Reports Using Extent Reports

import java.io.File;

 import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.SkipException;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.relevantcodes.extentreports.ExtentReports;

import com.relevantcodes.extentreports.ExtentTest;

import com.relevantcodes.extentreports.LogStatus;

public class ExtentReportsClass{

 ExtentReports extent;

 ExtentTest logger;

 @BeforeTest

 public void startReport(){

 //ExtentReports(String filePath,Boolean replaceExisting)

 //filepath - path of the file, in .htm or .html format - path where your report needs to generate.

//replaceExisting - Setting to overwrite (TRUE) the existing file or append to it

 //True (default): the file will be replaced with brand new markup, and all existing data will be lost. Use

this option to create a brand new report

//False: existing data will remain, new tests will be appended to the existing report. If the supplied path does not exist, a new file will be created.

```java
extent = new ExtentReports (System.getProperty("user.dir") +"/test-output/STMExtentReport.html", true);
//extent.addSystemInfo("Environment","Environment Name")
extent
        .addSystemInfo("Host Name", "SoftwareTesting")
        .addSystemInfo("Environment", "Automation Testing")
        .addSystemInfo("User Name", "TestEngineer");
        //loading the external xml file (i.e., extent-config.xml) that was placed under the base directory
        //You could find the xml file below. Create xml file in your project and copy paste the code
mentioned below
        extent.loadConfig(new File(System.getProperty("user.dir")+"\\extent-config.xml"));
}
@Test
public void passTest(){
//extent.startTest("TestCaseName", "Description")
//TestCaseName – Name of the test
//Description – Description of the test
//Starting test
logger = extent.startTest("passTest");
Assert.assertTrue(true);
//To generate the log when the test case is passed
logger.log(LogStatus.PASS, "Test Case Passed is passTest"); }
```

```java
@Test

public void failTest(){

logger = extent.startTest("failTest");

Assert.assertTrue(false);

logger.log(LogStatus.PASS, "Test Case (failTest) Status is passed");

}


@Test

public void skipTest(){

logger = extent.startTest("skipTest");

throw new SkipException("Skipping - This is not ready for testing ");

}

@AfterMethod

public void getResult(ITestResult result){

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

logger.log(LogStatus.FAIL, "Test Case Failed is "+result.getName());

logger.log(LogStatus.FAIL, "Test Case Failed is "+result.getThrowable());

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

logger.log(LogStatus.SKIP, "Test Case Skipped is "+result.getName());

}

// ending test

//endTest(logger) : It ends the current test and prepares to create a HTML report

extent.endTest(logger);

}
```

@AfterTest

public void endReport(){

//writing everything to document

//flush() - to write or update test information to your report.

      extent.flush();

      //Call close() at the very end of your session to clear all resources.

      //If any of your tests ended abruptly causing any side-affects (not all logs sent to ExtentReports,

information missing), this method will ensure that the test is still appended to the report with a warning

message.

      //You should call close() only once, at the very end (in @AfterSuite for example) as it closes the

underlying stream.

      //Once this method is called, calling any Extent method will throw an error.

      //close() - To close all the operations

      extent.close();

  }

}