1.5 Exporting Reports in Excel



This section will guide you to:

* Export a report in excel using Selenium WebDriver

**Development Environment:**

* Eclipse IDE for Enterprise Java Developers Version Oxygen.3a Release (4.7.3a)
* JavaDevelopment Kit Version 8
* Apache POI jars
* AT Excel Report jars
* Selenium standalone server jar

This guide has mainly four sub-sections, namely:

1.5.1 Creating a project with test case with multiple annotations

1.5.2 Adding AT Excel report jars

1.5.3 Executing the test suits to see the generated report in excel sheet

1.5.4 Pushing the code to your GitHub repositories

**Step 1.5.1** Creating a project with test cases with multiple annotations

* Open the Eclipse and create a Java project.
* Create multiple test case classes(Say Test\_01, Test02).
* Create a Base class to extend the test cases.

**Step 1.5.2** Adding AT Excel report jars

* Extent Reports jar file is already present in your practice lab in /home/ubuntu/libs directory.
* Add the Extent Reports jar file to your project: Right-click on project->Build path->Configure build path->Add external Jars.
* Click on Apply and then click OK.

**Step 1.5.3** Executing the test suites to see the generated report in Excel sheet

* Write the test script in the Test\_01 class.

**package** Testcases;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Test;

**import** base.Baseclass;

**public** **class** TEST\_01 **extends** Baseclass {

**@Test**

**public** void t\_001()

{

driver.findElement(By.xpath("(//\*[contains(text(),'Categories')])[1]")).click();

driver.findElement(By.xpath("//\*[contains(text(),'Central')]")).click();

**System**.out.println("Test\_01 executed successfully");

}

}

* Write the test script in the Test\_02 class.

**package** Testcases;

**import** org.openqa.selenium.By;

**import** org.testng.annotations.Test;

**import** base.Baseclass;

**public** **class** TEST\_02 **extends** Baseclass {

**@Test**

**public** void t\_002() {

driver.findElement(By.xpath("(//\*[contains(text(),'Popular')])[1]")).click();

**System**.out.println("Test\_02 executed successfully");

}

}

* Write the test script for the extended Base class, where all annotations are declared here.

**package** base;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.BeforeMethod;

**import** org.automationtesting.excelreport.Xl;

**public** **class** Baseclass {

**public** WebDriver driver;

**@BeforeMethod**

**public** void baseclass1()

{

**System**.setProperty("webdriver.gecko.driver", "/home/ubuntu/Downloads/gechodriver");

driver = **new** geckodriver();

driver.get("https://mvnrepository.com/");

}

**@AfterMethod**

**public** void quitDriver() {

driver.close();

}

**@AfterSuite**

**public** void generateReport() **throws** **Exception** {

Xl.generateReport("Report\_Excel.xlsx");

}

}

* Execute the test suite with multiple test cases and the testng.xml file will look like :

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name="ExportReport">

<test name="TEST1">

<classes>

<**class** name="Testcases.TEST\_01"></class>

</classes>

</test>

<test name="TEST2">

<classes>

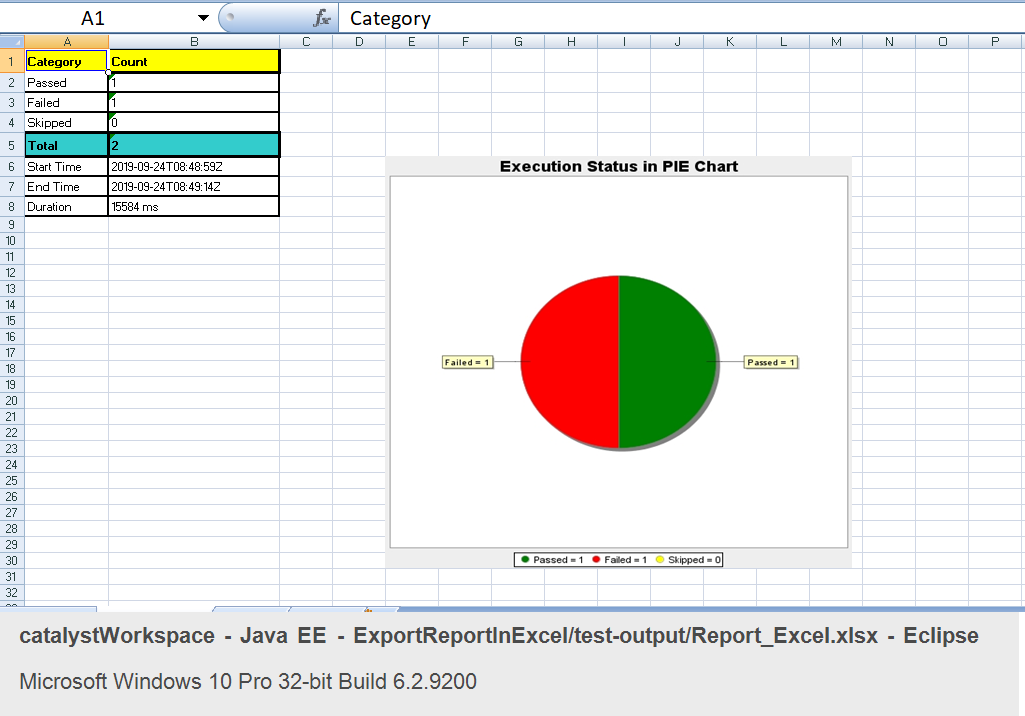
<**class** name="Testcases.TEST\_02"></class>

</classes>

</test>

</suite>

Finally, the executed script can generate the report in Excel and the graph will look like :



**Step 1.5.4:** 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 initially created using the following command:

git push -u origin master