Vanilla C# Automation Framework

**Tools Required:**

1. Microsoft Visual Studio 2015

**Prerequisites:**

1. Copy and extract the Project zip folder to C:\
2. Download Nunit Console Application

Download link: <http://nunit.org/?p=download>

1. Download Allure-CommandLine tool

Download Link: <https://github.com/allure-framework/allure-core/releases/tag/allure-core-1.4.23>

Get **[allure-commandline.zip](https://github.com/allure-framework/allure-core/releases/download/allure-core-1.4.23/allure-commandline.zip)** from above link

Unzip to C:\ directory.

1. Download allure-nunit

Download Link: <https://github.com/allure-framework/allure-nunit2/releases>

Unzip those to **%Nunit Console Installation Directory%\bin**

**Environment Variables Set up:**

ALLURE\_CLI\_HOME=Installation directory path\bin (”C:\allure-commandline\bin”)

OUPUT\_DIR= “C:\Vanilla\Vanilla\TestLogResults”

NUNIT\_HOME=Installation directory path\bin (“C:\Program Files (x86)\NUnit 2.6.4\bin”)

PROJ\_DIR=<ProjectLocationTillBinFolder>

**Steps to add Test Script in Framework**

1. Right Click on TestCases folder
2. Add Item
3. Select Test from left side visual C# items
4. Select Basic Unit Test
5. Rename UnitTest(#no).cs to TestCaseName.cs
6. Include below namespaces to TestCaseName.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Vanilla.Development;

using Vanilla.Common;

using System.Collections.Generic;

using NUnit.Framework;

1. Add TestData for the testcase

Path to testdata folder: <Project Location> \TestData\TestCaseData\

Create .json file

Right click on TestData folder

Add existing item

Browse to created <TestCaseName>.json file

[ \*TestCaseName added in step 6= TestCaseName of .json]

Sample TestCaseName.json file

{

'emailID': 'abc@afourtech.com',

'password': 'abcdefg'

}

1. Add details to config.json if required.

Path to Config,json: ProjectDirectory \\TestData\\Config\Config.json

1. In TestScript

**[Setup]**

* Add [Setup] attribute
* Write method for setup
* Add Try, Catch, finally blocks
* Initialize all objects required for the test case
* In finally block write logs to the log file

**[TearDown]**

Add [TearDown] attribute

Write method for teardown

Add try catch blocks

Close the browser

Refer annotations noted below

[SetUp]

public void setUp()

[Test]

public void V\_TC1\_Test()

[TearDown]

public void cleanUp()

1. Add pages to the Framework

* Add class to Development folder.
* Rename it to <PageName>Page.cs
* Add required namespaces to the created page
* Inherit page from BasePage
* Add constructor
* Add ObjectRepository file to <Project Location>ObjectRepository folder
* Load ObjRepository data for the page in Constructor

public MyDeskHomePage(bool isVerify=true):base(false)

{

objData = configuration.GetValuesFromXML(configuration.CurrentDirectory + "\\ObjectRepository\\" + this.GetType().Name + ".xml");

if (isVerify)

{

LocateControl();

}

}

* Add LocateControl method
  + Find core controls (webElements) on the loaded page.

[ \*PageName added in step = PageName.xml]

1. Add test cases to TestSuite

Open TestSuite.cs

Add TestCaseName() to list

[Suite]

public static IEnumerable Suite

{

get

{

ArrayList suite = new ArrayList();

suite.Add(new V\_TC1());

suite.Add(new UnitTest1());

return suite;

}

}

1. Execution and Reporting

**Execution**

* Open nunit console

Cd %NUNIT\_HOME%

* Execute below command:

Nunit-console.exe /fixture=Vanilla.TestSuite.TestSuite %PROJ\_DIR\debug\Vanilla.dll

**Allure report generation**

Cd %ALLURE\_CLI\_HOME

Execute below command:

Allure report generate %OUTPUT\_DIR%

It will generate the xml file for the allure reporting

And opens the final allure report.