

Tutorial Web UI Automation serenity-junit-archetype Selenium 2 + JUnit

Table of Contents

1. System requirements and environment setup: (See EVOZONdocumentation).....	1
2. Create Maven Project (serenity-junit-archetype, i.e. Selenium 2 + JUnit).....	2
2.1. Generate the project	2
2.2. Viewing options for archetype serenity	3
2.3. Selecting archetype for serenity and JUnit.....	3
2.4. Establish property values for the Maven project	3
3. Import the project in Eclipse	4
4. Run as JUnit test	6
5. Obtaining the documentation for the executed test cases	7
6. Viewing the Serenity report	8
7. Test Data Driven - data for test from csv file	9

1. System requirements and environment setup: (See EVOZONdocumentation)

Download and install:

Java JDK	http://www.oracle.com/technetwork/java/javase/downloads/index.html (JDK)
Maven	https://maven.apache.org/download.cgi (Binary zip)
Firefox (11)	http://www.filehippo.com/download_firefox/11868/
Chrome (22)	http://www.oldapps.com/google_chrome.php
Eclipse	https://eclipse.org/downloads/ (for Java EE Developers)

Note: in some cases the latest version of the browser might not work with the version of the webdriver you have.

a. System setup:

- Download and install Java JDK
- Download and place Maven on your c:\ drive
- Download and install desired web browsers

b. Windows 7 Setup:

Right Click on My Computer - Properties

Click on Advanced System Settings

Click on Environment Variables

In the System Variables section:

Add new variables

M2_HOME	C:\maven
M2	%M2_HOME%\bin

Add new variables

JAVA_HOME	C:\Program Files\Java\jdk1.7.0_04
JAVA	%JAVA_HOME%\bin

In the Path variable add at the end: ;%M2%;%JAVA%

c. To validate the installation is correct:

Click on start (Windows key) and type **cmd**

Type **mvn -version** (maven version should be displayed)

Type **java -version** (java version should be displayed)

2. Create Maven Project (serenity-junit-archetype, i.e. Selenium 2 + JUnit)

2.1. Generate the project

- click **Start** and open a Command prompt window with **cmd**
- use the command: **mvn archetype:generate**

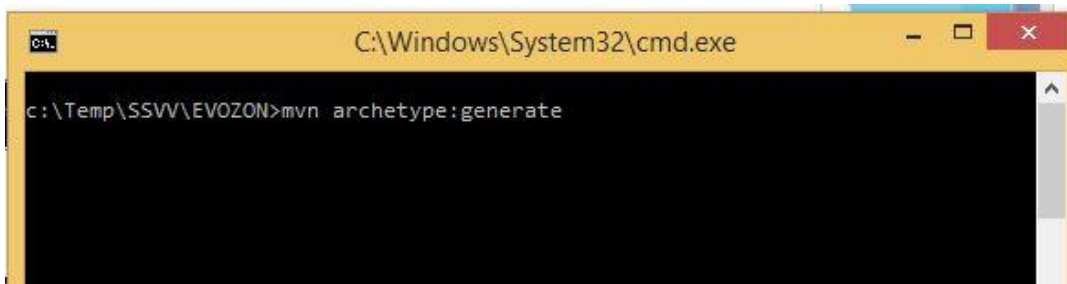


Figure 1a. Generate the Maven project (command)

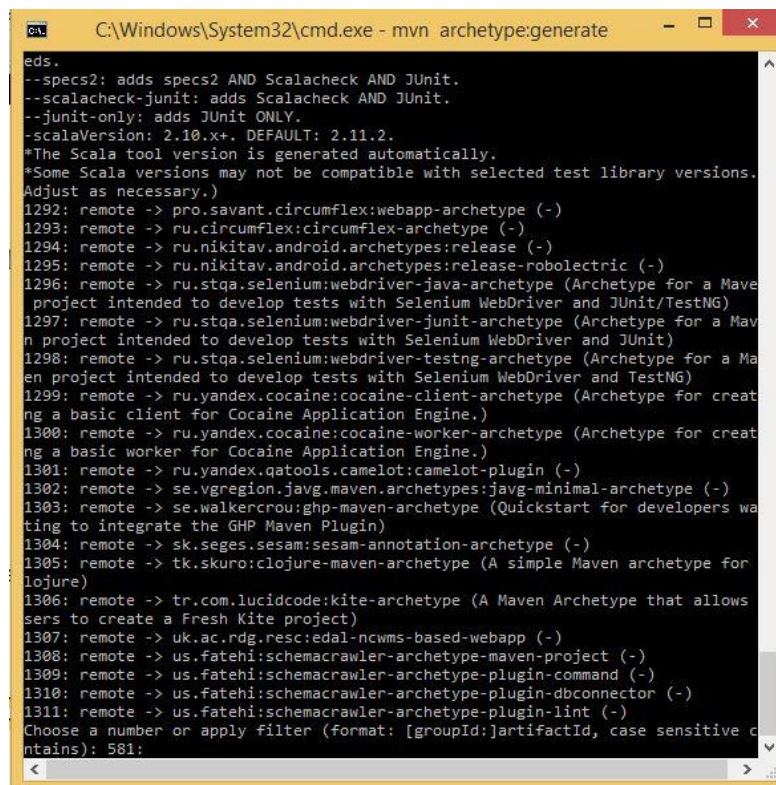


Figure 1b. Generate the Maven project (command execution)

2.2. Viewing options for archetype serenity

- Display the options for the Serenity archetype
- Enter option ...> serenity

```
1309: remote -> us.fatehi:schemacrawler-archetype-plugin-command (-)
1310: remote -> us.fatehi:schemacrawler-archetype-plugin-dbconnector (-)
1311: remote -> us.fatehi:schemacrawler-archetype-plugin-lint (-)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 581: serenity
Choose archetype:
1: remote -> net.serenity-bdd:serenity-cucumber-archetype (Serenity automated acceptance testing project using Selenium 2, JUnit and Cucumber-JVM)
2: remote -> net.serenity-bdd:serenity-jbehave-archetype (Serenity automated acceptance testing project using Selenium 2, JUnit and JBehave)
3: remote -> net.serenity-bdd:serenity-junit-archetype (Serenity automated acceptance testing project using Selenium 2 and JUnit)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): :
```

Figure 2. Filter serenity for archetype

2.3 Selecting archetype for serenity and JUnit

- Option 3 is selected (serenity-junit-archetype)>3

```
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 581: serenity
Choose archetype:
1: remote -> net.serenity-bdd:serenity-cucumber-archetype (Serenity automated acceptance testing project using Selenium 2, JUnit and Cucumber-JVM)
2: remote -> net.serenity-bdd:serenity-jbehave-archetype (Serenity automated acceptance testing project using Selenium 2, JUnit and JBehave)
3: remote -> net.serenity-bdd:serenity-junit-archetype (Serenity automated acceptance testing project using Selenium 2 and JUnit)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): : 3
Choose net.serenity-bdd:serenity-junit-archetype version:
1: 1.0.2
2: 1.0.6
3: 1.0.7
Choose a number: 3:
```

Figure 3. Choose option for serenity and junit

2.4. Establish property values for the Maven project

- groupId: SerenityWithJUnit
- artifactId: SerenityWithJUnitPrj
- version: <enter>
- package : <enter>
- at the end the properties of the project are confirmed with command “y”

```
Choose net.serenity-bdd:serenity-junit-archetype version:
1: 1.0.2
2: 1.0.6
3: 1.0.7
Choose a number: 3: 3
Define value for property 'groupId': : SerenityWithJUnit
Define value for property 'artifactId': : SerenityWithJUnitPrj
Define value for property 'version': 1.0-SNAPSHOT: :
Define value for property 'package': SerenityWithJUnit: :
Confirm properties configuration:
groupId: SerenityWithJUnit
artifactId: SerenityWithJUnitPrj
version: 1.0-SNAPSHOT
package: SerenityWithJUnit
Y: : y
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: serenity-
junit-archetype:1.0.7
[INFO] -----
[INFO] Parameter: groupId, Value: SerenityWithJUnit
[INFO] Parameter: artifactId, Value: SerenityWithJUnitPrj
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: SerenityWithJUnit
[INFO] Parameter: packageInPathFormat, Value: SerenityWithJUnit
[INFO] Parameter: package, Value: SerenityWithJUnit
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: SerenityWithJUnit
[INFO] Parameter: artifactId, Value: SerenityWithJUnitPrj
[INFO] project created from Archetype in dir: c:\Temp\SSVV\EVOZON\SerenityWithJU
nitPrj
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 06:00 min
[INFO] Finished at: 2015-04-06T15:21:50+03:00
[INFO] Final Memory: 15M/162M
[INFO] -----
c:\Temp\SSVV\EVOZON>
```

Figure 5. Establish properties for the Maven project

3. Import the project in Eclipse

- Open Eclipse
- Menu-> File->Import->Maven->Existing Maven Projects

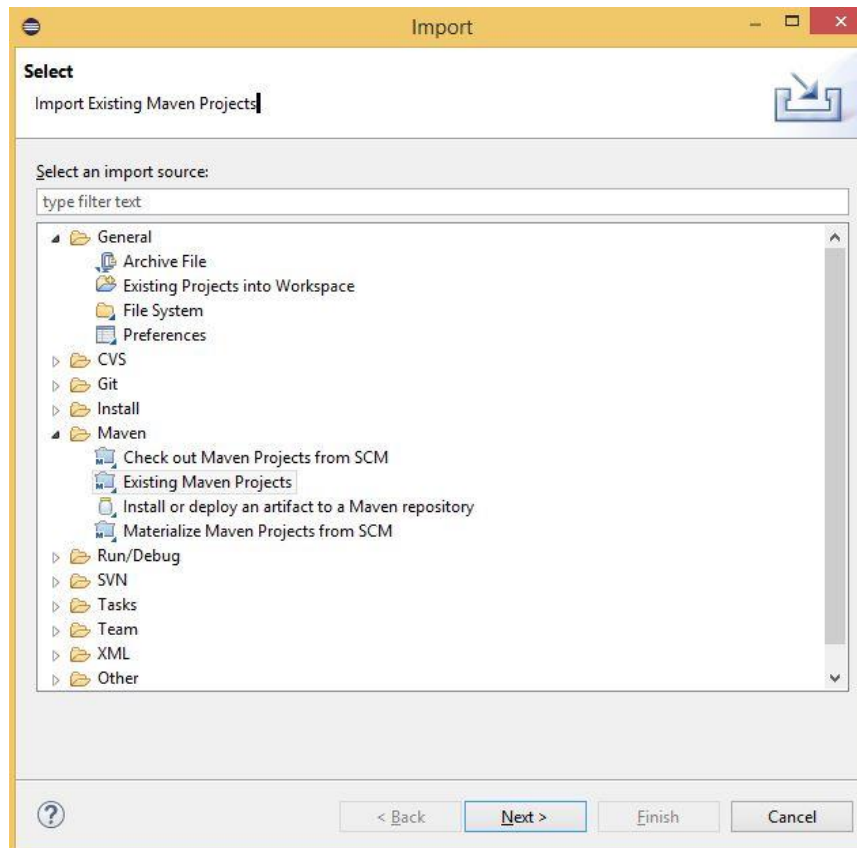


Figure 6 Maven project import in Eclipse

- Browse the files and select the created project and click Finish

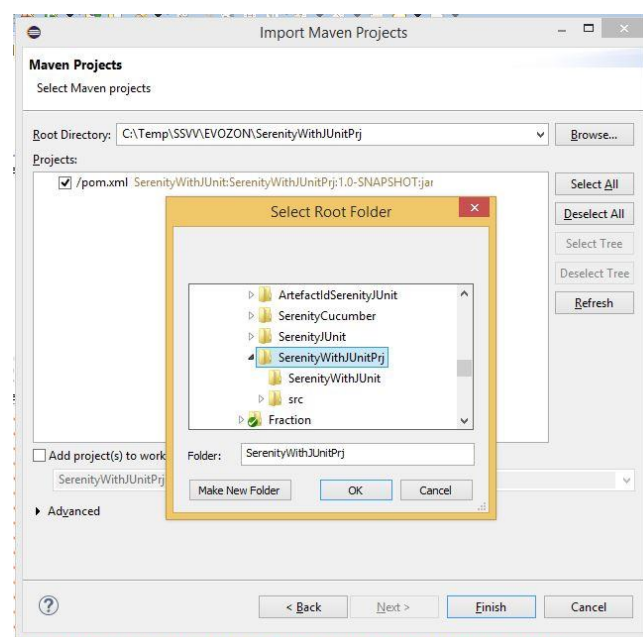


Figure 7 Select the Maven project created in command line

- The result is the project with the following package explorer structure

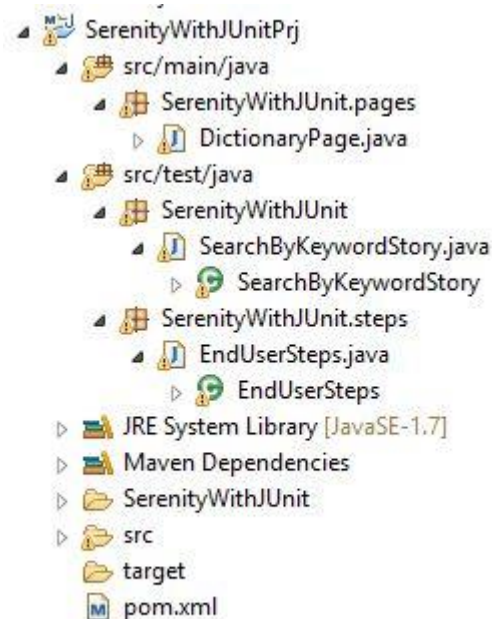


Figure 8 Structure of the Maven project in Eclipse

4. Run as JUnit test

- In Project Explorer-
- Right-click on the project (or just a TestCase) and select Run As-> JUnit test
- The Firefox browser is opened and the definitions for "pear" and "apple" are searched.

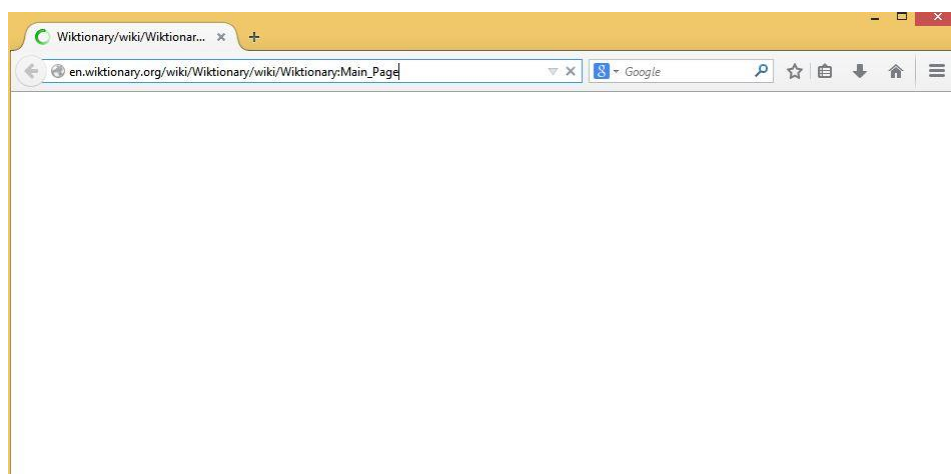


Figure 9 Execution of a test case in browser Firefox

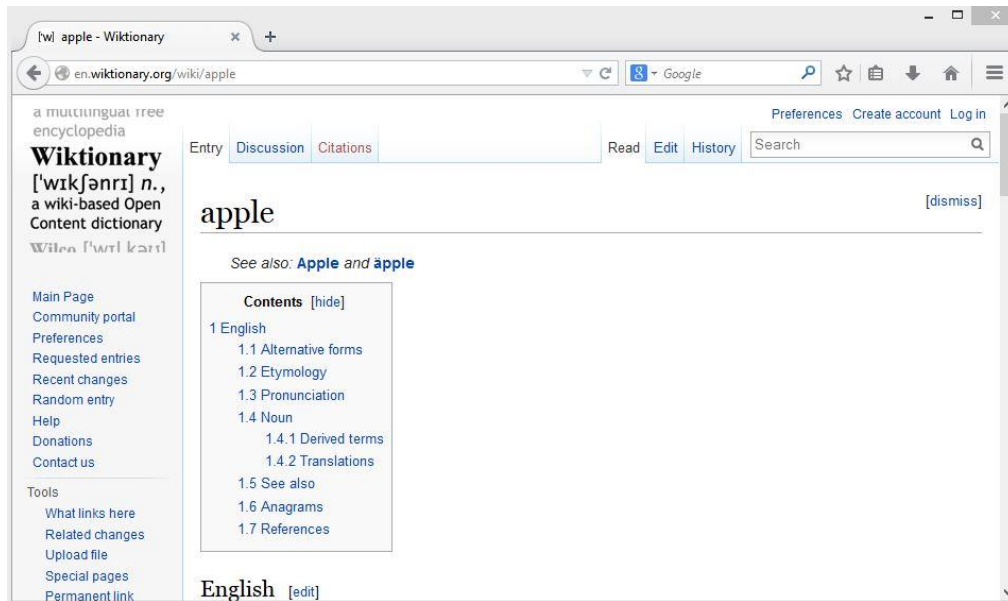


Figure 10 Finding definition of “apple”

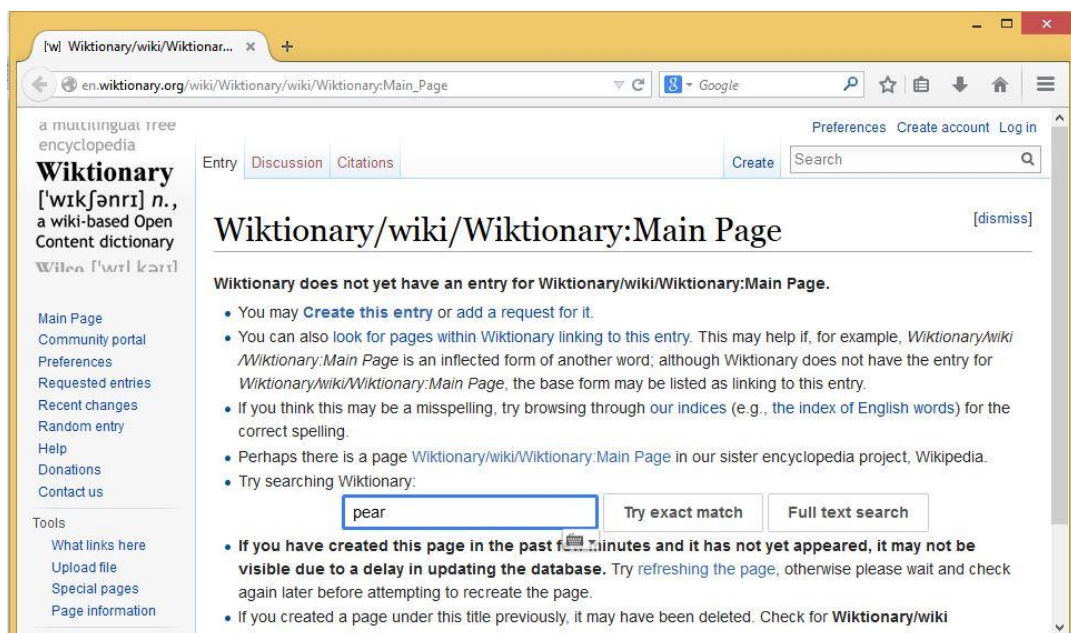


Figure 11 Finding definition of “pear”

5. Obtaining the documentation for the executed test cases

- click **Start** and open a Command prompt window with **cmd**
- Remark: execute the command from the project directory
- ...>**mvn serenity:aggregate**
- The generated report will be saved in the folder **\target\site\serenity**

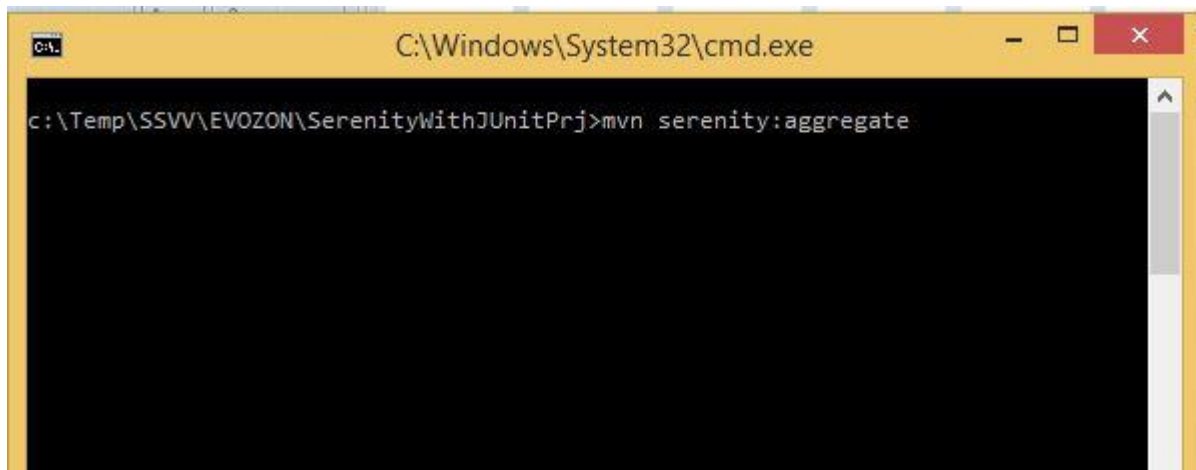


Figure 12 Generating the Serenity report

6. Viewing the Serenity report

- in the project directory ...target\site\serenity\index.html

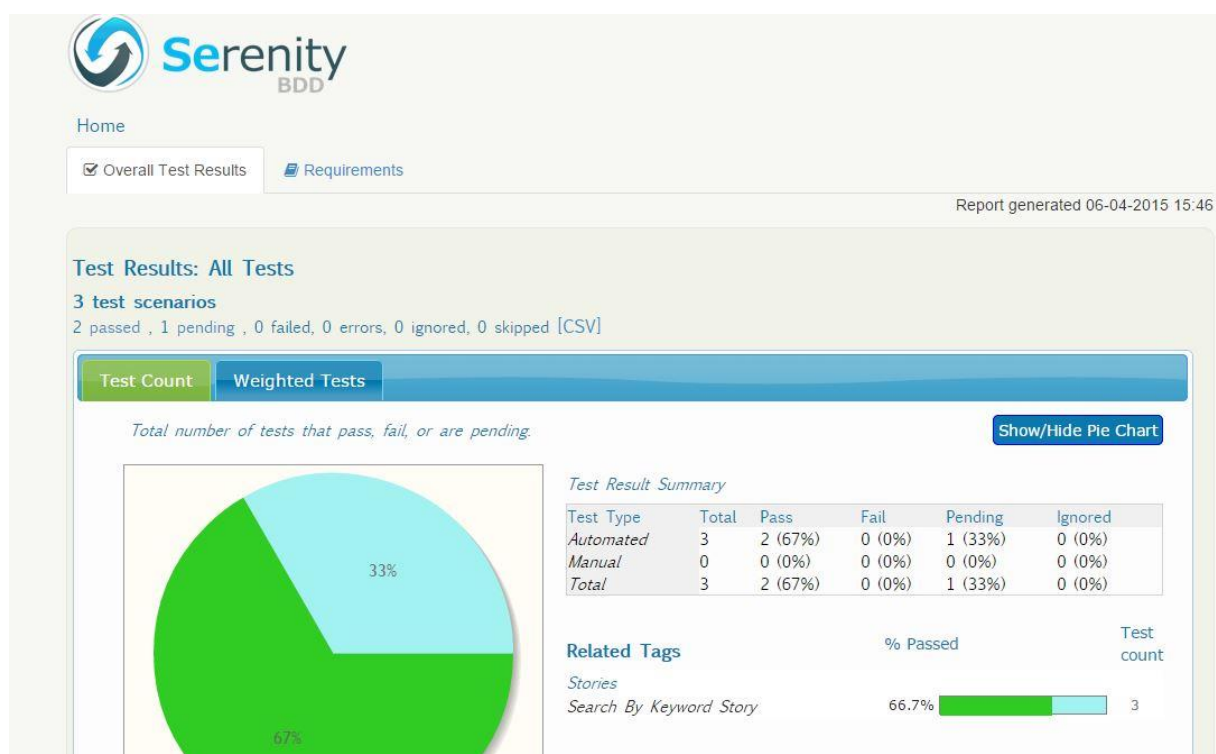


Figure 13 Serenity report

7. Test Data Driven - data for test from csv file

- create the csv file *WikiTestData.csv* with the content:
 - first line indicate the structure of the table with input data
- next lines contain input data for individual test cases.

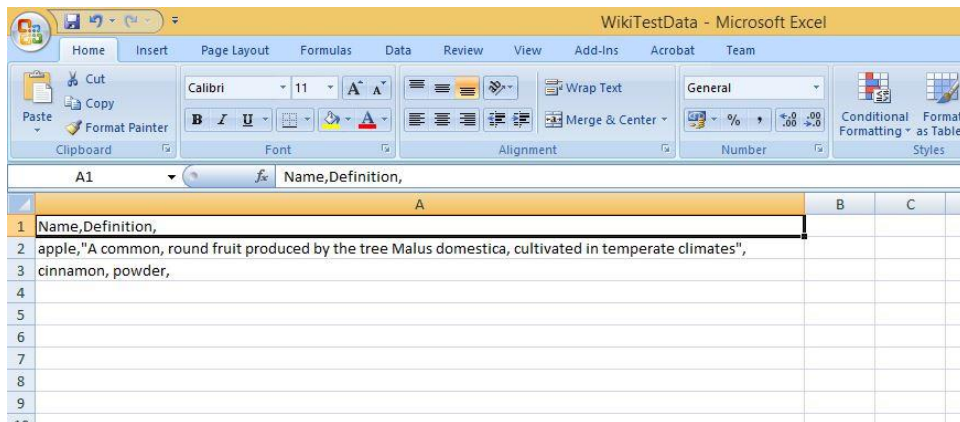


Figure 14 Create the file with the test cases data

- add the csv file to the *src/test/resources* directory
- add a new class to run with Ddt with Parameterized Runner
- run (see Section 4 of the current document)
- obtaining documentation (see Section 5 of the current document)
- view the serenity report (see Section 6 of the current document)

```
20
21 @RunWith(SerenityParameterizedRunner.class)
22 @UseTestDataFrom("src/test/resources/WikiTestData.csv")
23 public class SearchByKeywordStoryDdt {
24
25     @Managed(uniqueSession = true)
26     public WebDriver webdriver;
27
28     @ManagedPages(defaultUrl = "http://en.wiktionary.org/wiki/Wiktionary")
29     public Pages pages;
30
31     public String name;
32     public String definition;
33
34     @Qualifier
35     public String getQualifier() {
36         return name;
37     }
38
39     @Steps
40     public EndUserSteps endUser;
41
42     @Issue("#WIKI-1")
43     @Test
44     public void searchWikiByKeywordTestDDT() {
45         endUser.is_the_home_page();
46         endUser.looks_for(getName());
47         endUser.should_see_definition(getDefinition());
48     }
49 }
```

Figure 15 Test case class that is parametrized using a csv file