To run SpecFlow tests in Visual Studio Code, you need to have the SpecFlow extension installed. Then, you can use the command-line interface to run the tests. Follow these steps:

Add Command

dotnet new console -n SpecFlowOnlineBank

Setting up SpecFlow in Visual Studio Code:

Install the SpecFlow Extension for Visual Studio Code:

Launch Visual Studio Code.

Go to the Extensions view by clicking on the square icon on the Sidebar or pressing Ctrl+Shift+X.

Search for "SpecFlow" and click on "Install".

Initialize SpecFlow:

Open a terminal in Visual Studio Code (Terminal > New Terminal).

Run the following command to initialize SpecFlow:

arduino

Copy code

dotnet new specflow-basic

Install Necessary Packages:

Open the terminal in Visual Studio Code.

Run the following command to install necessary packages:

csharp

Copy code

dotnet add package SpecFlow.NUnit

dotnet add package Selenium.WebDriver

dotnet add package Selenium.WebDriver.ChromeDriver

dotnet restore

Running the SpecFlow Test:

Create the SpecFlow Feature File:

Under the Features directory, create a file named AmazonAddItemToCart.feature with the following content:

gherkin

Copy code

Feature: Add Item to Cart on Amazon

Scenario: Add item to the cart

Given I am on the Amazon home page

When I search for "TP-Link N450 WiFi Router - Wireless Internet Router for Home (TL-WR940N)"

And I click on the first search result

And I add the item to the cart

And I go to the cart

Then I validate the correct item and amount

Create the Step Definitions:

Under the Features directory, create a file named AmazonAddItemToCartSteps.cs with the following content:

csharp

Copy code

using System;

using OpenQA.Selenium;

using OpenQA.Selenium.Chrome;

using TechTalk.SpecFlow;

using NUnit.Framework;

namespace MyTestProject

{

[Binding]

public class AmazonAddItemToCartSteps

{

private IWebDriver driver;

private HomePage homePage;

private ProductPage productPage;

private CartPage cartPage;

[BeforeScenario]

public void Setup()

{

driver = new ChromeDriver();

driver.Manage().Window.Maximize();

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

homePage = new HomePage(driver);

homePage.GoTo();

}

[Given(@"I am on the Amazon home page")]

public void GivenIAmOnTheAmazonHomePage()

{

// This step is already handled in the BeforeScenario hook

}

[When(@"I search for ""(.\*)""")]

public void WhenISearchFor(string itemName)

{

// Search for the item

homePage.SearchItem(itemName);

}

[When(@"I click on the first search result")]

public void WhenIClickOnTheFirstSearchResult()

{

// Click on the first item

productPage = homePage.ClickFirstItem();

}

[When(@"I add the item to the cart")]

public void WhenIAddTheItemToTheCart()

{

// Add the item to the cart

productPage.AddToCart();

}

[When(@"I go to the cart")]

public void WhenIGoToTheCart()

{

// Navigate to the cart

cartPage = productPage.GoToCart();

}

[Then(@"I validate the correct item and amount")]

public void ThenIValidateTheCorrectItemAndAmount()

{

// Validate the correct item and amount

Assert.AreEqual("TP-Link Tri-Band BE9300 WiFi 7 Router Archer BE550 | 6-Stream 9.2Gbps | 𝗙𝘂𝗹𝗹 𝟮.𝟱𝗚 Ports | USB 3.0 | 6 Smart Internal Antennas | VPN Clients & Server | Easy Mesh, HomeS…", cartPage.GetItemName().Trim(), "Incorrect item in the cart.");

Assert.AreEqual("$249.00", cartPage.GetItemPrice().Trim(), "Incorrect item price in the cart.");

}

[AfterScenario]

public void Cleanup()

{

if (driver != null)

{

driver.Quit();

driver.Dispose();

}

}

}

}

Create the Page Objects:

Under the Features directory, create the following files:

HomePage.cs:

csharp

Copy code

using OpenQA.Selenium;

using System;

using System.Threading;

namespace MyTestProject

{

public class HomePage

{

private IWebDriver driver;

private string url = "https://www.amazon.com/s?k=tp-link+n450+wifi+router-wireless+internet+router+for+home+tl-wr940n&crid=3RFUQTWTJRYZP&sprefix=TP-Link+N450+WiFi+Router+-+Wireless+Internet+Router+for+Home+%28TL-WR940N%29%2Caps%2C1278&ref=nb\_sb\_ss\_ts-doa-p\_1\_71";

public HomePage(IWebDriver driver)

{

this.driver = driver;

}

public void GoTo()

{

driver.Navigate().GoToUrl(url);

Thread.Sleep(2000);

}

public void SearchItem(string itemName)

{

// Search for the item

IWebElement searchBox = driver.FindElement(By.Id("twotabsearchtextbox"));

searchBox.SendKeys(itemName);

searchBox.SendKeys(Keys.Return);

Thread.Sleep(3000); // Adding a delay to wait for search results to load

}

public ProductPage ClickFirstItem()

{

// Click on the first item

IWebElement firstResult = driver.FindElement(By.XPath("//span[contains(text(),'Tri-Band BE9300 WiFi 7 Router Archer BE550 | 6-Str')]"));

firstResult.Click();

Thread.Sleep(2000); // Adding a delay to wait for product page to load

return new ProductPage(driver);

}

}

}

ProductPage.cs:

csharp

Copy code

using OpenQA.Selenium;

using System.Threading;

namespace MyTestProject

{

public class ProductPage

{

private IWebDriver driver;

public ProductPage(IWebDriver driver)

{

this.driver = driver;

}

public void AddToCart()

{

// Add the item to the cart

IWebElement addToCartButton = driver.FindElement(By.Id("add-to-cart-button"));

addToCartButton.Click();

Thread.Sleep(2000); // Adding a delay to wait for cart to update

}

public CartPage GoToCart()

{

// Navigate to the cart

IWebElement cart = driver.FindElement(By.Id("nav-cart"));

cart.Click();

Thread.Sleep(9000); // Adding a delay to wait for the cart page to load

return new CartPage(driver);

}

}

}

CartPage.cs:

csharp

Copy code

using OpenQA.Selenium;

namespace MyTestProject

{

public class CartPage

{

private IWebDriver driver;

public CartPage(IWebDriver driver)

{

this.driver = driver;

}

public string GetItemName()

{

// Validate the correct item

IWebElement cartItem = driver.FindElement(By.CssSelector(".sc-product-title"));

return cartItem.Text.Trim();

}

public string GetItemPrice()

{

// Validate the correct item price

IWebElement cartItemPrice = driver.FindElement(By.Id("sc-subtotal-amount-buybox"));

return cartItemPrice.Text.Trim();

}

}

}

Run the Test:

Open a terminal in Visual Studio Code (Terminal > New Terminal).

Run the following command to execute the tests:

bash

Copy code

dotnet test

If everything is set up correctly, this command should compile your project and execute your SpecFlow tests. You will see the test results in the terminal.