

Cucumber Integration with Extent Report

ExtendManager.java

```
package com.Listeners;

import java.io.File;

import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.reporter.ExtentHtmlReporter;
import com.aventstack.extentreports.reporter.configuration.ChartLocation;
import com.aventstack.extentreports.reporter.configuration.Theme;

public class ExtendManager {

    private static ExtentReports extent;
    private static String reportFileName = "Test-Automation-Report"+"".html";
    private static String fileSeperator = System.getProperty("file.separator");
    private static String reportFilepath = System.getProperty("user.dir") +fileSeperator+ "TestReport";
    private static String reportFileLocation = reportFilepath +fileSeperator+ reportFileName;

    public static ExtentReports getInstance() {
        if (extent == null)
            createInstance();
        return extent;
    }

    //Create an extent report instance
    public static ExtentReports createInstance() {
        String fileName = getReportPath(reportFilepath);

        ExtentHtmlReporter htmlReporter = new ExtentHtmlReporter(fileName);

        htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);
        htmlReporter.config().setChartVisibilityOnOpen(true);
        htmlReporter.config().setTheme(Theme.STANDARD);
        htmlReporter.config().setDocumentTitle(reportFileName);
        htmlReporter.config().setEncoding("utf-8");
        htmlReporter.config().setReportName(reportFileName);
        htmlReporter.config().setTimeStampFormat("EEEE, MMMM dd, yyyy, hh:mm a '('zzz'')");

        extent = new ExtentReports();
        extent.attachReporter(htmlReporter);
        //Set environment details
        extent.setSystemInfo("OS", "Windows");
        extent.setSystemInfo("AUT", "QA");
    }
}
```

```

        return extent;
    }

    //Create the report path
    private static String getReportPath (String path) {
        File testDirectory = new File(path);
        if (!testDirectory.exists()) {
            if (testDirectory.mkdir()) {
                System.out.println("Directory: " + path + " is
created!" );
                return reportFileLocation;
            } else {
                System.out.println("Failed to create directory: " +
path);
                return System.getProperty("user.dir");
            }
        } else {
            System.out.println("Directory already exists: " + path);
        }
        return reportFileLocation;
    }
}

```

ExtendTestManager.java

```

package com.Listeners;

import java.util.HashMap;
import java.util.Map;

import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.ExtentTest;

public class ExtentTestManager {

    static Map<Integer, ExtentTest> extentTestMap = new HashMap<Integer,
ExtentTest>();
    static ExtentReports extent = ExtentManager.getInstance();

    @SuppressWarnings("deprecation")
    public static synchronized ExtentTest getTest() {
        return (ExtentTest) extentTestMap.get((int) (long)
(Thread.currentThread().getId()));
    }

    public static synchronized void endTest() {
        extent.flush();
    }

    @SuppressWarnings("deprecation")
    public static synchronized ExtentTest startTest(String testName) {
        ExtentTest test = extent.createTest(testName);
        extentTestMap.put((int) (long)
(Thread.currentThread().getId()), test);
        return test;
    }
}

```

```
}  
}
```

TestListerner.java

```
package com.Listeners;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.text.SimpleDateFormat;  
  
import org.openqa.selenium.OutputType;  
import org.openqa.selenium.TakesScreenshot;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.io.FileHandler;  
import org.testng.ITestContext;  
import org.testng.ITestListener;  
import org.testng.ITestResult;  
  
import com.aventstack.extentreports.MediaEntityBuilder;  
import com.aventstack.extentreports.Status;  
import com.aventstack.extentreports.model.Log;  
  
import org.apache.log4j.Logger;  
  
public class TestListener implements ITestListener {  
    private static Logger log = Logger.getLogger(Log.class.getName());  
  
    public void onStart(ITestContext context) {  
        System.out.println("*** Test Suite " + context.getName() + "  
started ***");  
    }  
  
    public void onFinish(ITestContext context) {  
        System.out.println("*** Test Suite " + context.getName() + "  
ending ***");  
        ExtentTestManager.endTest();  
        ExtentManager.getInstance().flush();  
    }  
  
    public void onTestStart(ITestResult result) {  
        System.out.println("*** Running test method " +  
result.getMethod().getMethodName() + "...");  
  
        ExtentTestManager.startTest(result.getMethod().getMethodName());  
    }  
  
    public void onTestSuccess(ITestResult result) {  
        System.out.println("*** Executed " +  
result.getMethod().getMethodName() + " test successfully...");  
        ExtentTestManager.getTest().log(Status.PASS, "Test passed");  
    }  
  
    public void onTestFailure(ITestResult result) {
```

```

        Log.info("*** Test execution " +
result.getMethod().getMethodName() + " failed...");
        Log.info((result.getMethod().getMethodName() + " failed!"));

        //ITestContext context = result.getTestContext();
        WebDriver driver = (WebDriver)
result.getTestContext().getAttribute("driver");

        String targetLocation = null;

        String testClassName = result.getInstanceName().trim();
        String timeStamp = new
SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new java.util.Date()); //
get timestamp
        String testMethodName = result.getName().toString().trim();
        String screenShotName = testMethodName + timeStamp + ".png";
        String fileSeparator = System.getProperty("file.separator");
        String reportsPath = System.getProperty("user.dir") +
fileSeparator + "TestReport" + fileSeparator
            + "screenshots";
        Log.info("Screen shots reports path - " + reportsPath);
        try {
            File file = new File(reportsPath + fileSeparator +
testClassName); // Set

                                                    // screenshots
                                                    // folder

            if (!file.exists()) {
                if (file.mkdirs()) {
                    Log.info("Directory: " +
file.getAbsolutePath() + " is created!");
                } else {
                    Log.info("Failed to create directory: " +
file.getAbsolutePath());
                }
            }

            File screenshotFile = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);
            targetLocation = reportsPath + fileSeparator +
testClassName + fileSeparator + screenShotName;// define

                                                    // location
            File targetFile = new File(targetLocation);
            Log.info("Screen shot file location - " +
screenshotFile.getAbsolutePath());
            Log.info("Target File location - " +
targetFile.getAbsolutePath());
            FileHandler.copy(screenshotFile, targetFile);

        } catch (FileNotFoundException e) {
            Log.info("File not found exception occurred while
taking screenshot " + e.getMessage());
        } catch (Exception e) {
            Log.info("An exception occurred while taking screenshot
" + e.getCause());

```

```

    }

    // attach screenshots to report
    try {
        ExtentTestManager.getTest().fail("Screenshot",
            MediaEntityBuilder.createScreenCaptureFromPath(targetLocation).build
            ());
    } catch (IOException e) {
        Log.info("An exception occurred while taking screenshot
" + e.getCause());
    }
    ExtentTestManager.getTest().log(Status.FAIL, "Test Failed");
}

public void onTestSkipped(ITestResult result) {
    System.out.println("*** Test " +
result.getMethod().getMethodName() + " skipped...");
    ExtentTestManager.getTest().log(Status.SKIP, "Test Skipped");
}

public void onTestFailedButWithinSuccessPercentage(ITestResult
result) {
    System.out.println("*** Test failed but within percentage % "
+ result.getMethod().getMethodName());
}
}

```

pom.xml

```

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

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>in.Amazon</groupId>
    <artifactId>AmazonWithCucumber</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <name>AmazonWithCucumber</name>
    <!-- FIXME change it to the project's website -->
    <url>http://www.example.com</url>

    <properties>
        <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
        <maven.compiler.source>1.7</maven.compiler.source>
        <maven.compiler.target>1.7</maven.compiler.target>
    </properties>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
core -->

```

```

        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-core</artifactId>
            <version>7.1.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
java -->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-java</artifactId>
            <version>7.1.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
testng -->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-testng</artifactId>
            <version>7.1.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
jvm-deps -->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-jvm-deps</artifactId>
            <version>1.0.6</version>
            <scope>provided</scope>
        </dependency>
        <!--
https://mvnrepository.com/artifact/net.masterthought/cucumber-reporting -->
        <dependency>
            <groupId>net.masterthought</groupId>
            <artifactId>cucumber-reporting</artifactId>
            <version>5.6.1</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.cucumber/gherkin --
>
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>gherkin</artifactId>
            <version>22.0.0</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/com.beust/jcommander -
->
        <dependency>
            <groupId>com.beust</groupId>
            <artifactId>jcommander</artifactId>
            <version>1.81</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
html -->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-html</artifactId>
            <version>0.2.7</version>
        </dependency>
        <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java --
>
        <dependency>

```

```

        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.10.0</version>
    </dependency>
    <dependency>
        <groupId>com.google.guava</groupId>
        <artifactId>guava</artifactId>
        <version>31.0.1-jre</version>
    </dependency>
    <dependency>
        <groupId>com.aventstack</groupId>
        <artifactId>extentreports</artifactId>
        <version>3.1.5</version>
    </dependency>
    <dependency>
        <groupId>com.sun.mail</groupId>
        <artifactId>javax.mail</artifactId>
        <version>1.6.2</version>
    </dependency>
    <dependency>
        <groupId>javax.activation</groupId>
        <artifactId>activation</artifactId>
        <version>1.1</version>
    </dependency>
    <dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-api</artifactId>
        <version>2.11.1</version>
    </dependency>
    <dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-core</artifactId>
        <version>2.11.1</version>
    </dependency>
    <dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-1.2-api</artifactId>
        <version>2.11.1</version>
    </dependency>
</dependencies>

<build>
    <pluginManagement><!-- lock down plugins versions to avoid
using Maven
        defaults (may be moved to parent pom) -->
    <plugins>
        <!-- clean lifecycle, see
        https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean_Lifecycle -->
        <plugin>
            <artifactId>maven-clean-
plugin</artifactId>
            <version>3.1.0</version>
        </plugin>
        <!-- default lifecycle, jar packaging: see
        https://maven.apache.org/ref/current/maven-
core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
        <plugin>

```

```

        <artifactId>maven-resources-
plugin</artifactId>
        <version>3.0.2</version>
    </plugin>
    <plugin>
        <artifactId>maven-compiler-
plugin</artifactId>
        <version>3.8.0</version>
    </plugin>
    <plugin>
        <artifactId>maven-surefire-
plugin</artifactId>
        <version>2.22.1</version>
    </plugin>
    <plugin>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.0.2</version>
    </plugin>
    <plugin>
        <artifactId>maven-install-
plugin</artifactId>
        <version>2.5.2</version>
    </plugin>
    <plugin>
        <artifactId>maven-deploy-
plugin</artifactId>
        <version>2.8.2</version>
    </plugin>
    <!-- site lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#site_Lifecycle -->
    <plugin>
        <artifactId>maven-site-plugin</artifactId>
        <version>3.7.1</version>
    </plugin>
    <plugin>
        <artifactId>maven-project-info-reports-
plugin</artifactId>
        <version>3.0.0</version>
    </plugin>
</plugins>
</pluginManagement>
</build>
</project>

```

testng.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite">
    <listeners>
        <listener class-name="com.Listeners.TestListener"></listener>
    </listeners>

    <test thread-count="5" name="Test">

```



```
        <classes>
            <class name="in.Amazon.TestSteps.TestRunner"></class>
        </classes>
    </test> <!--Test -->
</suite> <!--Suite -->
```