07-MAR-2020 Log4j

----------- =====

Log4j has 3 main components:

(i) Appender

--------

Responsible for publishing logging information to various destinations like Console/File/DB/SMTP (Simple Mail Transfer Protocol) (mail)

(ii) Layouts

------

Responsible for formatting logging information in different styles.

(iii) Logger

------

It is a class provided in Log4j

Responsible for capturing logging information

We have to initialize one logger object for each Java class

Syntax:

------

Logger obj= Logger.getLogger(CurrentClassName.class);

\*\*\* We use logger methods/levels to generate log statements like trace,debug,info,warn,error,fatal,.... etc

Ex: logger.info(msg);

logger.error(msg);

logger.debug(msg);

logger.fatal(msg);

Note:

Appender and layouts are maintained in " log4j.properties" file, where as logger class we use in the script.

Property Configurator:

---------------------

This class is used to initialize log4j properties file

Syntax:

------

Property Configurator.configure(property file name);

Working with log4j:

------------------

Step 1: Create new java project

Step 2: Configure Log4j library to java project

Navigation:

Download log4j.Jar file

http://www-us.apache.org/dist/logging/log4j/1.2.17/log4j-1.2.17.zip

extract zip file

configure "log4j-1.2.17" jar file to java project

Note:

based on requirement we can also configure WD jars and TestNg jars

Step 3:create log4j.properties file in project Home directory

navigation:

Right click on "java project"

New

File

enter file name as "log4j.properties"

Click on Finish

Step 4: write following code into log4j.properties file

log4j properties file configuration script

----------------------------------------

# Here we have defined root logger

log4j.rootLogger=DEBUG,CONSOLE,FILE

#Console configuration

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.ConversionPattern= %d{dd-MM-yyyy hh:mm:ss} %c %M -%m%n

#File configuration

log4j.appender.FILE=org.apache.log4j.FileAppender

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE.file=./TestResults/AppLogs.log

log4j.appender.FILE.layout.ConversionPattern= %d{dd-MM-yyyy hh:mm:ss} %c %M -%m%n

log4j.appender.FILE.Append=true

==================================================================================================================================================================

Step 5: create TestNg class and use Logger class to pass messages into log file

and console

package testng.framework;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class Log4jDemo {

private Logger logger;

public WebDriver driver;

@Test

public void setUp() {

logger=Logger.getLogger("Log4jDemo.class");

PropertyConfigurator.configure("log4j.properties");

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

logger.info("Browser executable file running");

driver=new ChromeDriver();

logger.info("Browser launched");

driver.get("https://opensource-demo.orangehrmlive.com/");

logger.info("Application opened");

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

}

}

====================================================================================================================================================================

3. Extent Reports v4 in Selenium Webdriver

=======================================

Extent Reports (Tool) is also used as Selenium Reporting Tool.

We can create interactive and detailed reports using Extent framework such as events,pie charts, screenshots, tags, devices,author.... etc

Steps to Generate Extent Reports in Selenium:

============================================

Step 1: Create a TestNG project in eclipse

Step 2: Download Extent reports Version4 JAR files

url: http://extentreports.com/community/

Step 3: Add the downloaded library (jar) files to your project (external jar file and Dependency jars)

\*\* Classes in Extent Reports jars:

==============================

(i) Extent Html Reporter:

--------------------

Using this class we can customize UI of the Reports like Report id, Report name, Theme,.... etc

(ii) Extent Reports:

--------------

To entry in reports

(iii) Extent Test:

-----------

Used to update reports

Step 4: Create Class using Extent Report classes to verify OrangeHRM page title and also provide screenshot of page with date stamp in Reports

package sample.pack;

import java.io.File;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.io.FileHandler;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

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

public class ExtentDemo {

ExtentHtmlReporter htmlReporter;

ExtentReports extent;

ExtentTest test;

WebDriver driver;

@BeforeClass

public void extentInit() {

//to specify the location

htmlReporter=new ExtentHtmlReporter("./TestResults//ExtentReporter.html");

// Title of the report

htmlReporter.config().setDocumentTitle("Automation Report");

//Name of the report

htmlReporter.config().setReportName("Functional Report");

//To Specify Theme like STANDARD/DARK

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

//create ExtentReports object and attach Reports

extent=new ExtentReports();

extent.attachReporter(htmlReporter);

extent.setSystemInfo("Hostname", "LocalHost");

extent.setSystemInfo("OS", "Windows10");

extent.setSystemInfo("Tester Name", "Madhukar");

extent.setSystemInfo("Browser", "Chrome");

//create ExtentTest object and specify Report Title

test=extent.createTest("OrangeHRM\_TitleTest");

}

@BeforeMethod

public void setUp()

{

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

driver = new ChromeDriver();

test.log(Status.INFO,"Chrome browser launched");

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

driver.get("https://opensource-demo.orangehrmlive.com/");

test.log(Status.INFO,"Navigated Orange HRM Application");

}

@Test

public void titleTest() throws IOException {

String pgTitle=driver.getTitle();

Assert.assertEquals("OrangeHRM", pgTitle);

test.log(Status.PASS, "Expected title exist");

File f=((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

//to get Date stamping with file

Date dt= new Date(); //it will return system date

DateFormat df1= new SimpleDateFormat("dd\_MM\_yyyy\_HH\_MM\_ss");//to convert into required format

//to save screenshot with date stamping

FileHandler.copy(f,new File("D:\\titleScreenshot"+df1.format(dt)+".png")) ;

test.addScreenCaptureFromPath("D:\\titleScreenshot"+df1.format(dt)+".png");// adding screen shot

}

@AfterMethod

public void tearDown() {

driver.close();

}

@AfterClass

public void endReport()

{

extent.flush();

}

}

=================================================================================================================================================================

Interview Question

How to take screenshot?

Using TakesScreenshot Interface Class

How to pass the screenshot value into report

Using Extent Reports

========================================================================END OF CLASS================================================================================