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

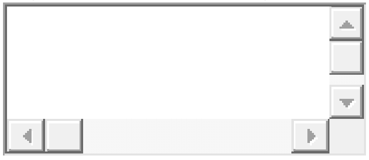
For successful implementation of any testing framework, it is mandatory that test cases can be run in multiple ways so that people with different competency levels can use it how they need to. Because running any test framework from the Terminal has its own advantages, such as overriding the run configurations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

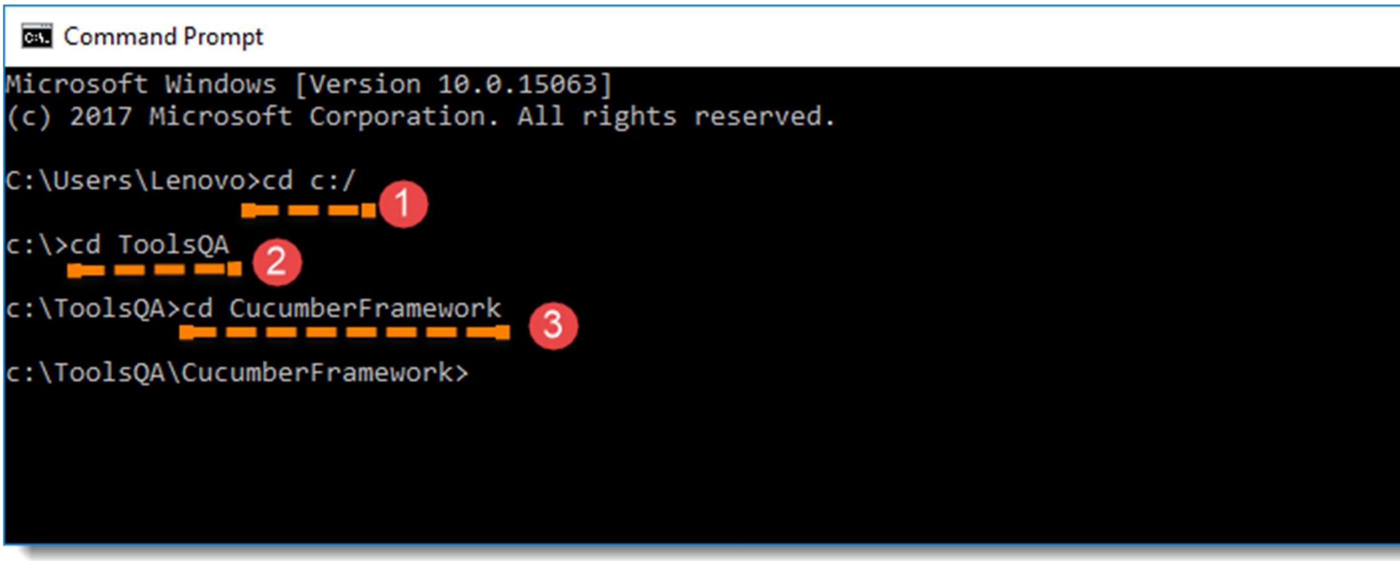
If in case you directly landed to this tutorial from search and not following this complete ***Selenium Cucumber Framework*** series, I suggest you to go through the first chapter of ***End 2 End Selenium Test*** . This chapter covers all the prerequisite for running test from Maven. Also to make sure that you have the below mentioned dependencies added in your Maven Project:



```
<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>
```

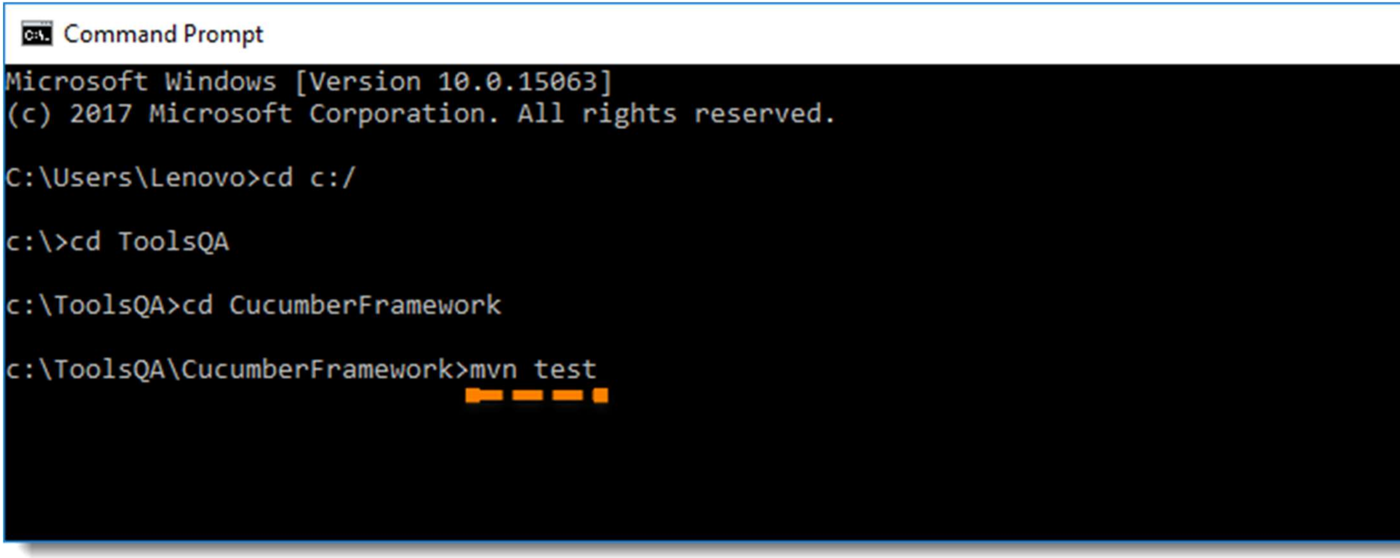
Run Test from Command Line

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



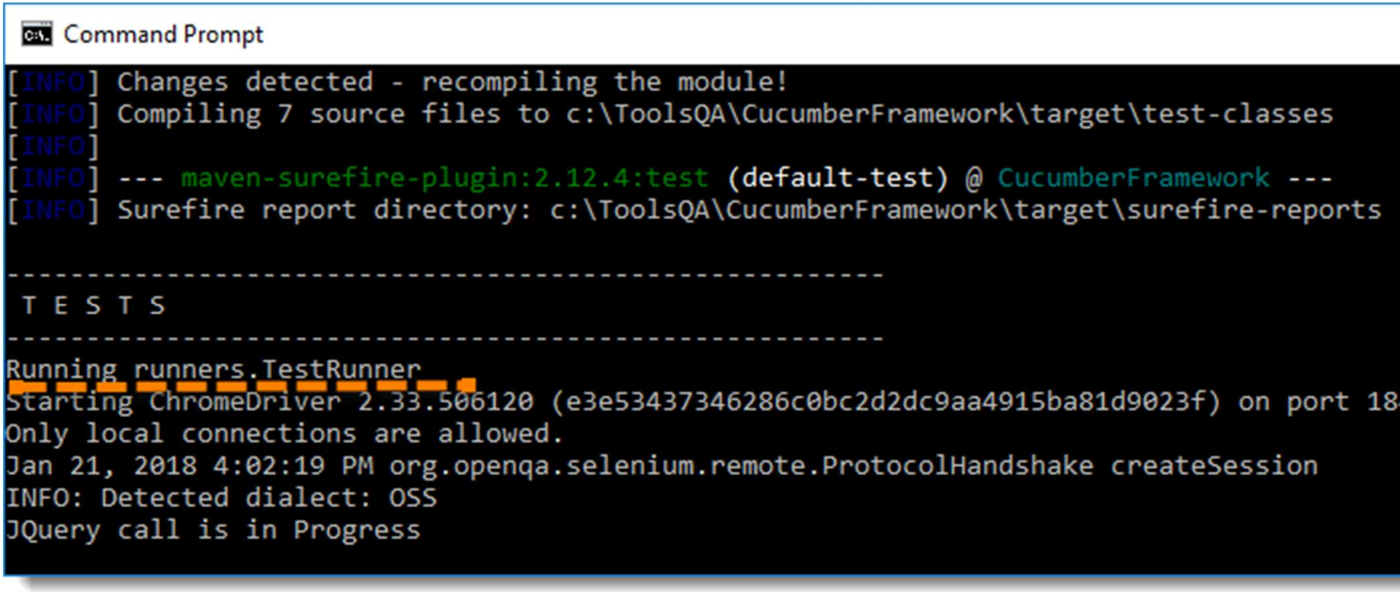
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>cd c:/

c:\>cd ToolsQA

c:\ToolsQA>cd CucumberFramework

c:\ToolsQA\CucumberFramework>
```

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***



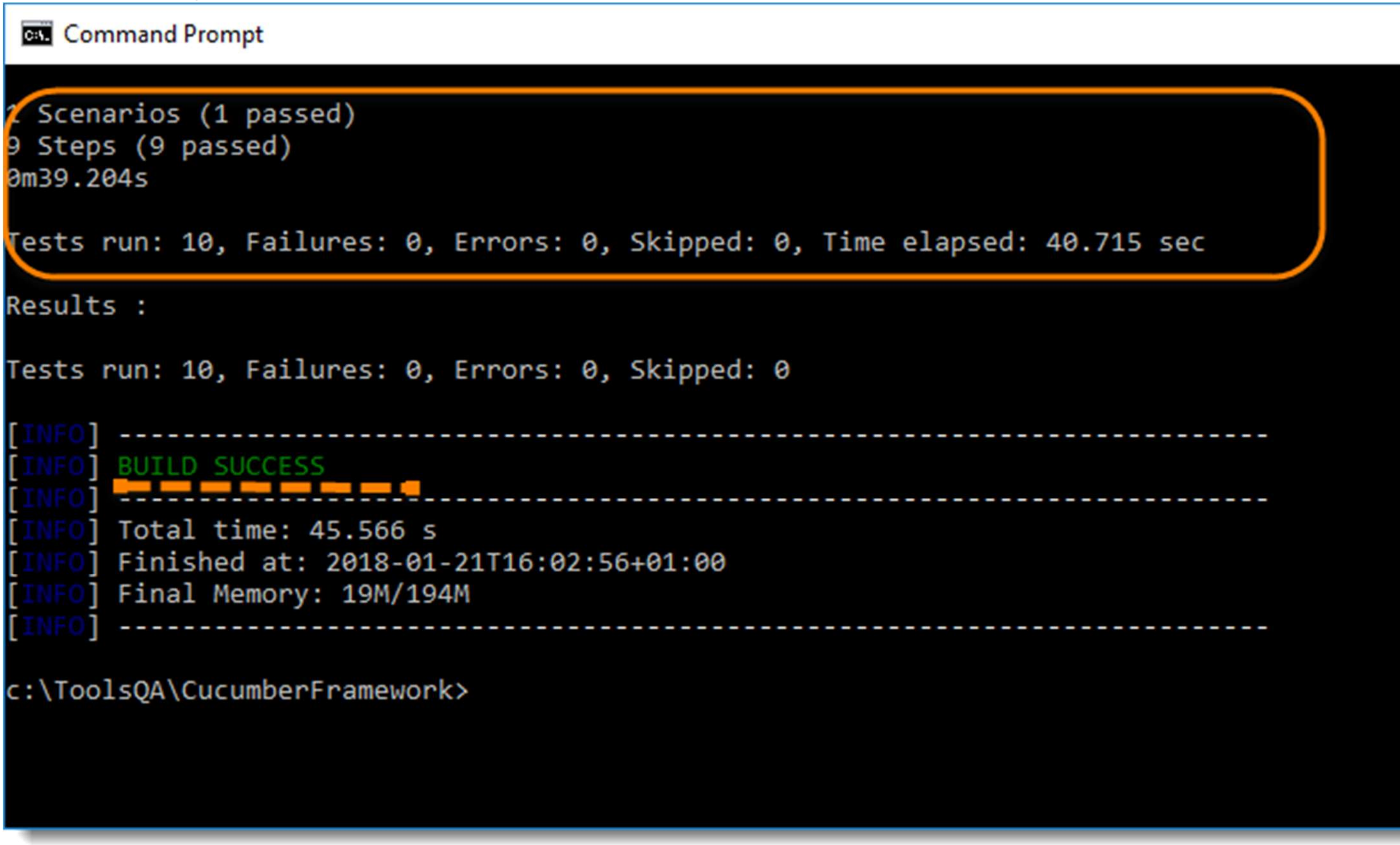
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>cd c:/

c:\>cd ToolsQA

c:\ToolsQA>cd CucumberFramework

c:\ToolsQA\CucumberFramework>mvn test
```

You would notice below that it actually triggered the ***TestRunner*** file.



```
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 7 source files to c:\ToolsQA\CucumberFramework\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ CucumberFramework ---
[INFO] Surefire report directory: c:\ToolsQA\CucumberFramework\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running runners.TestRunner
Starting ChromeDriver 2.33.506120 (e3e53437346286c0bc2d2dc9aa4915ba81d9023f) on port 18
Only local connections are allowed.
Jan 21, 2018 4:02:19 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
JQuery call is in Progress
```

Build Success Output



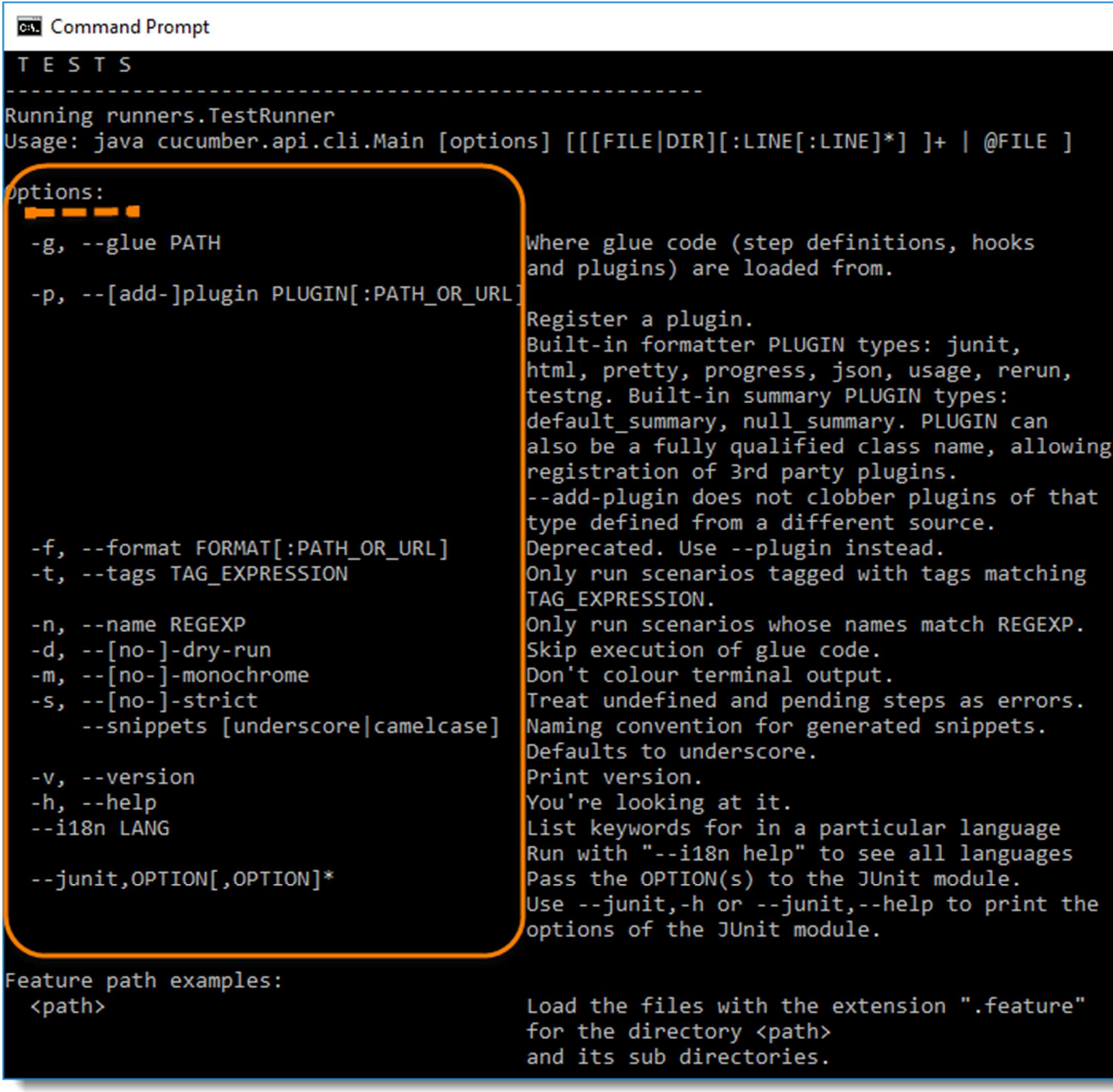
```
1 Scenarios (1 passed)
9 Steps (9 passed)
0m39.204s

Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 40.715 sec

Results :

Tests run: 10, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 45.566 s
[INFO] Finished at: 2018-01-21T16:02:56+01:00
[INFO] Final Memory: 19M/194M
[INFO] ------------------------------------------------------------------------

c:\ToolsQA\CucumberFramework>
```

***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”***

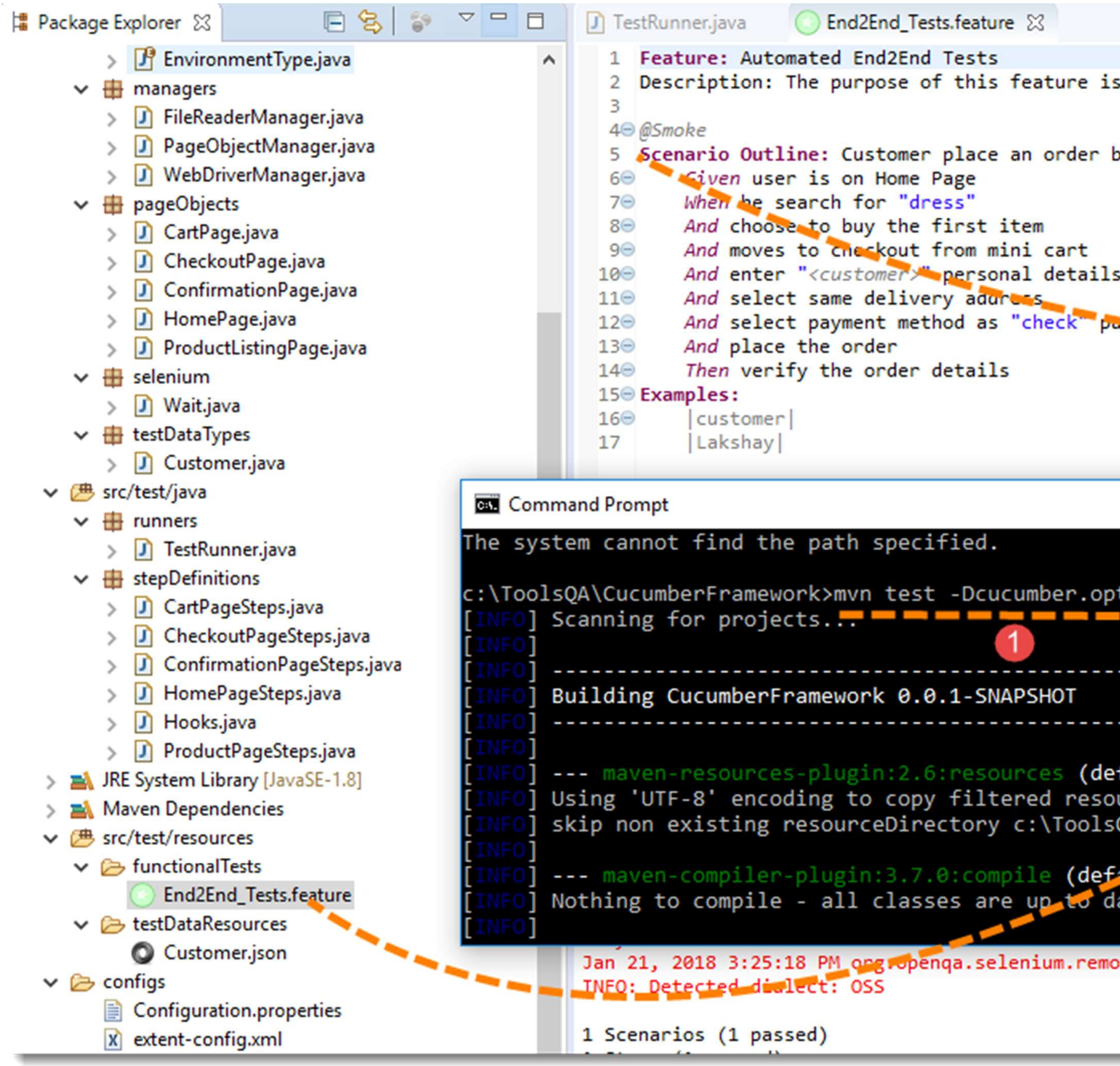


```
 T E S T S
-------------------------------------------------------
Running runners.TestRunner
Usage: java cucumber.api.cli.Main [options] [[[FILE|DIR][:LINE[:LINE]*] ]+ | @FILE ]

Options:

  -g, --glue PATH                        Where glue code (step definitions, hooks
                                         and plugins) are loaded from.
  -p, --[add-]plugin PLUGIN[:PATH_OR_URL]
                                         Register a plugin.
                                         Built-in formatter PLUGIN types: junit,
                                         html, pretty, progress, json, usage, rerun,
                                         testng. Built-in summary PLUGIN types:
                                         default_summary, null_summary. PLUGIN can
                                         also be a fully qualified class name, allowing
                                         registration of 3rd party plugins.
                                         --add-plugin does not clobber plugins of that
                                         type defined from a different source.
  -f, --format FORMAT[:PATH_OR_URL]      Deprecated. Use --plugin instead.
  -t, --tags TAG_EXPRESSION              Only run scenarios tagged with tags matching
                                         TAG_EXPRESSION.
  -n, --name REGEXP                      Only run scenarios whose names match REGEXP.
  -d, --[no-]-dry-run                    Skip execution of glue code.
  -m, --[no-]-monochrome                 Don't colour terminal output.
  -s, --[no-]-strict                     Treat undefined and pending steps as errors.
      --snippets [underscore|camelcase]  Naming convention for generated snippets.
                                         Defaults to underscore.
  -v, --version                          Print version.
  -h, --help                             You're looking at it.
  --i18n LANG                            List keywords for in a particular language
                                         Run with "--i18n help" to see all languages
  --junit,OPTION[,OPTION]*               Pass the OPTION(s) to the JUnit module.
                                         Use --junit,-h or --junit,--help to print the
                                         options of the JUnit module.

Feature path examples:
  <path>                                 Load the files with the extension ".feature"
                                         for the directory <path>
                                         and its sub directories.
```

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”*



```
1  Feature: Automated End2End Tests
2  Description: The purpose of this feature is
3
4  @Smoke
5  Scenario Outline: Customer place an order b
6      Given user is on Home Page
7      When he search for "dress"
8      And choose to buy the first item
9      And moves to checkout from mini cart
10     And enter "<customer>" personal details
11     And select same delivery address
12     And select payment method as "check" pa
13     And place the order
14     Then verify the order details
15 Examples:
16     |customer|
17     |Lakshay|
```

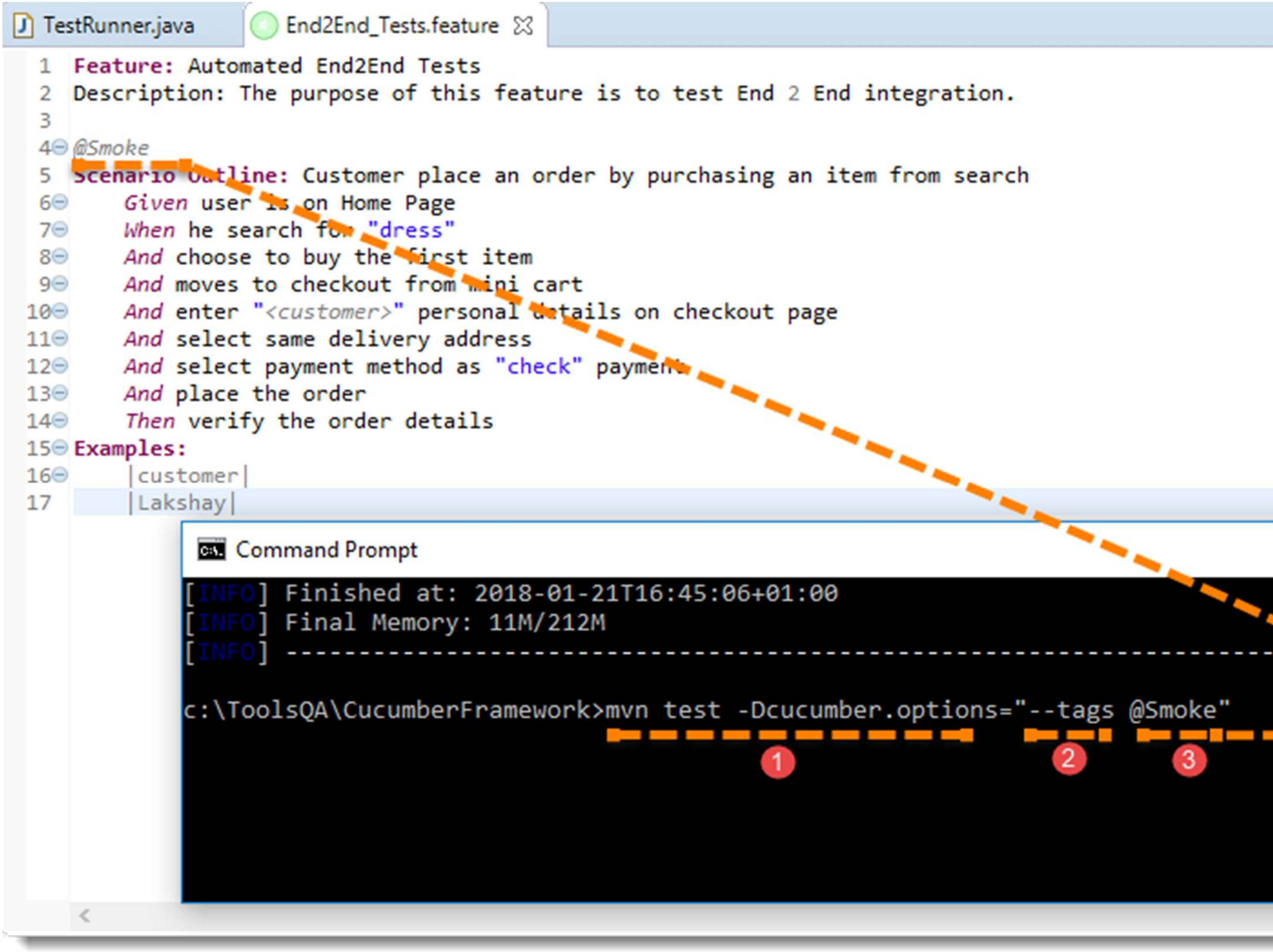
```
The system cannot find the path specified.

c:\ToolsQA\CucumberFramework>mvn test -Dcucumber.opt
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------
[INFO] Building CucumberFramework 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (de
[INFO] Using 'UTF-8' encoding to copy filtered reso
[INFO] skip non existing resourceDirectory c:\Tools
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (def
[INFO] Nothing to compile - all classes are up to da
[INFO]
```

```
Jan 21, 2018 3:25:18 PM org.openqa.selenium.remo
INFO: Detected dialect: OSS

1 Scenarios (1 passed)
```

***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”*



```
Feature: Automated End2End Tests
Description: The purpose of this feature is to test End 2 End integration.

@Smoke
Scenario Outline: Customer place an order by purchasing an item from search
    Given user is on Home Page
    When he search for "dress"
    And choose to buy the first item
    And moves to checkout from mini cart
    And enter "<customer>" personal details on checkout page
    And select same delivery address
    And select payment method as "check" payment
    And place the order
    Then verify the order details
Examples:
    |customer|
    |Lakshay|
```

```
[INFO] Finished at: 2018-01-21T16:45:06+01:00
[INFO] Final Memory: 11M/212M
[INFO] ------------------------------------------------------------

c:\ToolsQA\CucumberFramework>mvn test -Dcucumber.options="--tags @Smoke"
```

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*** 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”