);

//Assert.Equals(driver.FindElement(By.Id("title")).GetAttribute("textContent"), "Hello, world!");

//Identify login

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

//operation

loginLink.Click();

//assert

var userInputBox = driver.FindElement(By.Id("UserName"));

var pwdInputBox = driver.FindElement(By.Name("Password"));

Assert.IsTrue(userInputBox.Displayed);

userInputBox.SendKeys("admin");

pwdInputBox.SendKeys("password");

//click on the login

driver.FindElement(By.XPath("//input[@value='Log in']")).Submit();

var lnkEmployeeDetails = driver.FindElement(By.LinkText("Employee Details"));

Assert.IsTrue(lnkEmployeeDetails.Displayed);

}

public TestContext TestContext

{

get

{

return testContextInstance;

}

set

{

testContextInstance = value;

}

}

[TestInitialize()]

public void SetupTest()

{

//appURL = "http://localhost:5001";

appURL = "http://eaapp.somee.com/";

string browser = "chrome";

switch (browser)

{

case "chrome":

driver = new ChromeDriver();

break;

case "FireFox":

driver = new FirefoxDriver();

break;

case "IE":

driver = new InternetExplorerDriver();

break;

default:

driver = new ChromeDriver();

break;

}

}

[TestCleanup()]

public void MytestCleanup()

{

driver.Quit();

}

}

}

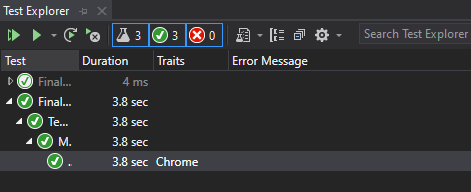
# NOTE

### To run UI test locally

1. Open cmd
2. Navigate to your blazor app’s location
3. Type: dotnet run
4. Run the UI tests

### To run tests:

Click View->Test explorer.



|  |  |
| --- | --- |
| **Error** | **Solution** |
| System.InvalidOperationException : session not created: This version of ChromeDriver only supports Chrome version 88 | Check your Chrome version.  Go to the <https://chromedriver.chromium.org/downloads> and download the correct version  Extract(unzip) the downloaded file.  In the setup, direct which driver you want to use (location of the .exe file you just download)  IWebDriver webDriver = new ChromeDriver(@"D:\Driver\web"); |