**Running Cucumber:**

Cucumber is a Junit extension. It is launched by running by Junit from the build tool or IDE.

**From the command line:**

By default, cucumber will treat anything ending in .java under root resources directory as a step definition file.

>**java io.cucumber.core.cli.Main**

We can also run features using build tool or IDE.

To use Junit to execute cucumber scenarios add the Cucumbr-Junit dependency in pom.

@RunWith(Cucumber.class)

This will execute all scenarios in same package as the runner, by default, glue code is also assumed to in the same package.

@CucumberOptions can be use to provide additional configuration to the runner.

@@@@

Note : Conditional hooks:

Hooks can be conditionally selected for execution based on the tags of the scenario. To run a particular hook only for certain scenarios, we can associate Before or After hook with tag expression.

@Before(@browser and not @headless)

**Tags:**

Tags are a great way to organize the features and scenarios,

They can be used for two purposes:

1. Running a subset of scenarios
2. Restricting hooks to subset of scenarios

**Building executable jar with Cucumber tests and maven configuration:**

* We use **maven-assembly-plugin** to configure in pom.xml, when we use command mvn clean package the jar is automatically created.
* The plugin uses descriptors to define what to put inside of the jar file.( there is predefined descriptors and customer descriptors xmls

<plugin >

<artifactId>maven-assembly-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

</configuration>

</plugin>

* To configure in the plugin to executable class main method. It also goes into plugin configuration. Jar file to be executable needs to have a main call referenced in MANIFEST file. it is automatically done for us by maven-assembly-plugin, when appropriately configured in pom.

<configuration>

...

<archive>

<manifest>

<mainClass>

runners.TestRunner

</mainClass>

</manifest>

</archive>

</configuration>

* Executable cucumber test runner class, to run cucumber we require test runner class with appropriate CucumberOptions configured using Junit. Here we are running assembled jar file on command line with executable class inside, therefore we need Runner class with main method to be properly recognized by java as executable class.

public final class TestRunner {

public static void main(String[] args) {

cucumber.api.cli.Main.main(args );

}

}

* To execute the jar file

java -jar cucumber-test-jar-with-dependencies.jar --glue cucumber/stepdefinition --plugin pretty classpath:features/MyTestFeature.feature