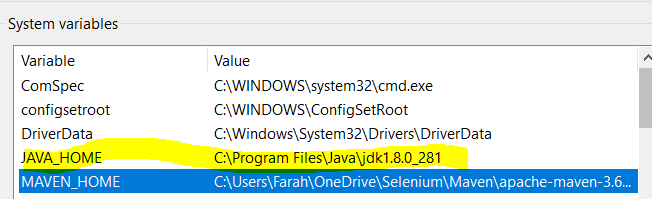
**Readme File – PetStore\_API\_Framework**

Java Setup

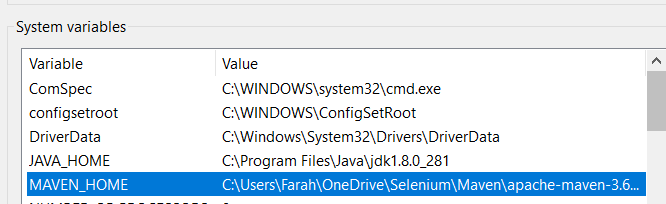
* Install Java on Computer
* Set environment variables



* Run java -version in cmd to check java installation

Maven Setup

* Download maven from given path : <https://maven.apache.org/download.cgi>
* Set Environment variables as shown below :



* Run mvn -version to check maven installation.

Generating Serenity project using maven archetypes:

* Open cmd prompt and run mvn archetype:generate -DarchetypeGroupId=net.serenity-bdd -DarchetypeArtifactId=serenity-cucumber-archetype
* Project is downloaded in our local system

Add Below dependency in pom.xml file:

<dependency>

<groupId>net.serenity-bdd</groupId>

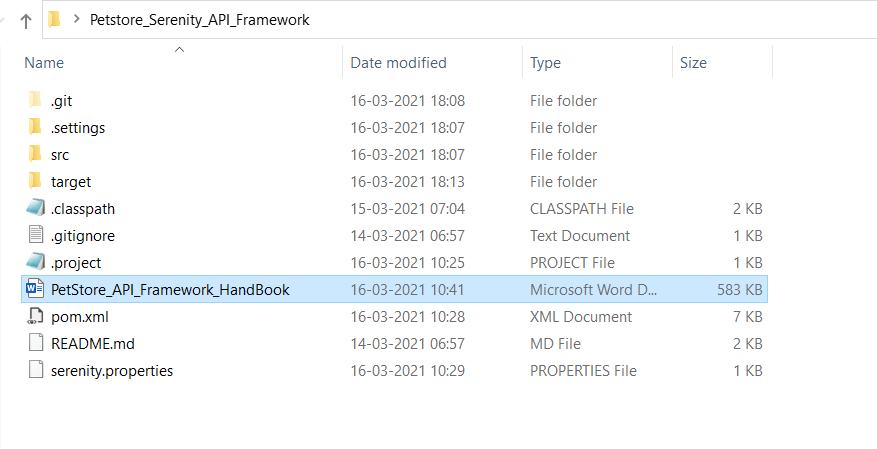
<artifactId>serenity-rest-assured</artifactId>

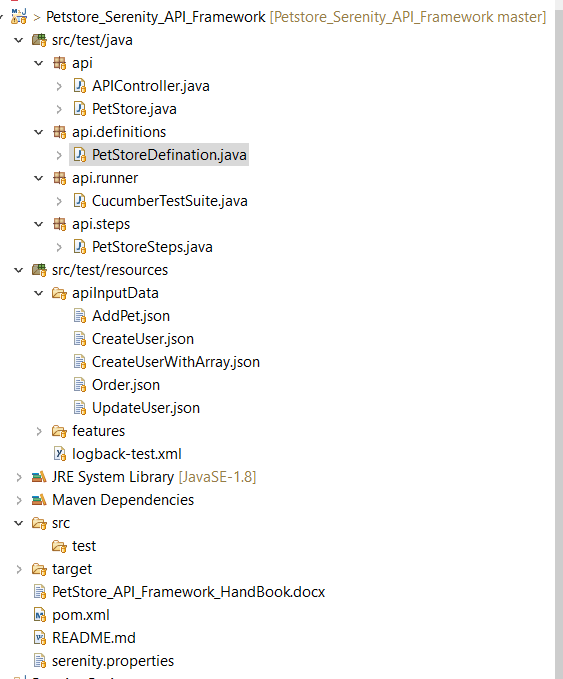
<version> 2.0.49 </version>

<scope>test</scope>

</dependency>

Serenity Framework Folder Structure

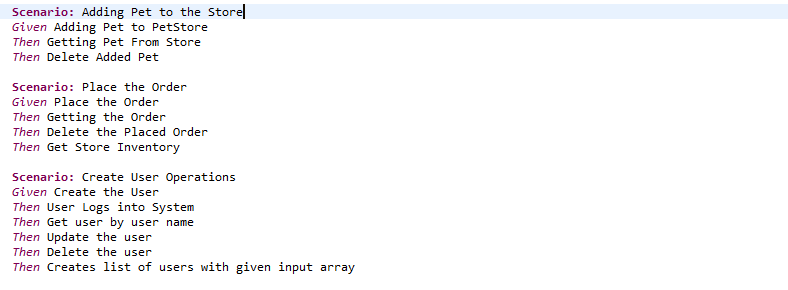




Cucumber Feature File:

**Feature File**

Test Scenarios are created in Feature File which contain an overall description of a feature as well as a number of scenarios. Feature files can be placed in different locations, but you can reduce the amount of configuration you need to do with serenity if you put them in the **src/test/resources/features**directory.



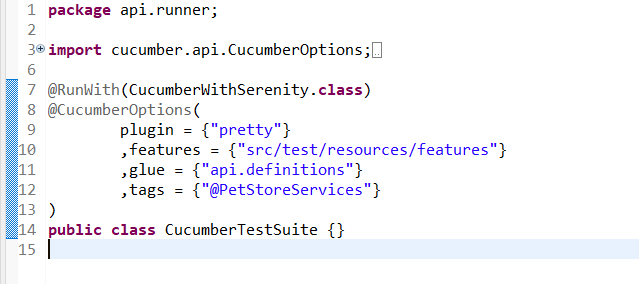
**Step Definition**

 It is a Java method with an expression which is used to link it to Gherkin steps. When Cucumber executes a Gherkin step, it will look for a matching step definition to execute. These use annotations like @given, @when and @then match lines in the scenario to java methods.



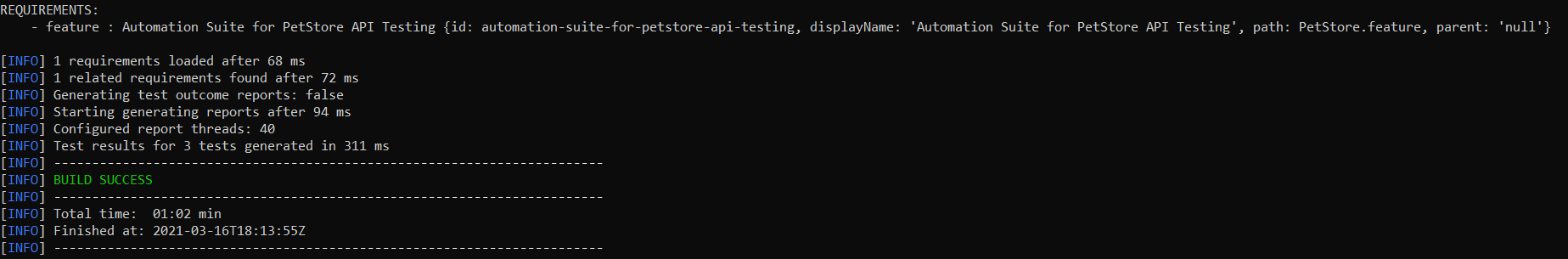
**Test Runner in Serenity**

Cucumber runs the feature files via JUnit and needs a dedicated Test Runner class to run the feature files. When you run the tests with serenity, you use the **CucumberWithSerenity**test runner. If the feature files are not in the same package as the test runner class, you also need to use the **@CucumberOptions**class to provide the root directory where the feature files can be found. It is the starting point for JUnit to start executing the tests. TestRunner class is created under src/ test/java. The test runner to run all of the feature files looks like this:



**Executing the tests**

* Open cmd prompt pointing to serenity framework path
* Run **mvn clean verify serenity:aggregate**



**Generating the Serenity reports:**

The test results will be recorded in the **target/site/serenity** directory

