**How to Generate Extent Reports in Selenium Webdriver [2023 Update]**

[](https://www.softwaretestingmaterial.com/)By[Rajkumar](https://www.softwaretestingmaterial.com/)Updated onJuly 11, 2023

**Guide To Generate Extent Reports In Selenium WebDriver** **i.e., Extent Spark Reporter (Extent Reports Version 5)**

In this article, we see one of the most popular and widely used Selenium Reporting Tools (Extent Reports). Selenium Results can be seen using different Selenium Reporting tools. Some of the Selenium WebDriver Reporting tools are as follows.

* [Selenium TestNG Report Generation](https://www.softwaretestingmaterial.com/testng-reports/)
* Selenium JUnit Report Generation
* Selenium Extent Reports (Most Popular Report)

**Table of Contents**

* [What are the Extent Reports?](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-what-are-the-extent-reports)
* [Advantages of Extent Reports](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-advantages-of-extent-reports)
* [Using Extent Reports in Selenium Webdriver](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-using-extent-reports-in-selenium-webdriver)
* [Generating Extent Reports In Selenium WebDriver](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-generating-extent-reports-in-selenium-webdriver)
  + [Pre-requisites to Generate Extent Reports](https://www.softwaretestingmaterial.com/generate-extent-reports/#prerequisites-to-generate-extent-reports)
  + [Steps To Generate Extent Reports](https://www.softwaretestingmaterial.com/generate-extent-reports/#steps-to-generate-extent-reports)
  + [Code to validate title and logo on Google home page.](https://www.softwaretestingmaterial.com/generate-extent-reports/#code-to-validate-title-and-logo-on-google-home-page)
  + [Code Explanation](https://www.softwaretestingmaterial.com/generate-extent-reports/#code-explanation)
  + [Extent Spark Reporter | Graphical Report with PIE Charts (Sample reports)](https://www.softwaretestingmaterial.com/generate-extent-reports/#extent-spark-reporter--graphical-report-with-pie-charts-sample-reports)
* [Wrapping up](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-wrapping-up)
* [FAQ's](https://www.softwaretestingmaterial.com/generate-extent-reports/#h-faq-s)
  + [Can we customize extent reports?](https://www.softwaretestingmaterial.com/generate-extent-reports/#can-we-customize-extent-reports)
  + [How to change title in the extent reports?](https://www.softwaretestingmaterial.com/generate-extent-reports/#how-to-change-title-in-the-extent-reports)
  + [How to change the theme to dark theme?](https://www.softwaretestingmaterial.com/generate-extent-reports/#how-to-change-the-theme-to-dark-theme)
  + [How to resolve HTTPS protocol error in extent reports?](https://www.softwaretestingmaterial.com/generate-extent-reports/#how-to-resolve-https-protocol-error-in-extent-reports)
  + [What is the use of extent reports?](https://www.softwaretestingmaterial.com/generate-extent-reports/#what-is-the-use-of-extent-reports)
  + [Is extent report part of TestNG?](https://www.softwaretestingmaterial.com/generate-extent-reports/#is-extent-report-part-of-testng)

1. 

Play Video

1. 

**What are the Extent Reports?**

ExtentReports is an open-source reporting library used in selenium test automation. Extent reports become the first choice of Selenium Automation Testers, even though Selenium comes with inbuilt reports using frameworks like JUnit and TestNG. With extent reports, you can offer a more extensive and insightful perspective on the execution of your automation scripts. So let’s see why Automation testers prefer Extent reports to others.

Ezoic

**Advantages of Extent Reports**

Some of the advantages of Extent Reports are as follows

* Extent reports are more customizable than others.
* Captures screenshots at every test step
* Displays the time taken for test case execution within the report.
* Extent API can produce more interactive reports, a dashboard view, graphical view, and emailable reports
* You can easily track multiple test case runs that occur within a single suite.
* It can be easily integrated with frameworks like JUnit, NUnit, & [TestNG](https://www.softwaretestingmaterial.com/testng-tutorial/)

**Using Extent Reports in Selenium Webdriver**

Extent Reports in Selenium contain two major classes that are used frequently.

* ExtentReports class
* ExtentTest class

**Syntax**

Ezoic

ExtentReports reports = new ExtentReports("Path of directory to store the resultant HTML file", true/false);

ExtentTest test = reports.startTest("TestName");

The ExtentReports class is used to generate an HTML reports based on a path specified by the user. The Boolean flag decides whether the existing report has to be overwritten or a new report must be generated. Value ‘true’ is the default value, meaning that all the existing data will be overwritten.

The ExtentTest class is used to log test steps on the previously generated HTML report.

Both the above classes can be used with the frequently used built-in methods that are listed below.

* startTest: Used to execute preconditions of a test case
* endTest: Used to execute postconditions of a test case
* Log: Used to log the status of each test step onto the resultant HTML report.
* Flush: Used to erase any previous data on the relevant report and create a whole new report

Test Status can be indicated by the following values:

Ezoic

* PASS
* FAIL
* SKIP
* INFO

**Syntax**

reports.endTest();

test.log(LogStatus.PASS,"Test Passed");

test.log(LogStatus.FAIL,"Test Failed");

test.log(LogStatus.SKIP,"Test Skipped");

test.log(LogStatus.INFO,"Test Info");

The Log method takes in two parameters, the first being the test status and the second being the message to be printed onto the resultant report.

**Generating Extent Reports In Selenium WebDriver**

The main scope of this article is to show the Extent Reports. This is the most popular and widely used Selenium Reporting tool in the current market.

Let’s see how to generate extent reports in Selenium WebDriver.

Ezoic

[***Extent Reports Version 4 – Previous Version [2020]***](https://www.softwaretestingmaterial.com/extent-reports-selenium-version-4/)

**Pre-requisites to Generate Extent Reports**

1. **Java** should be installed – [Install and setup Java](https://youtu.be/FBtapPZIrjo)
2. **TestNG** should be installed – [Install TestNG](https://www.softwaretestingmaterial.com/install-testng-plugin/)
3. **Extent Report Jars** – (Version 5.0.9)
4. **extent-config.xml**– It allows to configure HTML Report

**Steps To Generate Extent Reports**

**Step #1:**Firstly, create a TestNG project in eclipse

**Step #2:** Download extent library files (Extent reports Version 5 JAR file – Extent Spark Reporter) and add the downloaded library files to your project

[Download Extent Reports Jars](https://mvnrepository.com/artifact/com.aventstack/extentreports)

If you are using pom.xml file then you can add Extent Reports Maven dependency 5.0.9 in your maven project.

Ezoic

*<dependency>  
<groupId>com.aventstack</groupId>  
<artifactId>extentreports</artifactId>  
<version>5.0.9</version>  
</dependency>*

**Step #3:**Create a java class say ‘ExtentReportsClass’ and add the following code to it

[***Selenium Continuous Integration With Jenkins [Maven – Git – Jenkins]***](https://www.softwaretestingmaterial.com/selenium-continuous-integration/)

**Code to validate title and logo on Google home page.**

Given clear explanation in the comments section with in the program itself. Please go through it to understand the flow.

Ezoic

package myExtentReport;

import java.io.File;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.\*;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.SkipException;

import org.testng.annotations.\*;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentSparkReporter;

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

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

public class ExtentReportsClass {

public WebDriver driver;

public ExtentSparkReporter spark;

public ExtentReports extent;

public ExtentTest logger;

@BeforeTest

public void startReport() {

// Create an object of Extent Reports

extent = new ExtentReports();

spark = new ExtentSparkReporter(System.getProperty("user.dir") + "/test-output/STMExtentReport.html");

extent.attachReporter(spark);

extent.setSystemInfo("Host Name", "SoftwareTestingMaterial");

extent.setSystemInfo("Environment", "Production");

extent.setSystemInfo("User Name", "Rajkumar SM");

spark.config().setDocumentTitle("Title of the Report Comes here ");

// Name of the report

spark.config().setReportName("Name of the Report Comes here ");

// Dark Theme

spark.config().setTheme(Theme.STANDARD);

}

//This method is to capture the screenshot and return the path of the screenshot.

public static String getScreenShot(WebDriver driver, String screenshotName) throws IOException {

String dateName = new SimpleDateFormat("yyyyMMddhhmmss").format(new Date());

TakesScreenshot ts = (TakesScreenshot) driver;

File source = ts.getScreenshotAs(OutputType.FILE);

// after execution, you could see a folder "FailedTestsScreenshots" under src folder

String destination = System.getProperty("user.dir") + "/Screenshots/" + screenshotName + dateName + ".png";

File finalDestination = new File(destination);

FileUtils.copyFile(source, finalDestination);

return destination;

}

@BeforeMethod

public void setup() {

System.setProperty("webdriver.chrome.driver","C://AutomationFramework//Drivers//chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get("https://www.google.com/");

}

@Test

public void verifyTitle() {

logger = extent.createTest("To verify Google Title");

Assert.assertEquals(driver.getTitle(),"Google");

}

@Test

public void verifyLogo() {

logger = extent.createTest("To verify Google Logo");

boolean img = driver.findElement(By.xpath("//img[@id='hplogo']")).isDisplayed();

logger.createNode("Image is Present");

Assert.assertTrue(img);

logger.createNode("Image is not Present");

Assert.assertFalse(img);

}

@AfterMethod

public void getResult(ITestResult result) throws Exception{

if(result.getStatus() == ITestResult.FAILURE){

//MarkupHelper is used to display the output in different colors

logger.log(Status.FAIL, MarkupHelper.createLabel(result.getName() + " - Test Case Failed", ExtentColor.RED));

logger.log(Status.FAIL, MarkupHelper.createLabel(result.getThrowable() + " - Test Case Failed", ExtentColor.RED));

//To capture screenshot path and store the path of the screenshot in the string "screenshotPath"

//We do pass the path captured by this method in to the extent reports using "logger.addScreenCapture" method.

//String Scrnshot=TakeScreenshot.captuerScreenshot(driver,"TestCaseFailed");

String screenshotPath = getScreenShot(driver, result.getName());

//To add it in the extent report

logger.fail("Test Case Failed Snapshot is below " + logger.addScreenCaptureFromPath(screenshotPath));

}

else if(result.getStatus() == ITestResult.SKIP){

logger.log(Status.SKIP, MarkupHelper.createLabel(result.getName() + " - Test Case Skipped", ExtentColor.ORANGE));

}

else if(result.getStatus() == ITestResult.SUCCESS)

{

logger.log(Status.PASS, MarkupHelper.createLabel(result.getName()+" Test Case PASSED", ExtentColor.GREEN));

}

driver.quit();

}

@AfterTest

public void endReport() {

extent.flush();

}

}

**extent-config.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<extentreports>

<configuration>

<!-- report theme -->

<!-- standard, dark -->

<theme>standard</theme>

<!-- document encoding -->

<!-- defaults to UTF-8 -->

<encoding>UTF-8</encoding>

<!-- protocol for script and stylesheets -->

<!-- defaults to https -->

<protocol>https</protocol>

<!-- title of the document -->

<documentTitle>ExtentReports 2.0</documentTitle>

<!-- report name - displayed at top-nav -->

<reportName></reportName>

<!-- report headline - displayed at top-nav, after reportHeadline -->

<reportHeadline>Automation Report</reportHeadline>

<!-- global date format override -->

<!-- defaults to yyyy-MM-dd -->

<dateFormat>yyyy-MM-dd</dateFormat>

<!-- global time format override -->

<!-- defaults to HH:mm:ss -->

<timeFormat>HH:mm:ss</timeFormat>

<!-- custom javascript -->

<scripts>

<![CDATA[

$(document).ready(function() {

});

]]>

</scripts>

<!-- custom styles -->

<styles>

<![CDATA[

]]>

</styles>

</configuration>

</extentreports>

By using this external XML file (extent-config.xml), we could change the details such as Report Theme (either standard or dark), Report Title, Document Title etc.,

**Console Output:**

==============================================  
Default suite  
Total tests run: 2, Failures: 1, Skips: 0 ==============================================

Ezoic

Refresh the project after execution of above ExtentReportsClass.java file. You could find an HTML file named “***STMExtentReport.html***” in your test-output folder. Copy the location of the ***STMExtentReport.html*** file and open it by using any browser. You could see beautiful high rich HTML reports as shown below.

**Code Explanation**

i. Imported two classes ***ExtentReports*** and ***ExtentTest***.

***ExtentReports*:**By using this class we set the path where our reports need to generate.

***ExtentTest*:**By using this class we could generate the logs in the report.

Ezoic

ii. Took two methods with **@Test** annotation such as ***verifyTitle***, ***VerifyLogo***. another method ***startReport***with **@BeforeTest** annotation, another method ***endReport*** with **@AfterTest**annotation, another method ***getResult***with **@AfterMethod** annotation, and another method ***setup*** with **@BeforeMethod** annotation.

Here my intention is to generate a report with all the three types of results such as Pass, and Fail.

[***Add Screenshots of a failed Test Cases in Extent Reports***](https://www.softwaretestingmaterial.com/screenshots-extent-reports/)

iii. Used object of ***ExtentReports***class (i.e., *extent*) in the *startReport* method which was assigned to @BeforeTest annotation to generate the HTML report in the required path

iv. Used object of ***ExtentTest*** class (i.e., *logger*) in the remaining methods to write logs in the report.

Ezoic

v. Used ***ITestResult*** class in the @AfterMethod to describes the result of a test.

**Extent Spark Reporter | Graphical Report with PIE Charts (Sample reports)**

**TestCases**

**Exceptions**

**Tags**

**Device**

**Author**

**Graphical Report**

***Images source:***[*ExtentReports.com*](https://www.extentreports.com/docs/v5/wiki/spark/spark.html)

Check the below video to see “**Extent Reports Complete Guide**“

If you liked this video, then please subscribe to our [YouTube Channel](https://www.youtube.com/c/SoftwareTestingMaterialWebsite?sub_confirmation=1) for more video tutorials.

Ezoic

**Wrapping up**

Generating reports is a key part of any [automation framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/). Reports give you visibility into the health of your automation tests and can help you identify areas where improvements are needed.

There are many different options for generating reports, but one of the most popular is using the Extent Reports.

Extent reports is an open-source library that generates customizable HTML reports. So it becomes the best choice of Selenium Automation Testers to generate test reports and share them with the stakeholders in the organization.

In addition to outlining all the executed tests, Extent Reports also provide pie charts that visually summarize the test scripts.

[***Test Automation Framework Interview Questions***](https://www.softwaretestingmaterial.com/test-automation-framework-interview-questions/)

***Ezoic***

If you are not a regular reader of SoftwareTestingMaterial.com then I highly recommend you to signup for the free email newsletter using the below form.

**FAQ’s**

**Can we customize extent reports?**

**How to change title in the extent reports?**

**How to change the theme to dark theme?**

**How to resolve HTTPS protocol error in extent reports?**

**What is the use of extent reports?**

**Is extent report part of TestNG?**

Top of Form

**Like This Post?**

Subscribe and get free access to subscriber-only guides, templates, and checklists.

You are subscribing to email updates. Unsubscribe anytime. Your data is safe.

Yes, I'm In!

Bottom of Form



[**Rajkumar**](https://www.softwaretestingmaterial.com/author/smrajkumar27gmail-com/)

Rajkumar SM is a founder of SoftwareTestingMaterial. He is a certified Software Test Engineer by profession and a blogger & a YouTuber by a choice. He has extensive experience in the field of Software Testing. Furthermore, he loves to be with his wife and a cute little kid 'Freedom'.

Post navigation

**[PREVIOUS](https://www.softwaretestingmaterial.com/extent-reports-introduction/)**

[Extent Reports Introduction In Selenium WebDriver – Complete Guide](https://www.softwaretestingmaterial.com/extent-reports-introduction/)

**[NEXT](https://www.softwaretestingmaterial.com/webdriver-event-listeners/)**

[WebDriver Event Listeners In Selenium | Selenium Tutorial](https://www.softwaretestingmaterial.com/webdriver-event-listeners/)