**Selenium C# How to Generate Extent Reports**

Reports play a fundamental role when it comes to TESTING. Tester can now know the real-time reports of test suit execution. Reports made ease to know the ratio of Pass? Or Fail? Post-test suit execution and it is the only documentation to know about test execution results.

Everyone wish to see the detailed description of the test results. Don’t you? here is the solution for it. And, let us see how these reports can be achieved? in Selenium C# – NUnit framework automation testing.

To achieve detailed test execution results as HTML reports we need to rely on third party tool called => Extent Reports. These reports provide decent narration of test execution results and are depicted in PIE chart.

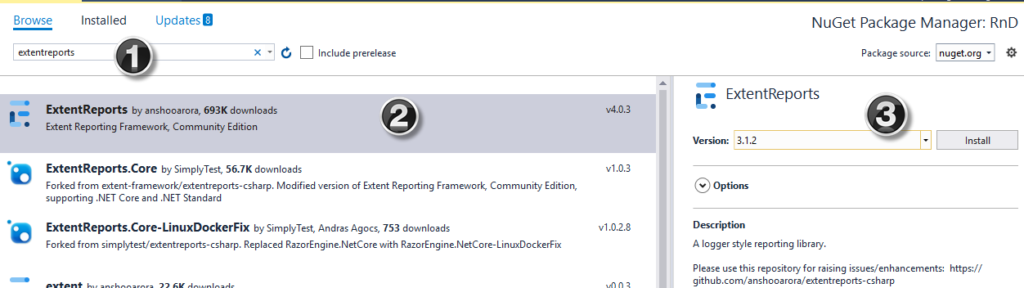
**How to Reference Extent Reports in MS Visual Studio**

Extent Reports can be directly referenced via **NuGet** Gallery:

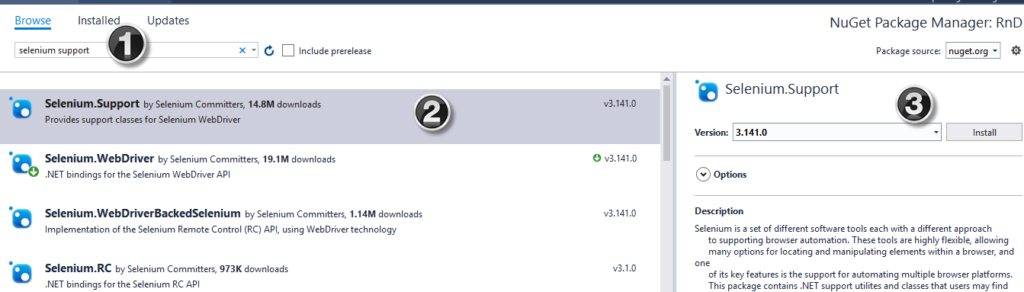
**Step 1)** Project**>** Manage NuGet Packages

**Step 2)** In the next window

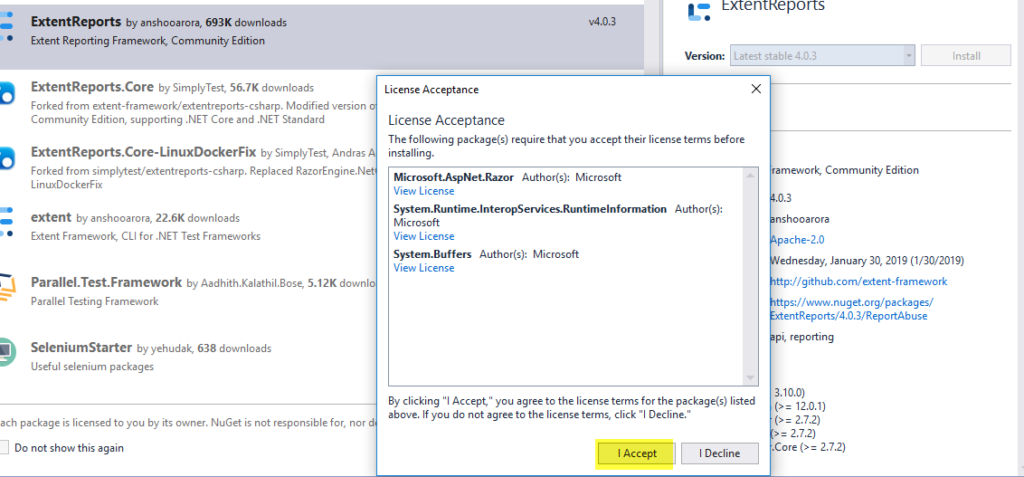
1. Search for **ExtentReports**
2. Select the search result
3. Click Install



**Step 3)**Install selenium support from NuGet package



**Step 3)**Click ‘I Accept’



**Step 4)** **Create a new C# class with the below code for Extent Reports.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using OpenQA.Selenium;

using OpenQA.Selenium.Support.UI;

using OpenQA.Selenium.Chrome;

using NUnit.Framework;

using NUnit.Framework.Interfaces;

using NUnit;

using AventStack.ExtentReports.Reporter;

using AventStack.ExtentReports;

using System.IO;

namespace RnD

{

[TestFixture]

public class TestDemo1

{

public IWebDriver driver;

public static ExtentTest test;

public static ExtentReports extent;

[SetUp]

public void Initialize()

{

driver = new ChromeDriver();

}

[OneTimeSetUp]

public void ExtentStart()

{

extent = new ExtentReports();

var htmlreporter = new ExtentHtmlReporter(@"D:\ReportResults\Report" + DateTime.Now.ToString("\_MMddyyyy\_hhmmtt") + ".html");

extent.AttachReporter(htmlreporter);

}

[Test]

public void BrowserTest()

{

test = null;

test = extent.CreateTest("T001").Info("Login Test");

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

driver.Navigate().GoToUrl("http://testing-ground.scraping.pro/login");

test.Log(Status.Info, "Go to URL");

//provide username

driver.FindElement(By.Id("usr")).SendKeys("admin");

//provide password

driver.FindElement(By.Id("pwd")).SendKeys("12345");

try

{

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(1));

wait.Until(ExpectedConditions.ElementIsVisible(By.XPath("//h3[contains(.,'WELCOME :)')]")));

//Test Result

test.Log(Status.Pass, "Test Pass");

}

catch (Exception e)

{

test.Log(Status.Fail, "Test Fail");

throw;

}

}

[TearDown]

public void closeBrowser()

{

driver.Close();

}

[OneTimeTearDown]

public void ExtentClose()

{

extent.Flush();

}

}

}

Post running test method, the test execution report looks as shown below: