Extent Report

21TH AUG, 2023

Contents

Extent Report 5 Features

Report Attachments

- To add attachments, like screen images, two settings need to be added to the extent.properties.
- Firstly property, named screenshot.dir, is the directory where the attachments are stored.
- Secondly is screenshot.rel.path, which is the relative path from the report file to the screenshot directory.

```
extent.reporter.spark.out=Reports/Spark.html


screenshot.dir=/Screenshots/
screenshot.rel.path=../Screenshots/
```

Extent PDF Reporter

- The PDF reporter summarizes the test run results in a dashboard and other sections with feature, scenario, and, step details.
- The PDF report needs to be enabled in the extent.properties file.

```
extent.reporter.pdf.start=true
extent.reporter.pdf.out=PdfReport/ExtentPdf.pdf
```

Ported HTML Reporter

- The HTML reporter summarizes the test run results in a dashboard and other sections with feature, scenario, and, step details.
- The HTML report needs to be enabled in the extent.properties file.

```
extent.reporter.html.start=true
extent.reporter.html.out=HtmlReport/ExtentHtml.html
```

Features

Customized Report Folder Name

- To enable the report folder name with date and\or time details, two settings need to be added to the extent.properties.
- These are basefolder.name and basefolder.datetimepattern. These will be merged to create the base folder name, inside which the reports will be generated.

```
basefolder.name=ExtentReports/SparkReport_
basefolder.datetimepattern=d_MMM_YY HH_mm_ss
```

Attach Image as Base64 String

- This feature can be used to attach images to the Spark report by setting the **src attribute of the img tag to a Base64** encoded string of the image.
- When this feature is used, no physical file is created. There is no need to modify any step definition code to use this.
- To enable this, use the below settings in extent.properties, which is false by default.

```
extent.reporter.spark.base64imagesrc=true
```

Environment or System Info Properties

- It is now possible to add environment or system info properties in the extent.properties or pass them in the maven command line.

```
systeminfo.os=windows
systeminfo.version=10
```

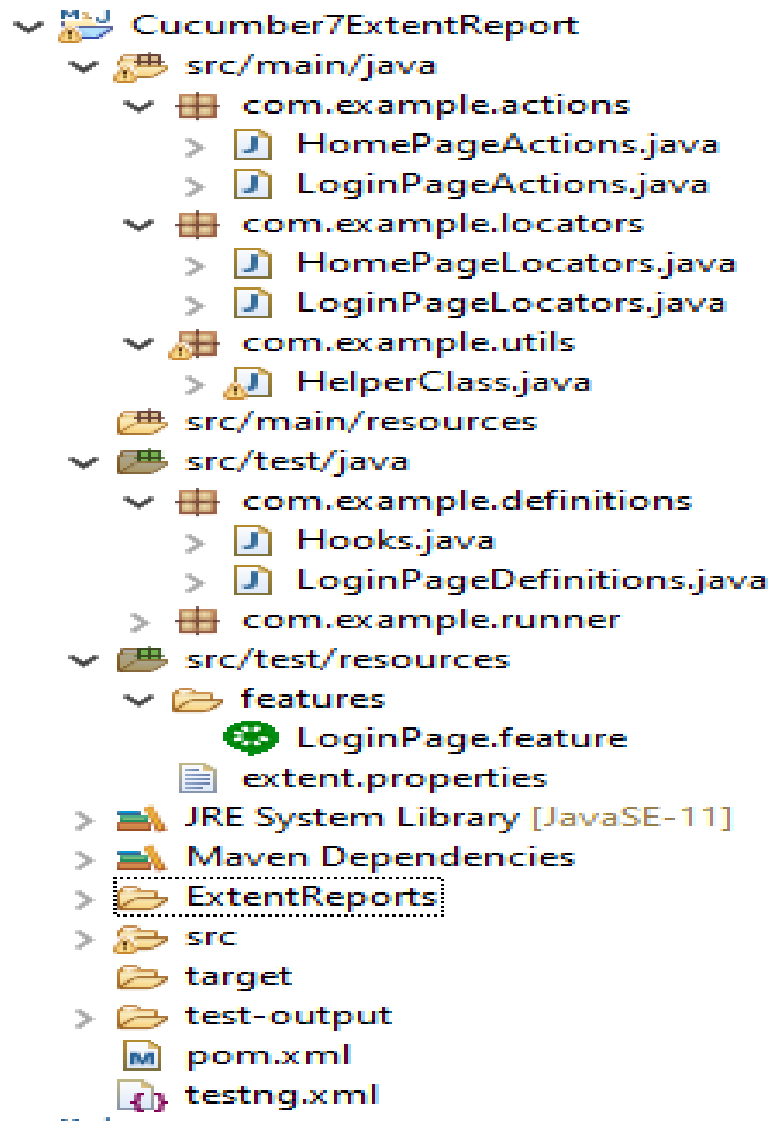
Prerequisite

Prerequisite

1. Java 8 or higher is needed for ExtentReport5
2. Maven or Gradle
3. JAVA IDE (like Eclipse, IntelliJ, or soon)
4. TestNG installed
5. Cucumber Eclipse plugin (in case using Eclipse)

Project Structure

Project Structure



```
Cucumber7ExtentReport
  src/main/java
    com.example.actions
      HomePageActions.java
      LoginPageActions.java
    com.example.locators
      HomePageLocators.java
      LoginPageLocators.java
    com.example.utils
      HelperClass.java
  src/main/resources
  src/test/java
    com.example.definitions
      Hooks.java
      LoginPageDefinitions.java
    com.example.runner
  src/test/resources
    features
      LoginPage.feature
    extent.properties
  JRE System Library [JavaSE-11]
  Maven Dependencies
  ExtentReports
  src
  target
  test-output
  pom.xml
  testng.xml
```

Implementation Steps

Implementation Steps

Step 1: Add Maven dependencies to the POM

Add ExtentReport dependency

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

Add tech grasshopper maven dependency for Cucumber

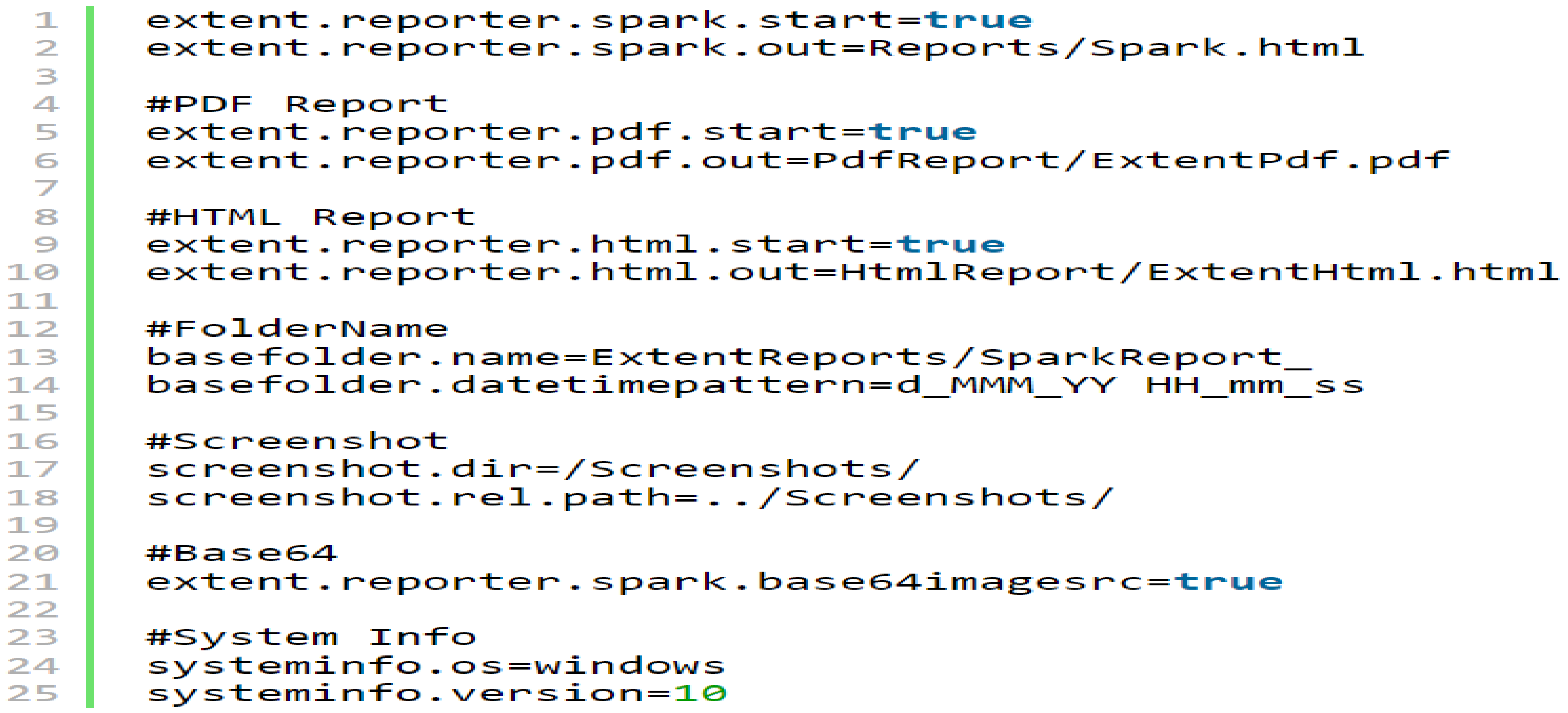
```xml
<dependency>
    <groupId>tech.grasshopper</groupId>
    <artifactId>extentreports-cucumber7-adapter</artifactI
    <version>1.7.0</version>
</dependency>
```

Step 2: Create a feature file in src/test/resources

- Create Simple feature file for checking a login in to any application.

```
Feature: Login to HRM Application
@ValidCredentials
Scenario: Login with valid credentials
Given User is on HRMLogin page "https://opensource-demo.orangehrmlive.com/"
When User enters username and password
Then User should be able to login sucessfully and new page open
```

Step 3: Create extent.properties file in src/test/resources



```
extent.reporter.spark.start=true
extent.reporter.spark.out=Reports/Spark.html

#PDF Report
extent.reporter.pdf.start=true
extent.reporter.pdf.out=PdfReport/ExtentPdf.pdf

#HTML Report
extent.reporter.html.start=true
extent.reporter.html.out=HtmlReport/ExtentHtml.html

#FolderName
basefolder.name=ExtentReports/SparkReport_
basefolder.datetimepattern=d_MMM_YY HH_mm_ss

#Screenshot
screenshot.dir=/Screenshots/
screenshot.rel.path=../Screenshots/

#Base64
extent.reporter.spark.base64imagesrc=true

#System Info
systeminfo.os=windows
systeminfo.version=10
```

Implementation Steps

Step 4: Create a Helper class in src/main/java

- Use Page Object Model with Cucumber and TestNG.
- Create a Helper class where we are
- Initializing the web driver,
- Initializing the web driver wait, defining the timeouts,
- and creating a private constructor of the class, it will declare the web driver,
- so whenever we create an object of this class, a new web browser is invoked.

Implementation Steps

Helper Class

```
public class HelperClass {
private static HelperClass helperClass;
private static WebDriver driver;
private static WebDriverWait wait;
public final static int TIMEOUT = 10;
HelperClass() {
// WebDriverManager.chromedriver().setup();
driver = new ChromeDriver();
wait = new WebDriverWait(driver, Duration.ofSeconds(TIMEOUT));
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(TIMEOUT));
driver.manage().window().maximize();
}
public static void openPage(String url) {
driver.get(url);
}
```

Implementation Steps

Helper Class

```
public static WebDriver getDriver() {
return driver;
}
public static void setUpDriver() {
if (helperClass==null) {
helperClass = new HelperClass();
}
}
public static void tearDown() {
if(driver!=null) {
driver.close();
driver.quit();
}
helperClass = null;
}
}
```

Step 5: Create Locator classes in src/main/java

- Create a locator class for each page that contains the detail of the locators of all the web elements.
- Create required locator classes
  - LoginPageLocators and HomePageLocators.

Implementation Steps

LoginPageLocators

```
package extentreport_demo;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class LoginPageLocators {
		@FindBy(name = "username")
		  public WebElement userName;

		  @FindBy(name = "password")
		  public WebElement password;

		  @FindBy(xpath =
"//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")
		  public WebElement login;

}
```

Implementation Steps

HomePageLocators

```
package extentreport_demo;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;

public class HomePageLocators {
            @FindBy(xpath = "//h6[text()='Dashboard']")
              public  WebElement homePageUserName;

}
```

Step 6: Create Action classes in src/main/java

- Create the action classes for each web page.
- These action classes contain all the methods needed by the step definitions.
- In this case, created 2 action classes – LoginPageActions and HomePageActions.

Implementation Steps

LoginPageActions

```
package extentreport_demo;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Enumeration;
import java.util.Properties;

import org.openqa.selenium.support.PageFactory;

public class LoginPageActions {
            LoginPageLocators loginPageLocators = null;
            String strUserName, strPassword;

   public LoginPageActions() {

      this.loginPageLocators = new LoginPageLocators();

      PageFactory.initElements(HelperClass.getDriver(),loginPageLocators);
   }
```

Implementation Steps

LoginPageActions

```
 // Set user name in textbox
public void setUserName(String strUserName) {
   loginPageLocators.userName.sendKeys(strUserName);
}

// Set password in password textbox
public void setPassword(String strPassword) {
   loginPageLocators.password.sendKeys(strPassword);
}
  // Click on login button
public void clickLogin() {
   loginPageLocators.login.click();
}
   public void login() {
         File file = new File("C:\\\\Users\\\\Arun Kumar\\\\eclipse-
workspace\\\\extentreport_demo\\src\\test\\resources\\data1.properties");

                    FileInputStream fileInput = null;
                    try {
                              fileInput = new FileInputStream(file);
                    } catch (FileNotFoundException e) {
                              e.printStackTrace();
                    }
```

Implementation Steps

LoginPageActions

```
                        Properties prop = new Properties();

                        //load properties file
                        try {
                                    prop.load(fileInput);
                        } catch (IOException e) {
                                    e.printStackTrace();
                        }

                        strUserName = prop.getProperty("username1");
                        strPassword = prop.getProperty("password1");

       // Fill user name
       this.setUserName(strUserName);

       // Fill password
       this.setPassword(strPassword);

       // Click Login button
       this.clickLogin();
    }
}
```

Implementation Steps

HomePageActions

```
import org.openqa.selenium.support.PageFactory;
public class HomePageActions {
HomePageLocators homePageLocators = null;
public HomePageActions() {
this.homePageLocators = new HomePageLocators();
PageFactory.initElements(HelperClass.getDriver(),homePageLocators);
}
// Get the User name from Home Page
public String getHomePageText() {
return homePageLocators.homePageUserName.getText();
}
}
```

Implementation Steps

Step 7: Create a Step Definition file in src/test/java

- Create the corresponding Step Definition file of the feature file.

Implementation Steps

LoginPageDefinitions

```
package extentreport_demo;

import org.testng.Assert;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class LoginPageDefinitions {
		LoginPageActions objLogin = new LoginPageActions();
    HomePageActions objHomePage = new HomePageActions();

    @Given("User is on HRMLogin page {string}")
    public void loginTest(String url) {

        HelperClass.openPage(url);

    }
```

Implementation Steps

LoginPageDefinitions

```
    @When("User enters username and password")
    public void goToHomePage() {

        // login to application
        objLogin.login();

        // go the next page

    }


    @Then("User should be able to login sucessfully and new page open")
    public void verifyLogin() {

        // Verify home page
        Assert.assertTrue(objHomePage.getHomePageText().contains("Dashboard"));

    }

}
```

Implementation Steps

Step 8: Create Hook class in src/test/java

- Create the hook class that contains the Before and After hook.
- @Before hook contains the method to call the setup driver which will initialize the chrome driver. This will be run before any test.
- @After to take screenshot and quit the browser.

Implementation Steps

HookClass

```
package extentreport_demo;

import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;

import extentreport_demo.HelperClass;
import io.cucumber.java.After;
import io.cucumber.java.Before;
import io.cucumber.java.Scenario;

public class HookClass {
                @Before
    public static void setUp() {
       HelperClass.setUpDriver();
    }

    @After
    public static void tearDown(Scenario scenario) {

        HelperClass.tearDown();
    }

}
```

Step 9: Create a Cucumber Test Runner class in src/test/java

- Add the extent report cucumber adapter to the runner class's CucumberOption annotation.

```
          import io.cucumber.testng.AbstractTestNGCucumberTests;

          import io.cucumber.testng.CucumberOptions;

  @CucumberOptions(tags = "", features =
"src/test/resources/features/LoginPage.feature", glue =
"com.example.definitions",

               plugin =
{"com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"})


public class CucumberRunnerTests extends AbstractTestNGCucumberTests {

  }
```

Step 10: Create the testng.xml for the project

- **Right-click on the project and select TestNG -> convert to TestNG.**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd"
<suite name="Suite">
  <test name="ExtentReport5 for Cucumber7">

  <classes>
  <class name = "com.example.runner.CucumberRunnerTests"/>
  </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

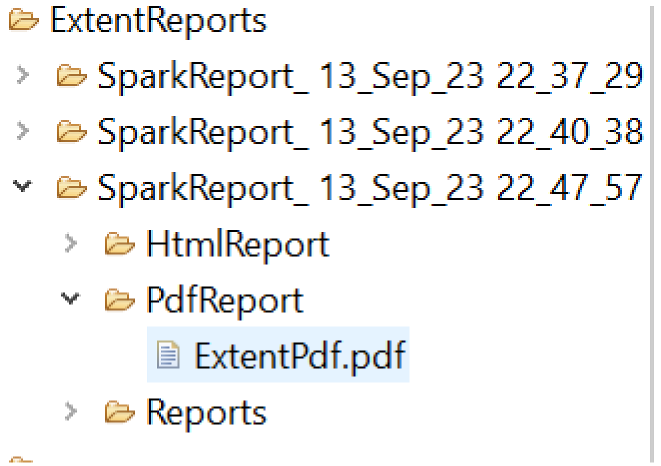
Implementation Steps

Step 11: Execute the code

- Right Click on the Runner class and select Run As -> TestNG Test.

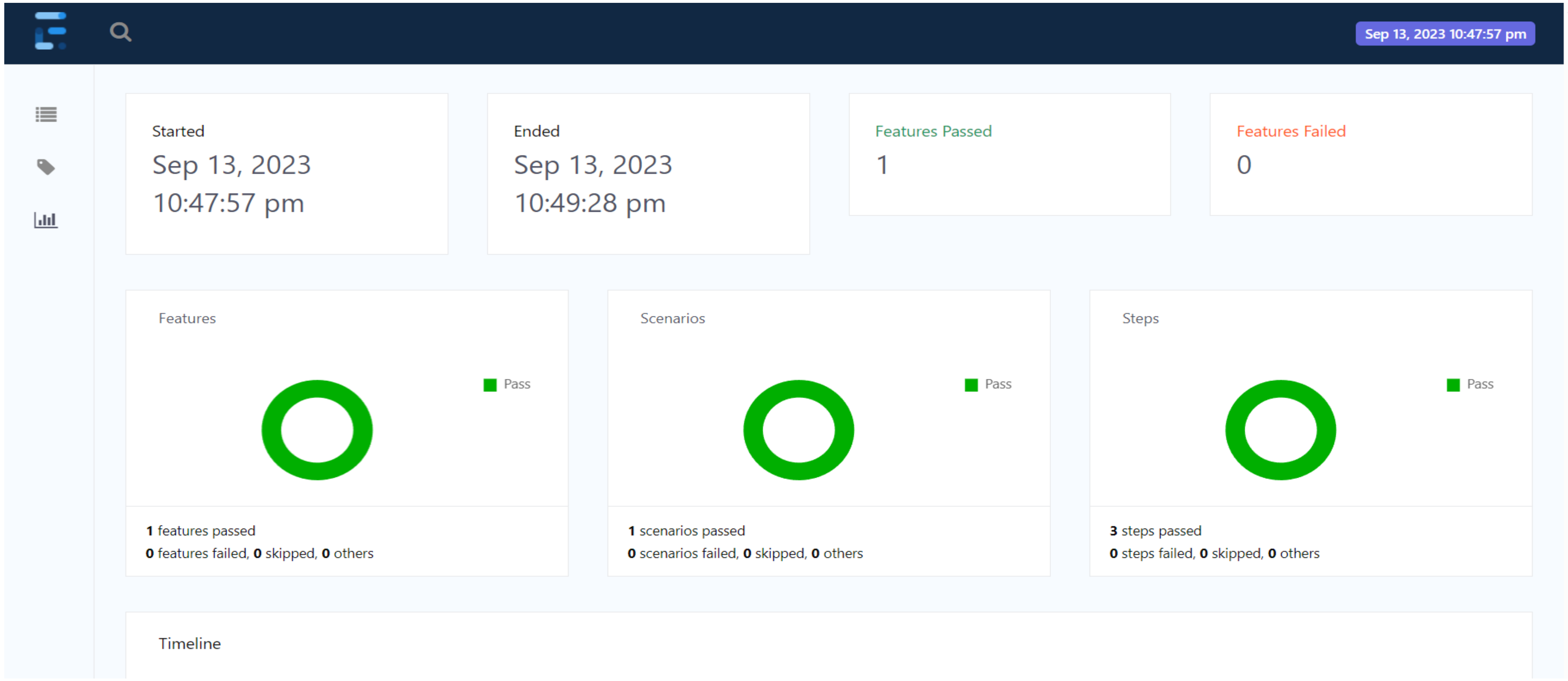
Step 12: View ExtentReport

- Refresh the project and will see a new folder – SparkReport_ which further contains 4 folders – HtmlReport, PdfReport, Reports, and Screenshots.



```
ExtentReports
  > SparkReport_ 13_Sep_23 22_37_29
  > SparkReport_ 13_Sep_23 22_40_38
  v SparkReport_ 13_Sep_23 22_47_57
      > HtmlReport
      v PdfReport
          ExtentPdf.pdf
      > Reports
```

Output



Sep 13, 2023 10:47:57 pm

Started
Sep 13, 2023
10:47:57 pm

Ended
Sep 13, 2023
10:49:28 pm

Features Passed
1

Features Failed
0

Features

🟩 Pass

1 features passed
0 features failed, **0** skipped, **0** others

Scenarios

🟩 Pass

1 scenarios passed
0 scenarios failed, **0** skipped, **0** others

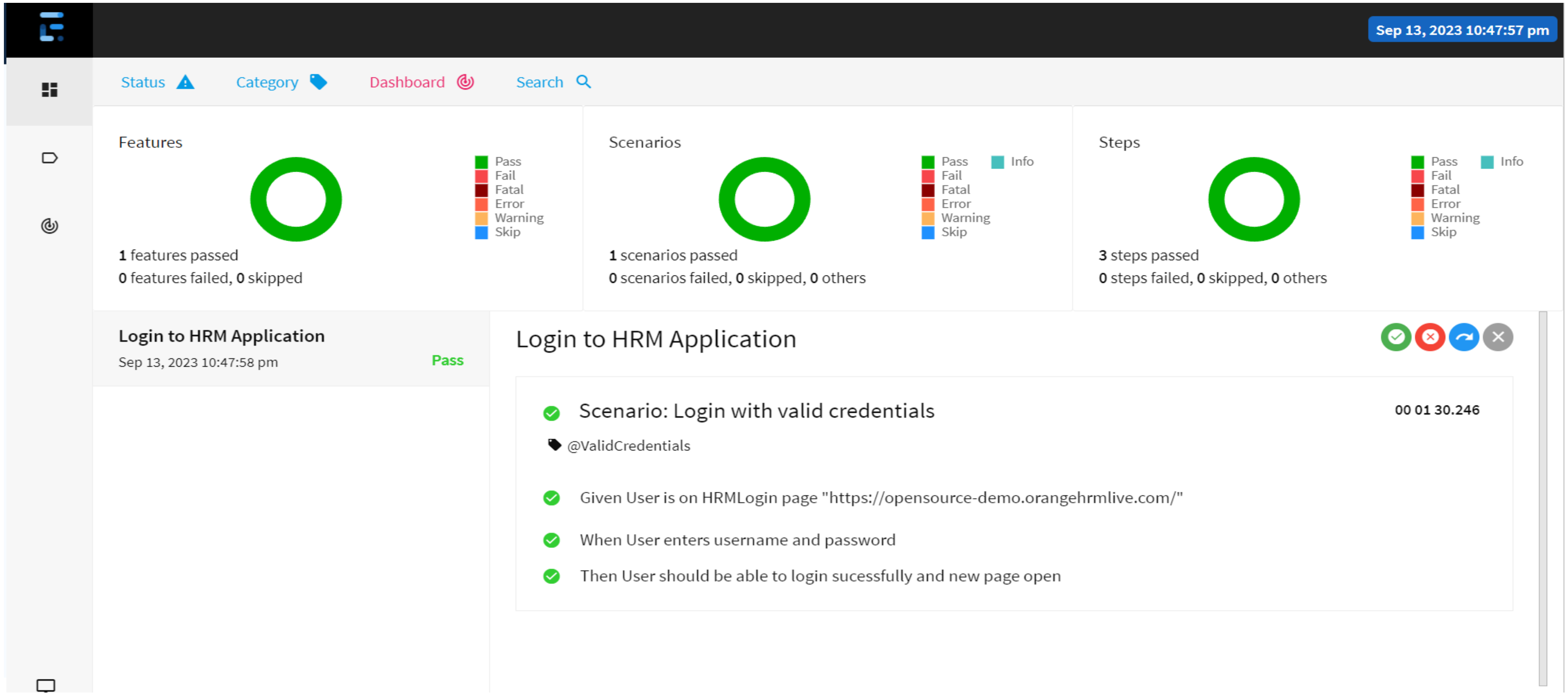
Steps

🟩 Pass

3 steps passed
0 steps failed, **0** skipped, **0** others

Timeline

Output


Sep 13, 2023 10:47:57 pm

Features

🟩 Pass
🟥 Fail
🟥 Fatal
🟧 Error
🟧 Warning
🟦 Skip

1 features passed
0 features failed, **0** skipped

Scenarios

🟩 Pass
🟥 Fail
🟥 Fatal
🟧 Error
🟧 Warning
🟦 Skip
🟦 Info

1 scenarios passed
0 scenarios failed, **0** skipped, **0** others

Steps

🟩 Pass
🟥 Fail
🟥 Fatal
🟧 Error
🟧 Warning
🟦 Skip
🟦 Info

3 steps passed
0 steps failed, **0** skipped, **0** others

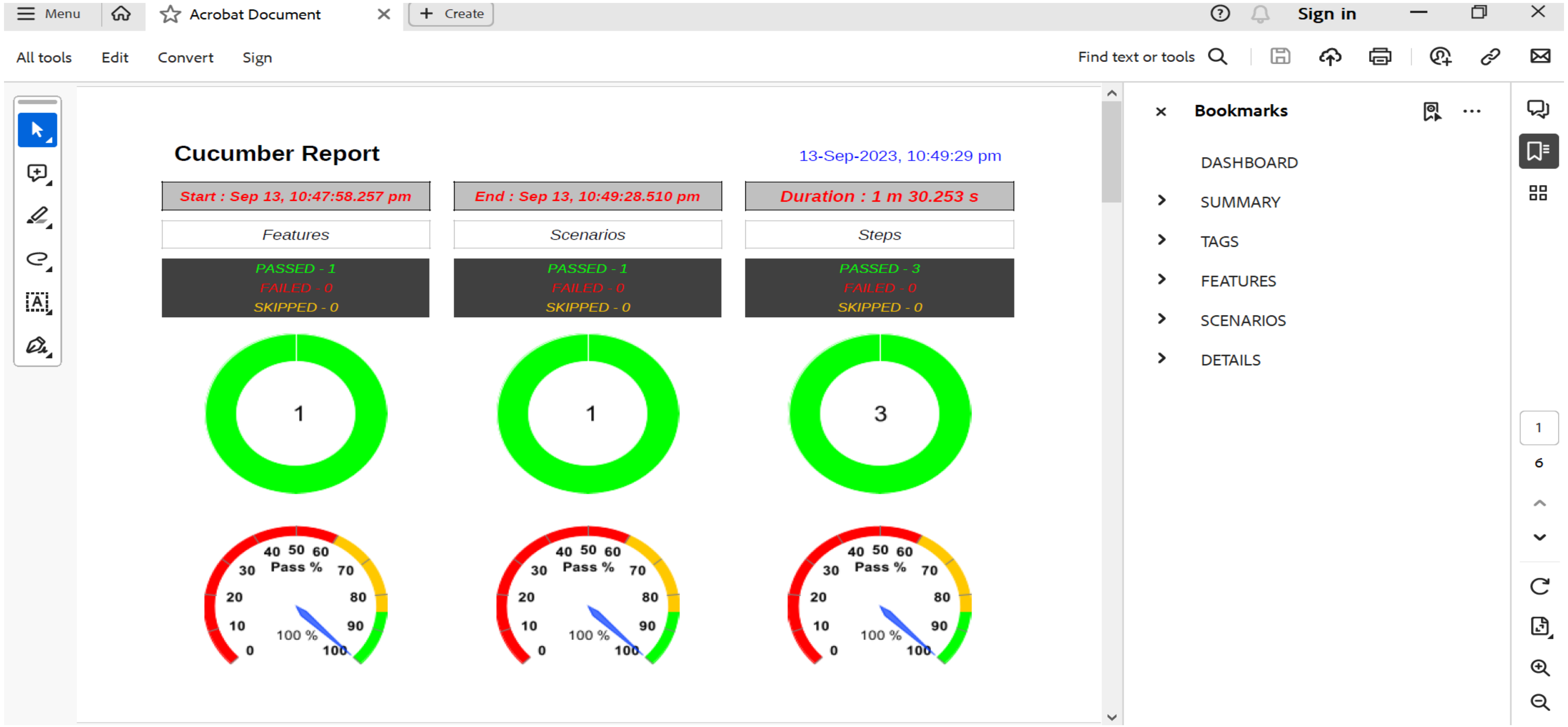
Login to HRM Application
Sep 13, 2023 10:47:58 pm Pass

Login to HRM Application

✓ Scenario: Login with valid credentials 00 01 30.246
@ValidCredentials

✓ Given User is on HRMLogin page "https://opensource-demo.orangehrmlive.com/"
✓ When User enters username and password
✓ Then User should be able to login sucessfully and new page open

Output



Cucumber Report

13-Sep-2023, 10:49:29 pm

Start : Sep 13, 10:47:58.257 pm	End : Sep 13, 10:49:28.510 pm	Duration : 1 m 30.253 s
Features	Scenarios	Steps
PASSED - 1 FAILED - 0 SKIPPED - 0	PASSED - 1 FAILED - 0 SKIPPED - 0	PASSED - 3 FAILED - 0 SKIPPED - 0
1	1	3
Pass % 100 %	Pass % 100 %	Pass % 100 %