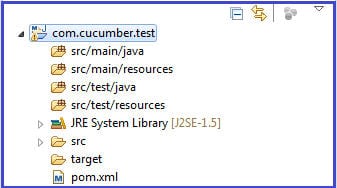
**Cucumber Selenium WebDriver Java Integration with Example:**

We will set up a Cucumber project with Maven

**Cucumber Project Setup**

**Step #1:** Create a New Maven Project:  
*Right Click -> New -> Others -> Maven -> Maven Project -> Next*

**Step #2:** Now the project will look like this:

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Cucumber-Selenium-1.jpg)

**Step #3**: Add below dependencies in pom.xml

**<dependencies>**

**<!-- CUCUMBER DEPENDENCIES -->**

**<dependency>**

**<groupId>info.cukes</groupId>**

**<artifactId>cucumber-java</artifactId>**

**<version>1.0.2</version>**

**<scope>test</scope>**

**</dependency>**

**<dependency>**

**<groupId>info.cukes</groupId>**

**<artifactId>cucumber-junit</artifactId>**

**<version>1.0.2</version>**

**<scope>test</scope>**

**</dependency>**

**<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->**

**<dependency>**

**<groupId>org.seleniumhq.selenium</groupId>**

**<artifactId>selenium-java</artifactId>**

**<version>3.4.0</version>**

**</dependency>**

**<dependency>**

**<groupId>org.seleniumhq.selenium</groupId>**

**<artifactId>selenium-server</artifactId>**

**<version>3.4.0</version>**

**</dependency>**

**<!-- https://mvnrepository.com/artifact/junit/junit -->**

**<dependency>**

**<groupId>junit</groupId>**

**<artifactId>junit</artifactId>**

**<version>4.12</version>**

**<scope>test</scope>**

**</dependency>**

**</dependencies>**

**Step #4**: Create a sample.feature file under src/test/resources.

@smokeTest  
**Feature**: To test my cucumber test is running  
I want to run a sample feature file.

**Scenario**: cucumber setup

**Given** sample feature file is ready  
**When** I run the feature file  
**Then** run should be successful

**Note: SMOKE TESTING**, also known as “Build Verification Testing”, is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work.

A smoke test is a quick run through of a site;  it focuses on critical functionality to ensure the site can perform basic features.   
‘Smoke testing‘ came to software testing from a similar hardware test -where the device passed if it did not catch fire (or smoked) the first time it was turned on!

For software purposes, an example of smoke testing could be for a hotel reservation site.

**Step #5**: Create a class under src/test/java which will implement all the steps.

**package com.smita.cucumber;**

**import cucumber.annotation.en.Given;**

**import cucumber.annotation.en.Then;**

**import cucumber.annotation.en.When;**

**/\*\*\* @author Smita\* \*/**

**public class StepDefinition {**

**@Given("^sample feature file is ready$")**

**public void givenStatment(){**

**System.out.println("Given statement executed successfully");**

**}**

**@When("^I run the feature file$")**

**public void whenStatement(){**

**System.out.println("When statement execueted successfully");**

**}**

**@Then("^run should be successful$")**

**public void thenStatment(){**

**System.out.println("Then statement executed successfully");**

**}**

**}**

**Step #6**: Create a JUnit runner to run the test.

@RunWith(Cucumber.class)

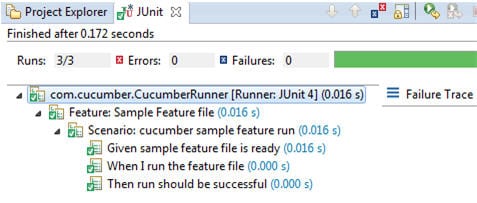
@Cucumber.Options(format={"pretty","html:reports/test-report"},tags= "@smokeTest")

public class CucumberRunner {

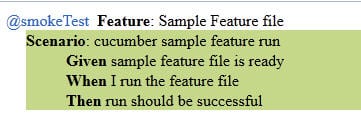
}

**Step #7**: Junit Result and Test Report:

Below is the report when the cucumber test is successful. The green bar in Junit describes the test is passed. Similarly, red bar describes that test has failed.

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Cucumber-Selenium-2.jpg)

If we want to use default reporting then navigate the path mentioned in Junit Runner. In this case, we have given path as reports->test-reports->index.html.

Open this report in Internet Explorer or in Firefox to verify the result. Below is the sample of the report:[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Cucumber-Selenium-3.jpg)