http://toolsqa.com/selenium-cucumber-framework/run-cucumber-test-from-command-line-terminal/

For successful implementation of any testing framework, it is mandatory that test cases can be run in multiple ways so that people with different competency levels can use it how they need to. Because running any test framework from the Terminal has its own advantages, such as overriding the run configurations mentioned in the code. So, now we will focus on various ways to run the Cucumber tests.

Run Cucumber Test from Command Line / Terminal

There are different ways to run Cucumber Test from command line. Tests can be run by using JUnit and Maven as well. But maven is the most suggested way and has extra benefits to it. This is why we started this Project as Maven project. And remember, Maven has a lot of advantages over other build tools, such as dependency management, lots of plugins and the convenience of running integration tests. Maven will allow our test cases to be run in different flavors, such as from the ***Terminal***, integrating with ***Jenkins***, and ***parallel******execution***.

***Maven Project***

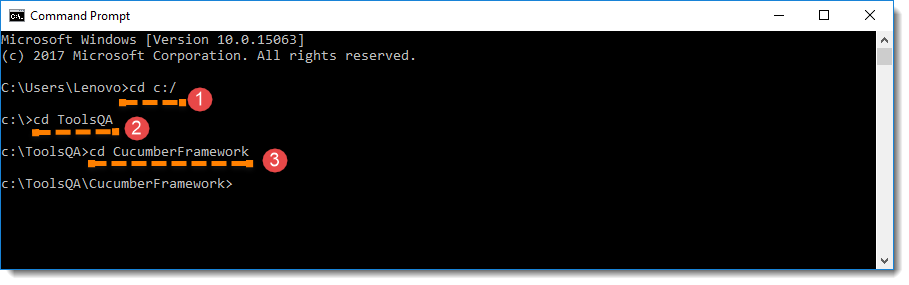
If in case you directly landed to this tutorial from search and not following this complete [***Selenium Cucumber Framework***](http://toolsqa.com/selenium-cucumber-framework) series, I suggest you to go through the first chapter of [***End 2 End Selenium Test***](http://toolsqa.com/selenium-cucumber-framework/selenium-end-to-end-automation-test/) . This chapter covers all the prerequisite for running test from Maven. Also to make sure that you have the below mentioned dependencies added in your Maven Project:

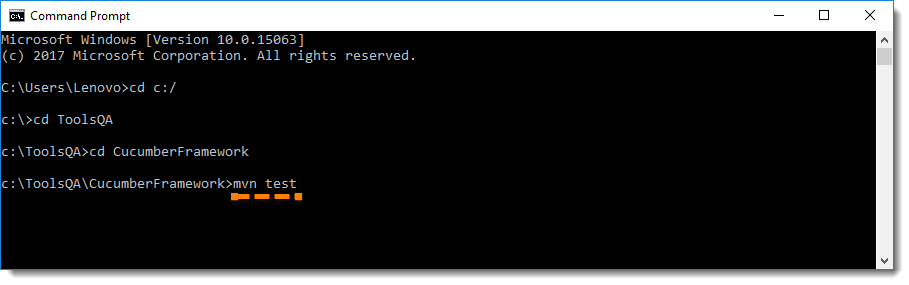
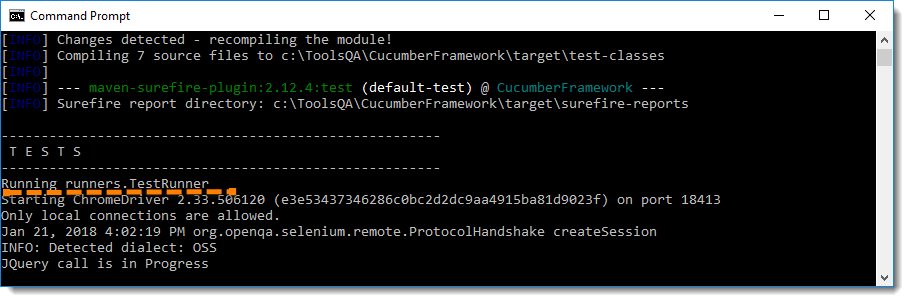


|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79 | <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>ToolsQA</groupId>    <artifactId>CucumberFramework</artifactId>    <version>0.0.1-SNAPSHOT</version>    <packaging>jar</packaging>      <name>CucumberFramework</name>    <url>http://maven.apache.org</url>      <properties>      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>    </properties>      <dependencies>  <dependency>      <groupId>junit</groupId>      <artifactId>junit</artifactId>      <version>4.12</version>      <scope>test</scope>  </dependency>       <dependency>      <groupId>org.seleniumhq.selenium</groupId>      <artifactId>selenium-java</artifactId>      <version>3.7.0</version>  </dependency>  <dependency>      <groupId>info.cukes</groupId>      <artifactId>cucumber-java</artifactId>      <version>1.2.5</version>  </dependency>  <dependency>      <groupId>info.cukes</groupId>      <artifactId>cucumber-jvm-deps</artifactId>      <version>1.0.5</version>      <scope>provided</scope>  </dependency>  <dependency>      <groupId>info.cukes</groupId>      <artifactId>cucumber-junit</artifactId>      <version>1.2.5</version>      <scope>test</scope>  </dependency>  <dependency>      <groupId>info.cukes</groupId>      <artifactId>cucumber-picocontainer</artifactId>      <version>1.2.5</version>      <scope>test</scope>  </dependency>  <dependency>      <groupId>com.aventstack</groupId>      <artifactId>extentreports</artifactId>      <version>3.1.2</version>  </dependency>  <dependency>      <groupId>com.vimalselvam</groupId>      <artifactId>cucumber-extentsreport</artifactId>      <version>3.0.2</version>  </dependency>    </dependencies>        <build>      <plugins>        <plugin>          <groupId>org.apache.maven.plugins</groupId>          <artifactId>maven-compiler-plugin</artifactId>          <version>3.7.0</version>          <configuration>            <source>1.8</source>            <target>1.8</target>             <encoding>UTF-8</encoding>          </configuration>        </plugin>         </plugins>     </build>    </project> |

***Run Test from Command Line***

1. Open the***command prompt*** and ***cd*** until the project root directory.



2. First, let’s run all the Cucumber Scenarios from the *command prompt*. Since it’s a Maven project and we have added Cucumber in ***test scope*** dependency and all features are also added in ***src/test*** packages, run the following command in the command prompt: ***mvn test***  
  
You would notice below that it actually triggered the ***TestRunner***file.  


***Build Success Output***

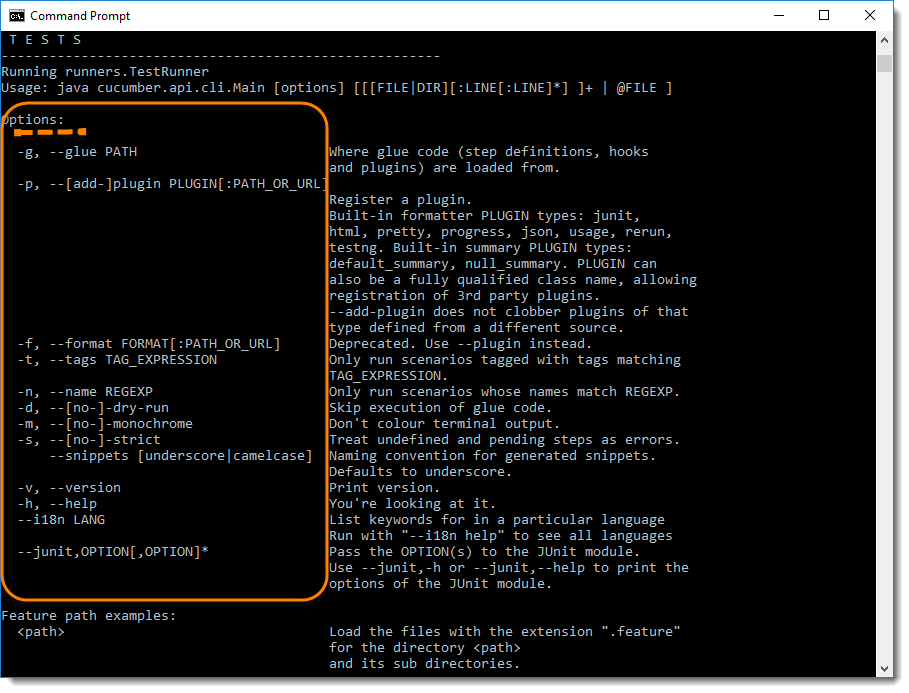


***Note****: mvn test runs Cucumber Features using Cucumber’s JUnit Runner. The @RunWith (Cucumber.class) annotation on the TestRunner class tells JUnit to kick off Cucumber. Cucumber run time parses the command-line options to know what Feature to run, where the Glue Code lives, what plugins to use, and so on.*

*On the other hand, if you run test from eclipse when you use the JUnit Runner, these options are generated from the @CucumberOptions annotation on your test.*

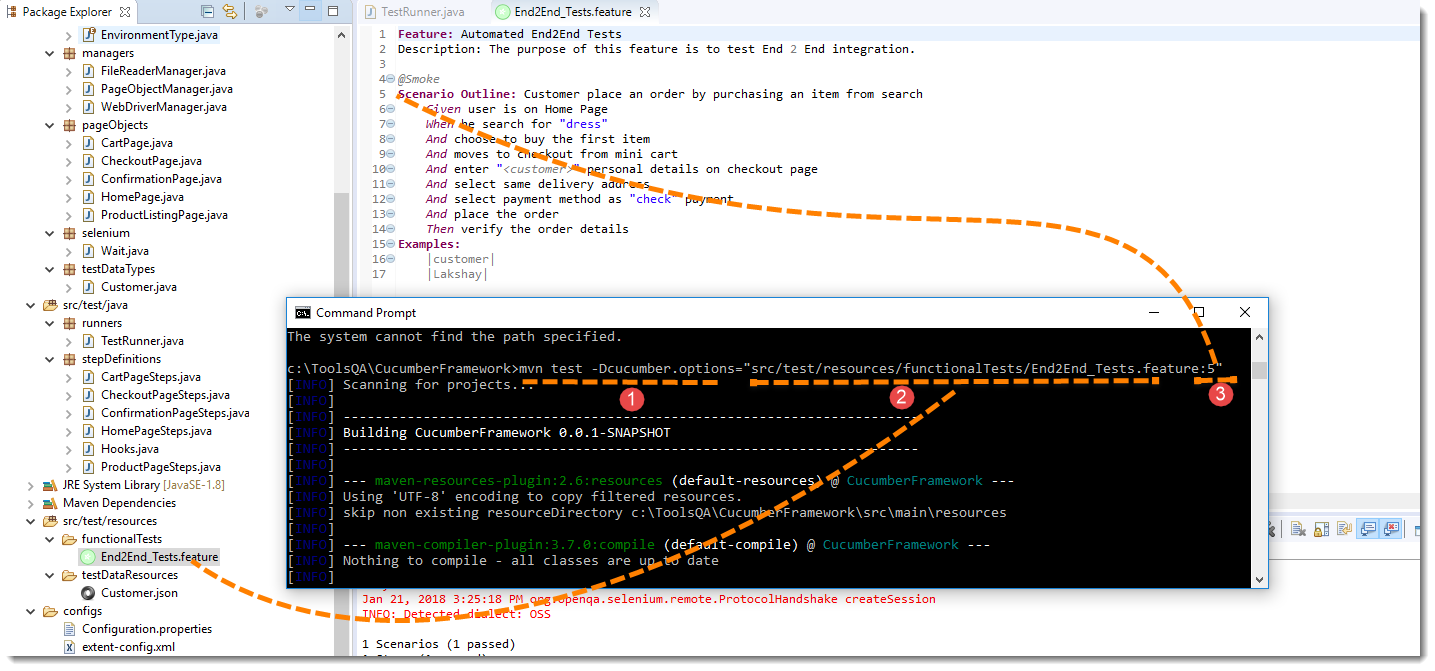
Overriding Cucumber Options

At times, you would need to override the configurations specified in *@CucumberOptions*. If we want to override the configurations mentioned in the ***Runner***, then we need to use following command: ***mvn test –DCucumber.options=”Your Options”***

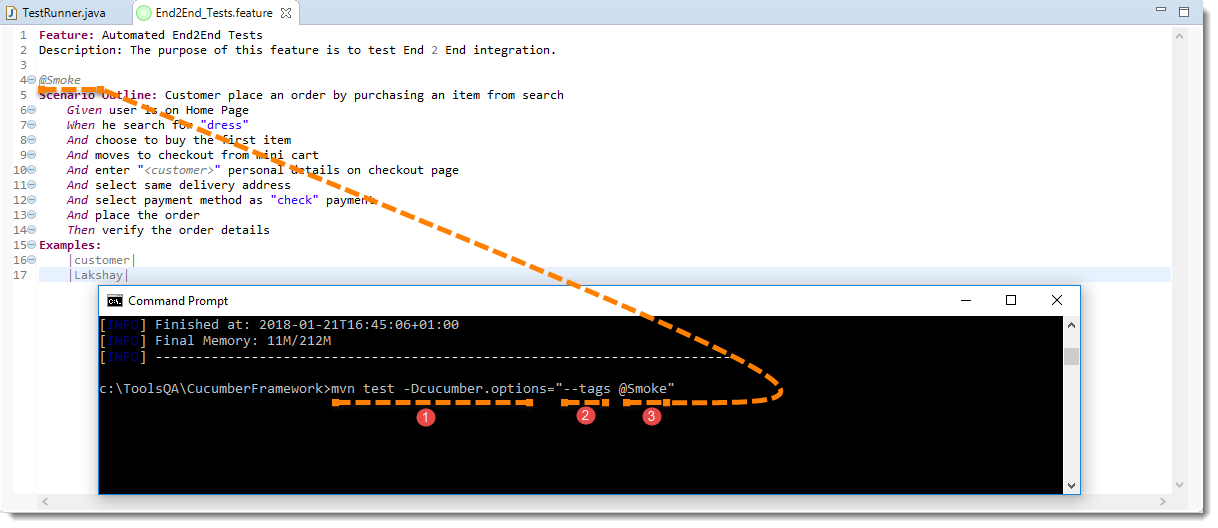
If you need help on these*Cucumber options*, then enter the following command in the command prompt and look at the output: ***mvn test -Dcucumber.options=”–help”***  


***Note :****This gives you the list of parameters that can be passed through command line using CucumberOptions.*

***Running a Scenario from Command Line***

If we want to run single ***Scenario*** from the ***cmd,****this is how we specify****:****mvn test -Dcucumber.options=”feature file path” + “line number of the scenario”*  
[](http://20tvni1sjxyh352kld2lslvc.wpengine.netdna-cdn.com/wp-content/gallery/cucumber/Running-Cucumber-Test-from-Command-Line-5.png)  
 ***Note****: In the preceding code,****“5”****is the Feature file line number where a Scenario starts.*

***Running Scenarios using Tags from Command Line***

If you want to run the test cases associated to ***Tags***, this is how we specify : *mvn test -Dcucumber.options=”–tags @tag Name”*  


***Overriding Report Plugin and Path***

If you want to generate a different report or if you wish to change the report path. This how you do it: *mvn test -Dcucumber.options=”–plugin junit:target/cucumber-reports/report.xml”.*

Please look at our [***Cucumber Report***](http://toolsqa.com/selenium-cucumber-framework/cucumber-reports/) to see what all reports you can generate using *@CucumberOptions*.

***Running a Feature file only from Command Line***

If you want cucumber to run just a single feature file or multiple feature file, you can pass parameter for the same from command line. This how you do it:

*mvn test -Dcucumber.options=”src/test/resources/functionalTests/End2End\_Tests.feature”*

***Passing multiple Parameter at once***

Its also possible to pass multiple options at once. This is how it can be done *mvn test -Dcucumber.options=”SomeThing” -Dcucumber.options=”SomeThing” -Dcucumber.options=”SomeThing”*

***mvn test -Dcucumber.options=”src/test/resources/functionalTests/End2End\_Tests.feature”-Dcucumber.options=”–tags @Smoke”***