1.3 Advanced Automation



This section will guide you to:

* Perform advanced automation with Selenium

**Development Environment:**

* Windows 10
* Visual Studio 2019 Community Version

This guide has six subsections, namely:

1. Creating a Unit Test project to do advanced automation with Selenium
2. Installing Selenium framework and drivers
3. Running a test case to test iframes and screenshot capture with Selenium
4. Building the project
5. Running all the tests in Test Explorer
6. Pushing the code to your GitHub repositories

**Step 1.3.1:** Creating a Unit Test project to test locating web elements in a page

* Open Visual Studio.
* From the top menu, click **File->New->Project.**
* Select **Unit Test Project (.NET Framework)** for Visual C# from the displayed project types and click **Next**.
* Put **Project Name** as Phase4Section1.11 and click **Create.**
* This will create a Unit Testing project.

**Step 1.3.2:**  Installing Selenium framework and drivers

* Selenium is locally installed in your practice lab. Refer **DotNet Lab guide: Phase 4** for more information.
* Use the following steps to install Selenium webdriver package within Visual Studio Code:
  + Open Visual Studio.
  + From the **Solution Explorer,** right click **Phase4Section1.4** and click **Manage Nuget Packages.**
  + Click on the **Browse** tab and search for Selenium.
  + Click on the Selenium item and click **Install.**
  + Repeat the above two steps for Selenium.Support and Selenium.Firefox.WebDriver.
  + Repeat the same steps for NUnit.
  + This will install the required Selenium libraries for the project.

**Step 1.3.3:** Running a test case to test iframes and screenshot capture with Selenium

* In **Solution Explorer** delete UnitTest1.cs. Right click **Phase4Section1.11** and click **Add->Class.** Enter **Name as** AdvAuto.cs and click **Add.**
* Add the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Drawing.Imaging;

**using** Microsoft.VisualStudio.TestTools.UnitTesting;

**using** OpenQA.Selenium;

**using** OpenQA.Selenium.Firefox;

**using** OpenQA.Selenium.Support.UI;

**namespace** Phase4Section1.\_11

{

[TestClass]

**public** **class** AdvAuto

{

[TestMethod]

**public** **void** IFrame()

{

**using** (**var** driver = **new** FirefoxDriver())

{

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

driver.Navigate().GoToUrl("file://D:/Specflow/Phase4Section1.4/iframetest.html");

**new** WebDriverWait(driver, TimeSpan.FromSeconds(10)).Until(

d => ((IJavaScriptExecutor)d).ExecuteScript("return document.readyState").Equals("complete"));

**var** iframes = driver.FindElementsByTagName("iframe");

Assert.IsTrue(iframes.Count == 2);

driver.SwitchTo().Frame(0);

**var** frameTitle = driver.PageSource.Contains("Corporate Training");

Assert.IsNotNull(frameTitle);

}

}

[TestMethod]

**public** **void** ScreenShot()

{

**using** (**var** driver = **new** FirefoxDriver())

{

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

driver.Navigate().GoToUrl("https://www.simplilearn.com");

**new** WebDriverWait(driver, TimeSpan.FromSeconds(10)).Until(

d => ((IJavaScriptExecutor)d).ExecuteScript("return document.readyState").Equals("complete"));

ITakesScreenshot screenshotDriver = driver **as** ITakesScreenshot;

Screenshot screenshot = screenshotDriver.GetScreenshot();

screenshot.SaveAsFile("d:/test.png", ScreenshotImageFormat.Png);

}

}

}

}

* In **Solution Explorer,** right click **Phase4Section1.11** and click **Add->New Item.** Choose Web->HTML Page.Enter **Name as** iframetest.html and click **Add.**
* Add the following script:

<!DOCTYPE html>

<**html** lang="en" xmlns="http://www.w3.org/1999/xhtml">

<**head**>

<**meta** charset="utf-8" />

<**title**></**title**>

</**head**>

<**body**>

IFrame Test<**br** />

<**iframe** src="https://www.simplilearn.com" frameborder="1" height="300" width="900"></**iframe**><**br** /><**br** />

<**iframe** src="https://www.yahoo.com" frameborder="1" height="300" width="900"></**iframe**><**br** /><**br** />

</**body**>

</**html**>

* In AdvAuto.cs change the location of the html file in driver.Navigate().GoToUrl() to point to the location of this file.
* In AdvAuto.cs change the location in method ScreenShot() for the line screenshot.SaveAsFile() to point to a path where you want the image to be saved.

**Step 1.3.4:** Building the project

* From the top menu, choose **Build->Build Solution.**
* If any compile errors are shown, fix them as required.

**Step 1.3.5:** Running all the tests in Test Explorer

* From the top menu, choose **Test->Test Explorer.**
* In Test Explorer, click on **Run All.**
* This will execute the tests and show the results in Test Explorer

**Step 1.3.6:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master