using System.Text;

using System.Threading.Tasks;

using TechTalk.SpecFlow;

using System.Configuration;

using OpenQA.Selenium;

using SeleniumExtras.PageObjects;

using TechTalk.SpecFlow.Assist;

using System.Net;

using NUnit.Framework;

using OpenQA.Selenium.Chrome;

using SiteKitTest.Pages;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Interactions;

using Baseclass.Contrib.SpecFlow.Selenium.NUnit.Bindings;

namespace SiteKitTest

{

[Binding]

[TestFixture]

class ContactStep

{

IWebDriver driver = new ChromeDriver();

[SetUp]

[Given(@"I am on homepage")]

public void GivenIAmOnHomepage()

{

driver.Navigate().GoToUrl("https://www.sitekit.net/");

PageActivator.activePage = new ContactPage();

}

[Test]

[When(@"I click on contact")]

public void WhenIClickOnContact()

{

PageActivator.activePage = PageActivator.activePage.As<ContactPage>().contactDetails();

Assert.That(driver.PageSource.Contains("0845 299 0900"), Is.EqualTo(true), "You are not on contact page");

}

[Then(@"I should see the contact details")]

public void ThenIShouldSeeTheContactDetails()

{

var oxford = driver.FindElement(By.Id("Oxford"));

var london = driver.FindElement(By.Id("London"));

Assert.Multiple(() =>

{

Assert.That(oxford.Displayed, Is.Not.Null, "Oxford");

Assert.That(london.Displayed, Is.Not.Null, "London");

});

}

[TearDown]

public void tearDown()

{

driver.Quit();

}

}

}

namespace SiteKitTest

{

[Binding]

[TestFixture]

class ContactStep

{

IWebDriver driver = new ChromeDriver();

// Actions action = new Actions(driver);

// Actions action = new Actions(driver);

// IWebDriver driver = new FirefoxDriver();

// ContactPage page = new ContactPage();

[SetUp]

[Given(@"I am on homepage")]

public void GivenIAmOnHomepage()

{

// Browser.Current.Navigate().GoToUrl(ConfigurationManager.AppSettings["seleniumBaseUrl"]);

// PageActivator.activePage = new ContactPage();

driver.Navigate().GoToUrl("https://www.sitekit.net/");

// PageActivator.activePage = new ContactPage();

}

[Test]

[When(@"I click on contact")]

public void WhenIClickOnContact()

{

// PageActivator.activePage = PageActivator.activePage.As<ContactPage>().contactDetails();

// PageActivator.currentPage.As<ContactDetailsPage>().contacting();

// Actions action = new Actions(driver);

// IWebElement acceptCookie = driver.FindElement(By.XPath("//\*[contains(text(),'Continue')]"));

// action.MoveToElement(acceptCookie).Click().Perform();

// IWebElement contact = driver.FindElement(By.LinkText("contact"));

var contact = driver.FindElement(By.XPath("//\*[contains(@href,'/#contact')]"));

// contact.Click();

Actions action = new Actions(driver);

action.MoveToElement(contact).Click().Build().Perform();

Assert.That(driver.PageSource.Contains("0845 299 0900"), Is.EqualTo(true), "You are not on contact page");

}

[Then(@"I should see the contact details")]

public void ThenIShouldSeeTheContactDetails()

{

var oxford = driver.FindElement(By.Id("Oxford"));

var london = driver.FindElement(By.Id("London"));

// assert multiple to

Assert.Multiple(() =>

{

Assert.That(oxford.Displayed, Is.Not.Null, "Oxford");

Assert.That(london.Displayed, Is.Not.Null, "London");

});

}

[TearDown]

public void tearDown()

{ making somechanges

driver.Quit();

// Browser.Current.Dispose();

}

}

}