A yellow and white sign

Description automatically generated

To run scenarios in parallel mode, then junit wont work. We need testng for that.

A yellow sign with white text

Description automatically generated

# Open pom and add this dependencies-

A screenshot of a computer code

Description automatically generated

# Another plugin to be added-

A screen shot of a computer code

Description automatically generated

# Slight changes in folder structure for testng-

Create a package called parallel in “src/test/java”.

And move the step def into it. settings will be as below.

A screenshot of a computer

Description automatically generated

Move the hooks also to parallel package.

Note-

Whatever feature files are present, they should be present under same package where step def is located.

runner files and step def needs to be under same package as per the cucumber documentation, so runner can pick the step def and app features correctly.

# Parallel folder created for features in src/test/resources-

A screenshot of a computer

Description automatically generated

We can give any name; not necessary the name should match the one with the step def.

# Three main folders-

Src/main/java – for page factory, driver factory, page class, utility, page class.

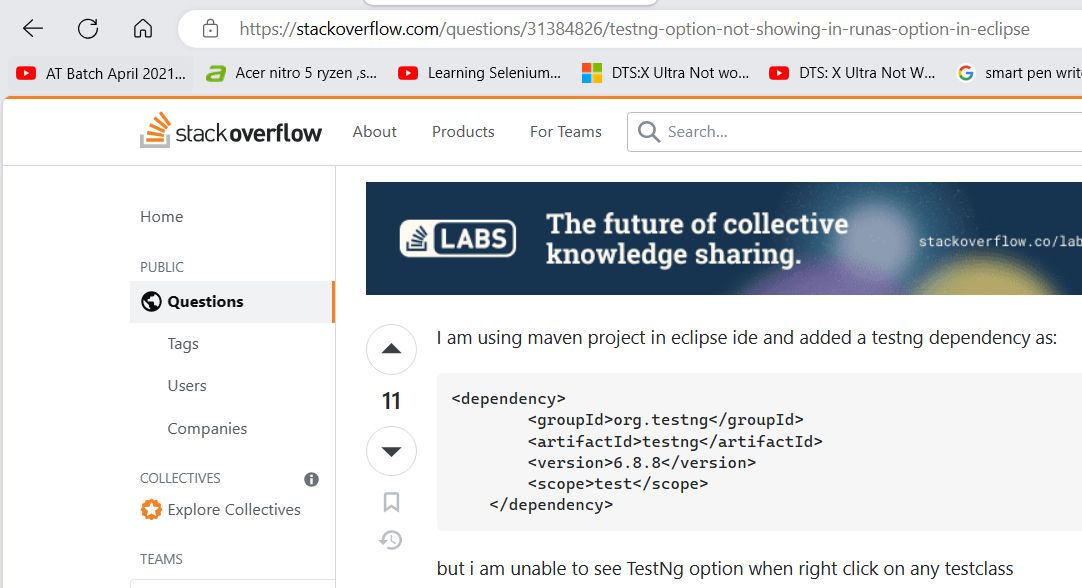
Src/test/java – step def and runner and hooks. All should be under same folder or package in this module.

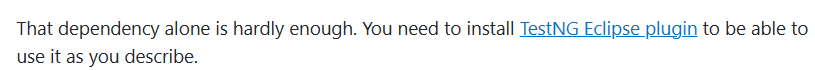
Src/test/resources – a folder which will have all the app features.

Once all done, right click on project – maven – update project.

Note-

Testng will not be there.





A screenshot of a computer program

Description automatically generated

[TestNG option not showing in RunAs option in Eclipse - Stack Overflow](https://stackoverflow.com/questions/31384826/testng-option-not-showing-in-runas-option-in-eclipse)

Run as testng-

A screenshot of a computer

Description automatically generated

# You may get this exception when you run first time-

|  |
| --- |
| Exception in thread "main" java.lang.UnsupportedClassVersionError: org/testng/remote/RemoteTestNG has been compiled by a more recent version of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0  at java.lang.ClassLoader.defineClass1(Native Method)  at java.lang.ClassLoader.defineClass(ClassLoader.java:763)  at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)  at java.net.URLClassLoader.defineClass(URLClassLoader.java:468)  at java.net.URLClassLoader.access$100(URLClassLoader.java:74)  at java.net.URLClassLoader$1.run(URLClassLoader.java:369)  at java.net.URLClassLoader$1.run(URLClassLoader.java:363)  at java.security.AccessController.doPrivileged(Native Method)  at java.net.URLClassLoader.findClass(URLClassLoader.java:362)  at java.lang.ClassLoader.loadClass(ClassLoader.java:424)  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)  at java.lang.ClassLoader.loadClass(ClassLoader.java:357)  at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:495) |

To solve this error we need to upgrade the java version to 11.

How to do.

Go to google and search for download jdk 11.

Click on oracle link and download.

Install the required version.

Open this pc.

Properties.

Advance system settings.

Environment variables.

Variable name will be JAVA\_HOME.

Replace the value - C:\Program Files\Java\jdk-11

Edit the path. Value should be %JAVA\_HOME%\bin

Now one other thing.

Go to this path

A screenshot of a computer

Description automatically generated

Copy this path also in advance system settings inside path variable.

This path should also be present in path variable-

C:\Program Files\Common Files\Oracle\Java\javapath

Now open cmd.

Check java version and maven version. They should be mapped to the java version (latest one).

A computer screen shot of a program

Description automatically generated

Open eclipse.

Sometimes eclipse will not be compatible with java 11 (eclipse neon especially). Download the latest version of eclipse.

Change in pom file.

A computer code with text

Description automatically generated

Go and check the compiler version.

A screenshot of a computer code

Description automatically generated

Run the code using testng now and check-

Delete all the test output folders.

Right click on project and update it using maven.

Run as testng-

A screenshot of a computer

Description automatically generated

Open the report of cucumber-

A screenshot of a computer

Description automatically generated

Failed screenshot-

A screenshot of a computer

Description automatically generated

Report getting generated-

A screenshot of a computer

Description automatically generated

But extent spark html and extent spark pdf are not getting generated, not sure why. For Naveen it was working.

# To run with maven and also with Jenkins we use the includes keyword-

A screen shot of a computer

Description automatically generated

Open cmd.

Go to the project location.



A screenshot of a computer program

Description automatically generated

Same old cucumber report-

A screenshot of a computer

Description automatically generated

Now pdf report generated-

A screenshot of a computer

Description automatically generated

Thread report-

A screenshot of a computer

Description automatically generated

Three output folders created after running from maven-

A close up of text

Description automatically generated

# Project structure-

A screenshot of a computer program

Description automatically generated

# Codes from this chapter-

|  |
| --- |
| Runner file-  package parallel;  import org.junit.runner.RunWith;  import org.testng.annotations.DataProvider;  import io.cucumber.junit.Cucumber;  import io.cucumber.junit.CucumberOptions;  import io.cucumber.testng.AbstractTestNGCucumberTests;  @RunWith(Cucumber.class)  @CucumberOptions(features = { "src/test/resources/parallel" },  glue = { "parallel" },  plugin = { "pretty",  "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",  "timeline:test-output-thread/" }  )  public class ParalelRunner extends AbstractTestNGCucumberTests {  /\*\*  \* we need to override a method from abstracttestngcucumbertests class  \* parallel=true means run in parallel, else it will run one by one.  \*  \* @return 2d array containing all scenarios.  \*/  @Override  @DataProvider(parallel = true)  public Object[][] scenarios() {  return super.scenarios();  }  } |

|  |
| --- |
| Pom file-  <?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>CucumberPOMSeriesByKaranJulySeven</groupId>  <artifactId>CucumberPOMSeriesByKaranJulySevenA</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>CucumberPOMSeries</name>  <!-- FIXME change it to the project's website -->  <url>http://www.example.com</url>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <java.version>11</java.version>  <junit.version>4.13.1</junit.version>  <cucumber.version>6.9.0</cucumber.version>  <maven.compiler.version>3.8.1</maven.compiler.version>  <maven.surefire.version>2.22.2</maven.surefire.version>  </properties>  <dependencies>  <dependency>  <groupId>io.cucumber</groupId>  <artifactId>cucumber-java</artifactId>  <version>${cucumber.version}</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>io.cucumber</groupId>  <artifactId>cucumber-junit</artifactId>  <version>${cucumber.version}</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>${junit.version}</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>org.seleniumhq.selenium</groupId>  <artifactId>selenium-java</artifactId>  <version>3.141.59</version>  </dependency>  <dependency>  <groupId>io.github.bonigarcia</groupId>  <artifactId>webdrivermanager</artifactId>  <version>4.2.2</version>  </dependency>  <!--  https://mvnrepository.com/artifact/tech.grasshopper/extentreports-cucumber6-adapter -->  <dependency>  <groupId>tech.grasshopper</groupId>  <artifactId>extentreports-cucumber6-adapter</artifactId>  <version>2.17.0</version>  </dependency>  <dependency>  <groupId>org.testng</groupId>  <artifactId>testng</artifactId>  <version>6.14.3</version>  <scope>test</scope>  </dependency>  <dependency>  <groupId>io.cucumber</groupId>  <artifactId>cucumber-testng</artifactId>  <version>${cucumber.version}</version>  <scope>test</scope>  </dependency>  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->  <dependency>  <groupId>org.apache.poi</groupId>  <artifactId>poi</artifactId>  <version>4.1.2</version>  </dependency>  <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->  <dependency>  <groupId>org.apache.poi</groupId>  <artifactId>poi-ooxml</artifactId>  <version>4.1.2</version>  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>${maven.compiler.version}</version>  <configuration>  <encoding>UTF-8</encoding>  <source>${java.version}</source>  <target>${java.version}</target>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-surefire-plugin</artifactId>  <version>${maven.surefire.version}</version>  <configuration>  <!-- to run with jenkins and to run using maven use the below includes -->  <includes>  <include>\*\*/ParalelRunner.java</include>  </includes>  <parallel>methods</parallel>  <!--<threadCount>4</threadCount>-->  <useUnlimitedThreads>true</useUnlimitedThreads>  </configuration>  </plugin>  <!-- <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-failsafe-plugin</artifactId>  <version>3.1.2</version>  <executions>  <execution>  <goals>  <goal>integration-test</goal>  </goals>  <configuration> -->  <!-- UNCOMMENT - To add any exclusions if required -->  <!-- <excludes> <exclude>\*\*/\*IT\*.java</exclude>  </excludes> -->  <!-- <includes> -->  <!--UNCOMMENT BELOW LINE - To execute feature files with a single runner -->  <!--give path of runner in the below step -->  <!-- <include>\*\*/MyTestRunner.java</include> -->  <!-- UNCOMMENT BELOW LINE - To execute feature  files with multiple runners -->  <!-- <include>\*\*/\*Runner.java</include> -->  <!--</includes> -->  <!-- UNCOMMENT BELOW 3 LINES - To execute using  parallel or combination  option -->  <!-- we can write parallel with classes also.-->  <!-- <parallel>methods</parallel>  <threadCount>10</threadCount>  <perCoreThreadCount>true</perCoreThreadCount> -->  <!-- UNCOMMENT BELOW 3 LINES - To execute using  forking or combination option -->  <!-- <forkCount>2</forkCount>  <reuseForks>true</reuseForks>  <reportsDirectory>${project.build.directory}/failsafe-reports\_${surefire.forkNumber}</reportsDirectory>-->  <!--</configuration>  </execution>  </executions>  </plugin>-->  </plugins>  </build>  </project> |