**Creating our own report:**

If you see the default report, you can understand that it will show the method names instead of actual test cases. It will be ok in other models as we write the tests like methods and we name the methods like test case names. But in case of keyword driven framework, the method that we pass the data is always same and in our case it is prepareKeywords. So The report always show that the test case name is prepareKeywords. So we better prepare our own report which shows the test case names.

For this we need to capture the test case status and the error messages so that we can use them in reporting. Our report will be similar to emailable report but it shows the actual test cases.

We will do this using listeners (ITestListener) concept.

First, we will create different maps to hold different values. Since we already have the class driverHodling which holds a map, we define other maps under the same class.

**driverHolding class:**

**package** executionEngine;

**import** java.util.HashMap;

**import** java.util.Map;

**import** org.openqa.selenium.WebDriver;

**public** **class** driverHolding {

**public** **static** Map<String, WebDriver> *mapdriver* = **new** HashMap<String, WebDriver>();

**public** **static** Map<String, String> *mapteststarttime* = **new** HashMap<String, String>();

**public** **static** Map<String, String> *maptestendtime* = **new** HashMap<String, String>();

**public** **static** Map<String, String> *mapteststatus* = **new** HashMap<String, String>();

**public** **static** Map<String, String> *maptestexceptions* = **new** HashMap<String, String>();

}

We defined four more maps here.

mapteststarttime – This is an hashmap of two strings. The first string is test case no. like TC01, TC02 etc and the second one is start time of the test case. This we will use it in onTestStart() method of the listener as this method will be invoked when the test starts. We will get the test case number as follows.

String testcase = (String)(result.getParameters())[0];

We can get all the parameters as result.getParameters(). Since the test case name we are passing it as first parameter, we get it by using the index value [0].

maptestendtime - We use it in the methods onTestSuccess() and onTestFailure(). Since whenever the test is passed or failed, these methods will be invoked.

mapteststatus - We use it in the methods onTestSuccess() and onTestFailure() to capture the status of the tests.

maptestexceptions – We use it in the method onTestFailure to capture the exception.

**Listeners Class:**

**package** executionEngine;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.io.StringWriter;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Collections;

**import** java.util.Date;

**import** java.util.Map;

**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.openqa.selenium.remote.RemoteWebDriver;

**import** org.openqa.selenium.remote.SessionId;

**import** org.testng.ITestContext;

**import** org.testng.ITestListener;

**import** org.testng.ITestResult;

**public** **class** Listeners **implements** ITestListener{

@Override

**public** **void** onTestStart(ITestResult result) {

// **TODO** Auto-generated method stub

DateFormat dateFormat = **new** SimpleDateFormat("HH:mm:ss");

Date date = **new** Date();

String currenttime = dateFormat.format(date);

String testcase = (String)(result.getParameters())[0];

executionEngine.driverHolding.*mapteststarttime*.put(testcase, currenttime);

}

@Override

**public** **void** onTestSuccess(ITestResult result) {

// **TODO** Auto-generated method stub

String testcase = (String)(result.getParameters())[0];

driverHolding.*mapdriver*.remove(testcase+"driver");

executionEngine.driverHolding.*mapteststatus*.put(testcase, "Pass");

DateFormat dateFormat = **new** SimpleDateFormat("HH:mm:ss");

Date date = **new** Date();

String currenttime = dateFormat.format(date);

executionEngine.driverHolding.*maptestendtime*.put(testcase, currenttime);

System.***out***.println(testcase + " passed.");

}

@Override

**public** **void** onTestFailure(ITestResult result){

// **TODO** Auto-generated method stub

String testcase = (String)(result.getParameters())[0];

WebDriver driver = driverHolding.*mapdriver*.get(testcase+"driver");

executionEngine.driverHolding.*mapteststatus*.put(testcase, "Fail");

StringWriter sw = **new** StringWriter();

result.getThrowable().printStackTrace(**new** PrintWriter(sw));

String stacktrace = sw.toString();

executionEngine.driverHolding.*maptestexceptions*.put(testcase, stacktrace);

DateFormat dateFormat = **new** SimpleDateFormat("HH:mm:ss");

Date date = **new** Date();

String currenttime = dateFormat.format(date);

executionEngine.driverHolding.*maptestendtime*.put(testcase, currenttime);

driver.quit();

driverHolding.*mapdriver*.remove(testcase+"driver");

System.***out***.println(testcase + " failed.");

}

@Override

**public** **void** onTestSkipped(ITestResult result) {

// **TODO** Auto-generated method stub

String testcase = (String)(result.getParameters())[0];

executionEngine.driverHolding.*mapteststatus*.put(testcase, "Skipped");

System.***out***.println(testcase + " skipped.");

}

@Override

**public** **void** onTestFailedButWithinSuccessPercentage(ITestResult result) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** onStart(ITestContext context) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** onFinish(ITestContext context) {

String suitename = context.getCurrentXmlTest().getSuite().getName();

String testname = context.getName();

**try** {

createHTMLReport.*createReport*(suitename, testname);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

In the onTestStart() method we get the current time and test case and add it to the map mapteststarttime.

In the onTestSuccess() method we get the passed test case and current time. We add the test case name and the value “pass” to the map mapteststatus as this method will only be invoked when a test is passed. We add the test case name and the current time to the map maptestendtime.

In the onTestFailure() method we get the failed test case and current time. We add the test case name and the value “fail” to the map mapteststatus as this method will only be invoked when a test is failed. We add the test case name and the current time to the map maptestendtime.

In the onTestSkipped() method we get the skipped test case and current time. We add the test case name and the value “skipped” to the map mapteststatus. Since the test is skipped we don’t add the end time here.(The test won’t be executed at all)

In the onFinish() method, since this method will be invoked when all the execution is finished we call the createReport() method of createHTMLReport class by passing the suitename and test name.

Let’s create the class createHTMLReport and createReport.