package Tests;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Properties;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.testng.ITestContext;

import org.testng.ITestResult;

import org.testng.Reporter;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.BeforeSuite;

import Pages.HomePage;

import Pages.LoginPage;

import Utilities.DriverInstance;

import Utilities.ReadConfig;

public class BaseClass {

int stepNum;

public static WebDriver driver;

public ReadConfig readConfig;

public static Properties prop;

public static FileInputStream fis;

String browser;

String email;

String pass;

static String ssPath;

LoginPage loginPage;

HomePage homePage;

@BeforeSuite

public void initialSetup() {

readConfig = new ReadConfig();

browser = readConfig.getBrowser();

email = readConfig.getEmail();

pass = readConfig.getPassword();

ssPath = readConfig.getSSPath();

}

protected WebDriver launchBrowser(String browser) {

stepNum++;

ITestContext context = Reporter.getCurrentTestResult().getTestContext();

DriverInstance driverInst = new DriverInstance(browser);

driver = driverInst.getDriver();

context.setAttribute("driver", driver);

Reporter.log(stepNum + " Launch Browser");

return driver;

}

protected void getURL(WebDriver driver) {

stepNum++;

String url = readConfig.getURL();

driver.get(url);

Reporter.log(stepNum + " Get URL");

}

public File takeScreenshot(ITestResult result, WebDriver driver) {

String methodName = result.getName();

// Create a timestamp for the screenshot

String timestamp = new SimpleDateFormat("yyyyMMddHHmmss").format(new Date());

// Create the file name using the method name and timestamp

String fileName = methodName + "\_" + timestamp + ".png";

String filePath = ssPath + fileName;

TakesScreenshot ts = (TakesScreenshot) driver;

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

File destination = new File(filePath);

try {

FileUtils.copyFile(source, destination);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return destination;

}

@AfterSuite

public void tearDown() {

driver.quit();

}

}

package Utilities;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.reporter.ExtentSparkReporter;

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

import Tests.BaseClass;

public class TestNG\_Report extends BaseClass {

ExtentSparkReporter reporter;

ExtentReports report;

ExtentTest test;

public void setupReport() {

reporter = new ExtentSparkReporter("Cotecna\_Limbs.html");

report = new ExtentReports();

report.attachReporter(reporter);

}

public void setEnvirementSetup() {

report.setSystemInfo("Author", "Nitin Sehrawat");

report.setSystemInfo("OS", "Window");

report.setSystemInfo("Browser", prop.getProperty("browser"));

reporter.config().setDocumentTitle("Cotecna CPS Lims");

reporter.config().setReportName("Cotecna Lims Automation Report");

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

}

}

package Utilities;

import java.io.File;

import org.testng.ITestContext;

import org.testng.ITestListener;

import org.testng.ITestResult;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import Tests.BaseClass;

public class Listeners extends TestNG\_Report implements ITestListener {

@Override

public void onStart(ITestContext context) {

setupReport();

setEnvirementSetup();

System.out.println("Test case execution started");

}

@Override

public void onTestStart(ITestResult result) {

System.out.println(result.getName() + " Method Started");

}

@Override

public void onTestSuccess(ITestResult result) {

test = report.createTest(result.getName());

test.log(Status.PASS, MarkupHelper.createLabel("TestCase Pass", ExtentColor.GREEN));

if (result.getStatus() == 1) {

System.out.println("====== SUCCESS ======");

}

}

@Override

public void onTestFailure(ITestResult result) {

File ss = takeScreenshot(result, driver);

test = report.createTest(result.getName());

System.out.println(ss.getName());

// File ss = new File(filePath);

if (ss.exists()) {

test.fail("Captured Screenshot: "+ss.getName() + " " + test.addScreenCaptureFromPath(ss.getAbsolutePath()));

}

test.log(Status.FAIL, MarkupHelper.createLabel("TestCase Fail", ExtentColor.RED));

if (result.getStatus() == 2) {

System.out.println("====== Failure ======");

}

}

@Override

public void onTestSkipped(ITestResult result) {

test = report.createTest(result.getName());

test.log(Status.WARNING, MarkupHelper.createLabel("TestCase Fail Due to Timeout", ExtentColor.RED));

System.out.println("TestCase Skipped : " + result.getName() + " " + result.getStatus());

}

@Override

public void onTestFailedWithTimeout(ITestResult result) {

test = report.createTest(result.getName());

test.log(Status.WARNING, MarkupHelper.createLabel("TestCase Fail Due to Timeout", ExtentColor.RED));

System.out.println(

"====== TestCase Failed due to Timeout : " + result.getName() + " " + result.getStatus() + " ======");

}

@Override

public void onFinish(ITestContext context) {

if (context.getFailedTests().size() == 0) {

System.out.println("=============== All Test Cases Executed Successfully ===============");

} else {

System.out.println("====== Total Test Case Failed : " + context.getFailedTests().size() + " ======");

}

report.flush();

}

}

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

<suite name="Suite">

<listeners>

<listener class-name="Utilities.Listeners"></listener>

</listeners>

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

<classes>

<class name="Tests.TestCase1" />

</classes>

</test> <!--

Test -->

</suite> <!--

Suite -->