Spice up your Selenium Test Automation Framework with Extent Reporting

Test automation reporting helps us to understand the heath of each test automation execution run. How many test automation scenarios passed and how many failed, why test failed, screen shot of failure.

In this article I will show you how we can integrate the latest extent reporting framework for our test automation framework. First of all we need to add the following dependency

To start lets add the following dependency for the maven project.

```
<dependency>
    <groupId>com.aventstack</groupId>
    <artifactId>extentreports</artifactId>
    <version>4.0.9</version>
</dependency>
```

Next I have created a reusable extent reporting common function..

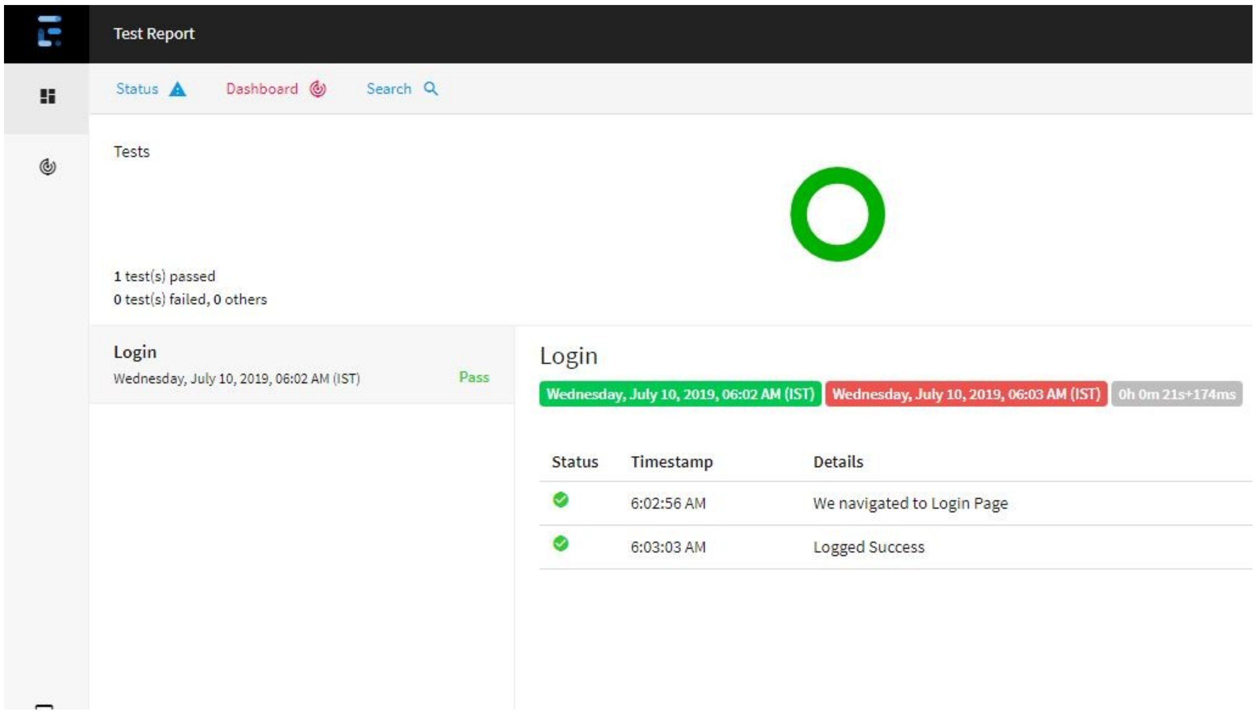
```
package ExtentReports;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.Theme;

public class Extent {

   ExtentReports extent;

   ExtentTest logger;

   ExtentHtmlReporter htmlReporter;

  public void setUpReport ()

  {

            htmlReporter = new ExtentHtmlReporter(System.getProperty("user.dir") +"/test-output/testReport.html");
```

```
    extent = new ExtentReports();
     extent.attachReporter(htmlReporter);
       //To add system or environment info by using the setSystemInfo method.
     extent.setSystemInfo("OS", System.getProperty("os.name"));
     extent.setSystemInfo("Browser", "Chrome");
      //configuration items to change the look and feel
     //add content, manage tests etc
     htmlReporter.config().setDocumentTitle("Extent Report Demo");
     htmlReporter.config().setReportName("Test Report");
     htmlReporter.config().setTheme(Theme.STANDARD);
     htmlReporter.config().setTimeStampFormat("EEEE, MMMM dd, yyyy, hh:mm a '('zzz')'");
    }
  public void logEventsPass (String value)
  {
          logger.log(Status.PASS, value);
  }

  public void logEventsFail (String value)
  {
          logger.log(Status.FAIL, value);
  }
    public void startTestCase (String testcaseName)
  {
          logger = extent.createTest(testcaseName);
  }
  public void createFinalReport()
  {
          extent.flush();
}
```

Finally we call the methods of this class within our test script. First call the setup method, thereafter call the start test case method. At the validation points call the logEventsPass and logEventsFail method.

```
extRpt.setUpReport();

  extRpt.startTestCase("Login"); //Should put after starting each test case

 //Loggin at each validation point

 if (loginPageTitle.toLowerCase().contains("guru99 bank"))

 {

              extRpt.logEventsPass("We navigated to Login Page");

 }

 else

 {

    extRpt.logEventsFail("We didn't navigated to Login Page");

 }

//Genarate report finally

extRpt.createFinalReport();
```

You will get a report similar to bellow.



1 test(s) passed
0 test(s) failed, 0 others

Login
Wednesday, July 10, 2019, 06:02 AM (IST) Pass

Login
Wednesday, July 10, 2019, 06:02 AM (IST) Wednesday, July 10, 2019, 06:03 AM (IST) 0h 0m 21s+174ms

Status	Timestamp	Details
✓	6:02:56 AM	We navigated to Login Page
✓	6:03:03 AM	Logged Success