**Reference -** [**https://github.com/sahajamit/cucumber-jvm-testngSETUP**](https://github.com/sahajamit/cucumber-jvm-testngSETUP) **PROCEDURE**1) CREATE PROJECT STRUCTURE AS POSTED BELOW.2) Set the POM dependencies as seen below.3) mvn clean install -U -Dmaven.test.skip=true. Perform build project with tests skipped and ensure build success -4) Set the testng.xml to correct number of thread count and set the listener, test classes pointing to right browsers.5) Perform build project with tests skipped and ensure build success.4) mvn clean test5) right click testng.xml and perform debug action if exceptions are reported – Make sure the test feature file classes are compiled and pointing to the right suite.**TESTNG RULES 1) Maximum Concurrency - one scenario feature file per runner 2) Custom test suite name unique for a runner test class 3) Never break the previous logic flow4) Report TestNG to be handled wisely5) Thread count = three four6) rebuild and keep the test runner classes compiled - Call the session starter line directly in feature file and keep tags in runner7) 300 test cases in multiple browsers in multiple operating system 8) Run the TestNG.xml using debug mode to clear off the exceptions - right click the testing.xml and run in debug modeA1 LISTENER TESTNG CUCUMBER RUNNER FEATURE FILE STEP DEFINITION CONNECTIONMAVEN PROJECT STRUCTURE lib selenium-server-standalone-2.53.1.jar.zipsrc/main/java/com/cucumber/testng/examples/FeatureObject.javasrc/test/java/com/cucumber/testng/examples/TestNGExecutionListener.java // ListenerBaseStepDef1.java // Step Definition RunCukesTestNGRunner.java // Runner with AbstractTestNGCucumberTestssrc/test/resources/utils/LocalDriverFactory.javaLocalDriverManager.javasrc/test/resources/com.cucumber.testng.examples/ThreadLocalDemo.javafeature1.featurefeature2.featureREADME.MD pom.xmltestng-cucumber-runner-connection.xmltestng-multi-browser.xml**[cucumber-jvm-testng-integration](https://github.com/sahajamit/cucumber-jvm-testng-integration)/[src](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src)/[test](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test)/[java](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java)/[com](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com)/[cucumber](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber)/[testng](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber/testng)/[examples](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber/testng/examples)/**GenerateReport.java**package com.cucumber.testng.examples;import net.masterthought.cucumber.ReportBuilder;import java.io.File;import java.util.ArrayList;import java.util.List;public class GenerateReport { public static void GenerateMasterthoughtReport(){ try{ String RootDir = System.getProperty("user.dir"); File reportOutputDirectory = new File("target/Masterthought"); List<String> list = new ArrayList<String>(); list.add("target/cucumber1.json"); list.add("target/cucumber2.json"); String pluginUrlPath = ""; String buildNumber = "1"; String buildProject = "cucumber-jvm"; boolean skippedFails = true; boolean pendingFails = true; boolean undefinedFails = true; boolean missingFails = true; boolean flashCharts = true; boolean runWithJenkins = false; boolean highCharts = false; boolean parallelTesting = true; boolean artifactsEnabled = false; String artifactConfig = ""; ReportBuilder reportBuilder = new ReportBuilder(list, reportOutputDirectory, pluginUrlPath, buildNumber, buildProject, skippedFails, pendingFails, undefinedFails, missingFails, flashCharts, runWithJenkins, highCharts, parallelTesting); reportBuilder.generateReports(); }catch(Exception e){ e.printStackTrace(); } }}[cucumber-jvm-testng-integration](https://github.com/sahajamit/cucumber-jvm-testng-integration)/[src](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src)/[test](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test)/[java](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java)/[com](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com)/[cucumber](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber)/[testng](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber/testng)/[examples](https://github.com/sahajamit/cucumber-jvm-testng-integration/tree/master/src/test/java/com/cucumber/testng/examples)/**RunCukesTestNGRunner.java**package com.cucumber.testng.examples;import cucumber.api.CucumberOptions;import cucumber.api.testng.AbstractTestNGCucumberTests;@CucumberOptions(features = "src/test/resources/feature1.feature", tags = "@calculator", format = { "pretty", "html:target/site/cucumber-pretty", "rerun:target/rerun.txt", "json:target/cucumber1.json" })public class RunCukesTestNGRunner extends AbstractTestNGCucumberTests {}[cucumber-jvm-testng-integration](https://github.com/sahajamit/cucumber-jvm-testng-integration)/**testng.xml listener runner thread count parallel setting** <?xml version="1.0" encoding="UTF-8"?><!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>"><suite name="Cucumber Parallel Test" verbose="1" thread-count="3" parallel="tests" configfailurepolicy="continue"> <listeners> <listener class-name="com.cucumber.testng.examples.TestNGExecutionListener"></listener> </listeners> <test name="Cucumber TestNG Test 1" annotations="JDK" preserve-order="true"> <classes> <class name="com.cucumber.testng.examples.RunCukesTestNGRunner"/> </classes> </test> <test name="Cucumber TestNG Test 2" annotations="JDK" preserve-order="true"> <classes> <class name="com.cucumber.testng.examples.RunCukesTestNGRunner"/> </classes> </test></suite>**cucumber-jvm-testng-integration/src/test/java/com/cucumber/testng/examples/TestNGExecutionListener.java**package com.cucumber.testng.examples;import org.testng.IExecutionListener;public class TestNGExecutionListener implements IExecutionListener { @Override public void onExecutionStart() { System.out.println("TestNG is staring the execution"); } @Override public void onExecutionFinish() { System.out.println("Generating the Masterthought Report"); GenerateReport.GenerateMasterthoughtReport(); System.out.println("TestNG has finished, the execution"); }}**POM.XML - cucumber-jvm-testing cucumber-java cucumber-jvm-deps cucumber-testng cucumber-reporting testng maven-compiler-plugin maven-surefire-plugin**<?xml version="1.0" encoding="UTF-8"?><project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 <http://maven.apache.org/xsd/maven-4.0.0.xsd>"> <modelVersion>4.0.0</modelVersion> <groupId>cucumber-jvm-testng</groupId> <artifactId>cucumber-jvm-testng</artifactId> <version>1.0-SNAPSHOT</version> <dependencies> <dependency> <groupId>info.cukes</groupId> <artifactId>cucumber-java</artifactId> <version>1.2.4</version> <scope>test</scope> </dependency> <dependency> <groupId>info.cukes</groupId> <artifactId>cucumber-jvm-deps</artifactId> <version>1.0.5</version> <scope>test</scope> </dependency> <dependency> <groupId>info.cukes</groupId> <artifactId>cucumber-testng</artifactId> <version>1.2.4</version> <scope>compile</scope> <exclusions> <exclusion> <groupId>junit</groupId> <artifactId>junit</artifactId> </exclusion> </exclusions> </dependency> <dependency> <groupId>org.testng</groupId> <artifactId>testng</artifactId> <version>6.9.8</version> <scope>test</scope> </dependency> <dependency> <groupId>net.mastrthought</groupId> <artifactId>cucumber-reporting</artifactId> <version>0.6.0</version> </dependency><!-- <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-ie-driver> --> <dependency> <groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-ie-driver</artifactId> <version>3.0.1</version> </dependency> <!-- <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java> --> <dependency> <groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-java</artifactId> <version>3.0.1</version> </dependency> <!-- <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-remote-driver> --> <dependency> <groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-remote-driver</artifactId> <version>3.0.1</version> </dependency> <!-- <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver> --> <dependency> <groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-chrome-driver</artifactId> <version>3.0.1</version> </dependency> </dependencies> <build> <plugins>**Generating Cucumber runner per feature file** <plugin> <groupId>com.github.temyers</groupId> <artifactId>**cucumber-jvm-parallel-plugin**</artifactId> <version>**2.1.0<**/version> <executions> <execution> <id>**generateRunners**</id> <phase>generate-test-sources</phase> <goals> <goal>**generateRunners**</goal> </goals> <configuration> <!-- Mandatory --> <!-- comma separated list of package names to scan for glue code --> <glue>**foo, bar**</glue> <!-- These are optional, with the default values --> <!-- Where to output the generated tests --> <outputDirectory>**${project.build.directory}/generated-test-sources/cucumber**</outputDirectory> <!-- The directory, which must be in the root of the runtime classpath, containing your feature files. --> <featuresDirectory>**src/test/resources/features/**</featuresDirectory> <!-- Directory where the cucumber report files shall be written --> <cucumberOutputDir>**target/cucumber-parallel**</cucumberOutputDir> <!-- comma separated list of output formats --> <format>json</format> <!-- CucumberOptions.strict property --> <strict>true</strict> <!-- CucumberOptions.monochrome property --> <monochrome>true</monochrome> <!-- The tags to run, maps to CucumberOptions.tags property --> <tags></tags> <!-- If set to true, only feature files containing the required tags shall be generated. --> <filterFeaturesByTags>false</filterFeaturesByTags> <!-- Generate TestNG runners instead of JUnit ones. --> <useTestNG>false</useTestNG> <!-- The naming scheme to use for the generated test classes. One of 'simple' or 'feature-title' --> <namingScheme>simple</namingScheme> <!-- The class naming pattern to use. Only required/used if naming scheme is 'pattern'.--> <namingPattern>**Parallel{c}IT**</namingPattern> <!-- One of [SCENARIO, FEATURE]. SCENARIO generates one runner per scenario. FEATURE generates a runner per feature. --> <parallelScheme>SCENARIO</parallelScheme> <!-- Specify a custom package name for generated sources. Default is no package.--> <packageName></packageName> </configuration> </execution> </executions></plugin> <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-surefire-plugin</artifactId> <version>2.18.1</version> <configuration> <suiteXmlFiles> <suiteXmlFile>testng.xml</suiteXmlFile> </suiteXmlFiles> <forkCount>3</forkCount> <reuseForks>true</reuseForks> <argLine>-Xmx1024m -XX:MaxPermSize=256m</argLine> </configuration> </plugin> <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-compiler-plugin</artifactId> <configuration> <source>1.6</source> <target>1.6</target> </configuration> </plugin> </plugins> </build></project>**Basic Integration**To have a simple integration you just need to create a runner class and just extend it from AbstractTestNGCucumberTests.mvn clean install -U -Dmaven.test.skip=truemvn clean testright click testng.xml and perform debug action**Advanced Integration**To have full-fledged integration with TestNg where you can use testng xml along with parallel execution. Here is the sample code:@CucumberOptions(features = "src/test/resources/com.cucumber.testng.examples/date\_calculator1.feature", plugin = "json:target/cucumber2.json")public class RunCukesByCompositionGrp1\_Test2 { @Test(groups = "examples-testng", description = "Example of using TestNGCucumberRunner to invoke Cucumber") public void runCukes() { new TestNGCucumberRunner(getClass()).runCukes(); }}

Or use AbstractTestNGCucumberTestsRunCukesTest.javapackage com.cucumber.testng.examples;import cucumber.api.CucumberOptions;import cucumber.api.testng.AbstractTestNGCucumberTests;@CucumberOptions(features = "src/test/resources/com.cucumber.testng.examples/date\_calculator1.feature", tags = "@calculator", format = { "pretty", "html:target/site/cucumber-pretty", "rerun:target/rerun.txt", "json:target/cucumber1.json" })public class RunCukesTest extends AbstractTestNGCucumberTests {}To see this whole thing running simply checkout this projecy and run this command:mvn clean test**Reference -** <https://rationaleemotions.wordpress.com/2013/07/31/parallel-webdriver-executions-using-testng/>**ADDITION OF MULTIPLE BROWSER FACTORY LAUNCH** **LocalDriverFactory.java** import org.openqa.selenium.WebDriver; import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.firefox.FirefoxDriver; import org.openqa.selenium.ie.InternetExplorerDriver;class LocalDriverFactory { static WebDriver createInstance(String browserName) { WebDriver driver = null; if (browserName.toLowerCase().contains("firefox")) { driver = new FirefoxDriver(); return driver; } if (browserName.toLowerCase().contains("internet")) { driver = new InternetExplorerDriver(); return driver; } if (browserName.toLowerCase().contains("chrome")) { driver = new ChromeDriver(); return driver; } return driver; } }**LocalDriverManager.java** import org.openqa.selenium.WebDriver;public class LocalDriverManager { private static ThreadLocal<WebDriver> webDriver = new ThreadLocal<WebDriver>();public static WebDriver getDriver() { return webDriver.get(); }static void setWebDriver(WebDriver driver) { webDriver.set(driver); } } **WebDriverListener.java** import org.openqa.selenium.WebDriver; import org.testng.IInvokedMethod; import org.testng.IInvokedMethodListener; import org.testng.ITestResult;public class WebDriverListener implements IInvokedMethodListener {@Override public void beforeInvocation(IInvokedMethod method, ITestResult testResult) { if (method.isTestMethod()) { String browserName = method.getTestMethod().getXmlTest().getLocalParameters().get("browserName"); WebDriver driver = LocalDriverFactory.createInstance(browserName); LocalDriverManager.setWebDriver(driver); } }@Override public void afterInvocation(IInvokedMethod method, ITestResult testResult) { if (method.isTestMethod()) { WebDriver driver = LocalDriverManager.getDriver(); if (driver != null) { driver.quit(); } } } } **ThreadLocalDemo.java** import org.testng.annotations.Test;public class ThreadLocalDemo { @Test public void testMethod1() { invokeBrowser("http://www.gmail.com"); }@Test public void testMethod2() { invokeBrowser("http://www.google.com");}private void invokeBrowser(String url) { System.out.println("Thread id = " + Thread.currentThread().getId()); System.out.println("Hashcode of webDriver instance = " + LocalDriverManager.getDriver().hashCode()); LocalDriverManager.getDriver().get(url);} }**testng-multi-browser.xml** <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>"> <suite name="Suite" parallel="methods"> <listeners> <listener class-name="WebDriverListener"></listener> </listeners> <test name="Test"> <parameter name="browserName" value="firefox"></parameter> <classes> <class name="organized.chaos.ThreadLocalDemo" /> </classes> </test> <!-- Test --> <test name="Test"> <parameter name="browserName" value="chrome"></parameter> <classes> <class name="organized.chaos.ThreadLocalDemo" /> </classes> </test> <!-- Test --><test name="Test"> <parameter name="browserName" value="iexplore"></parameter> <classes> <class name="organized.chaos.ThreadLocalDemo" /> </classes> </test> <!-- Test --> </suite> <!-- Suite --> mvn clean install -U mvn clean testtestng.xml right click debug - for debug configurations [TestNG] Running: /githome/PlayGround/testbed/src/test/resources/threadLocalDem.xmlThread id = 10 Hashcode of webDriver instance = 1921042184 Thread id = 9 Hashcode of webDriver instance = 2017986718=============================================== Suite Total tests run: 2, Failures: 0, Skips: 0 ==============================================