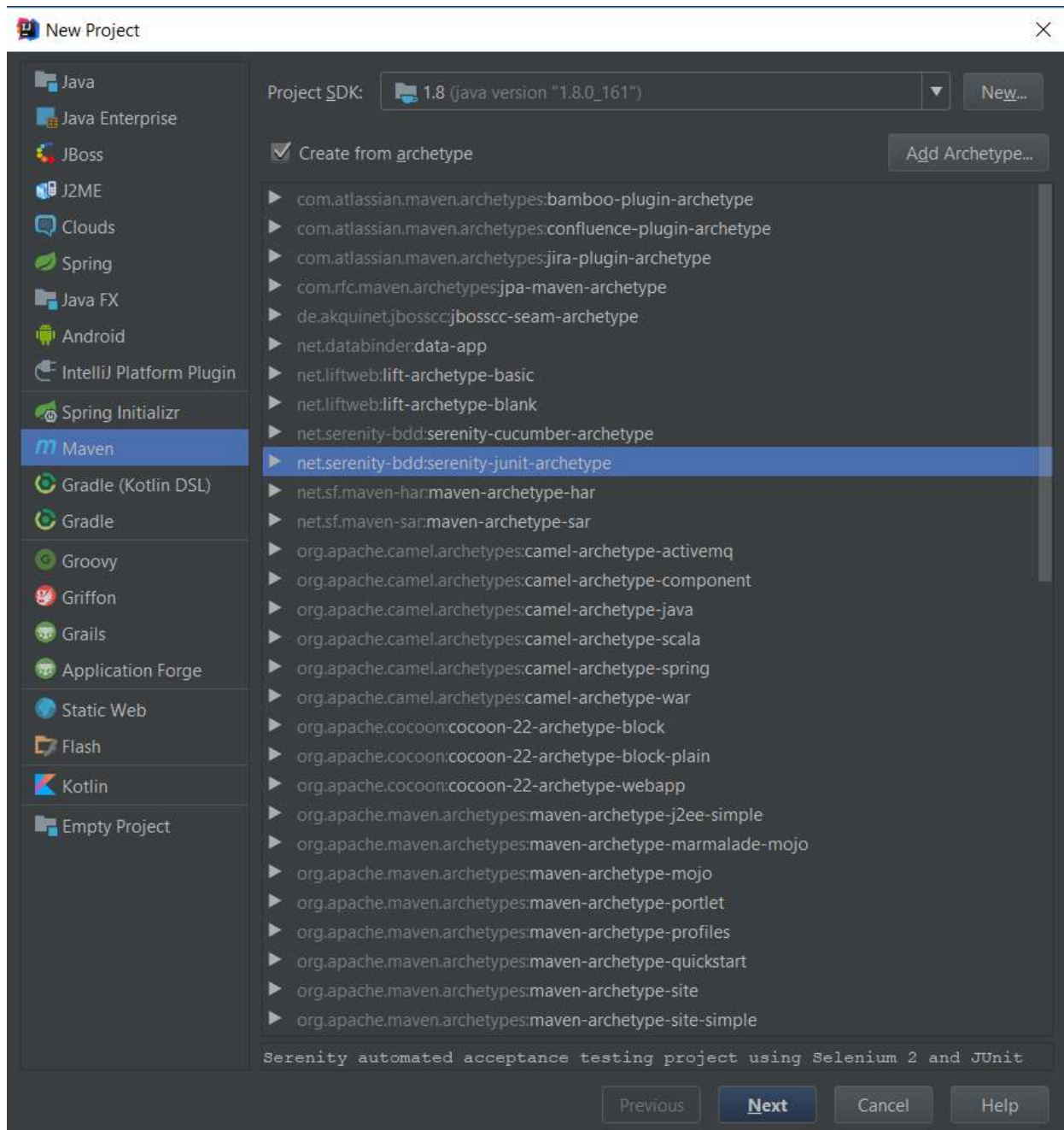


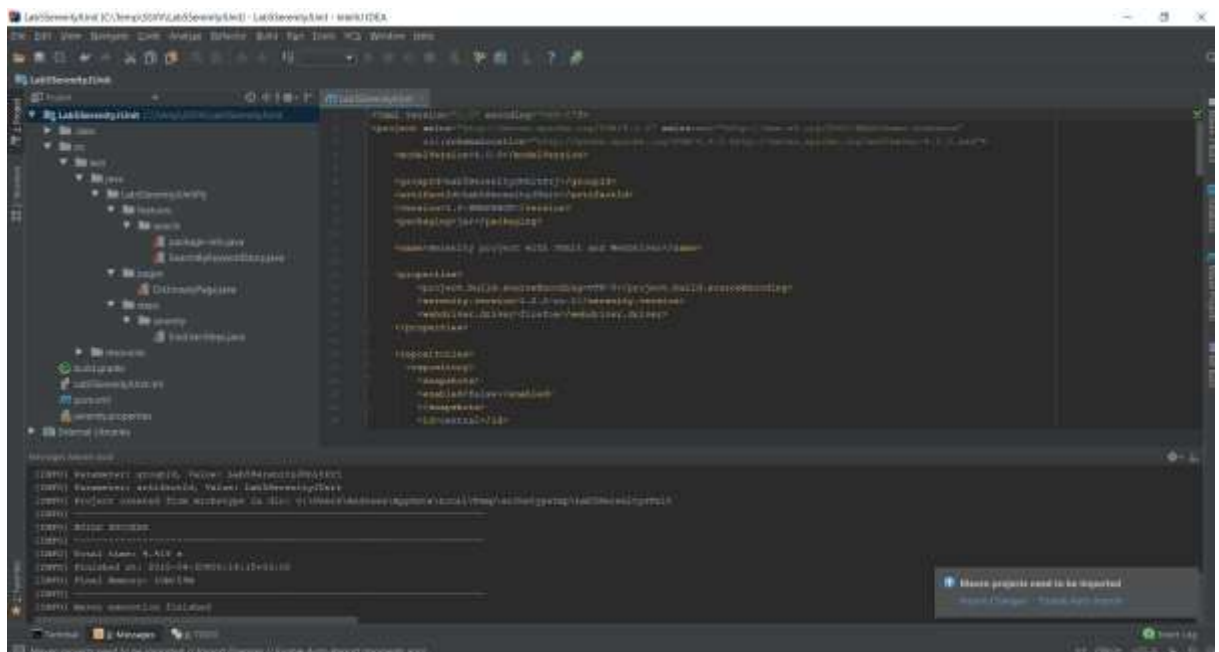
Tutorial Web UI Automation serenity-junit-archetype Serenity + JUnit

1. Create Maven Project (serenity-junit-archetype)

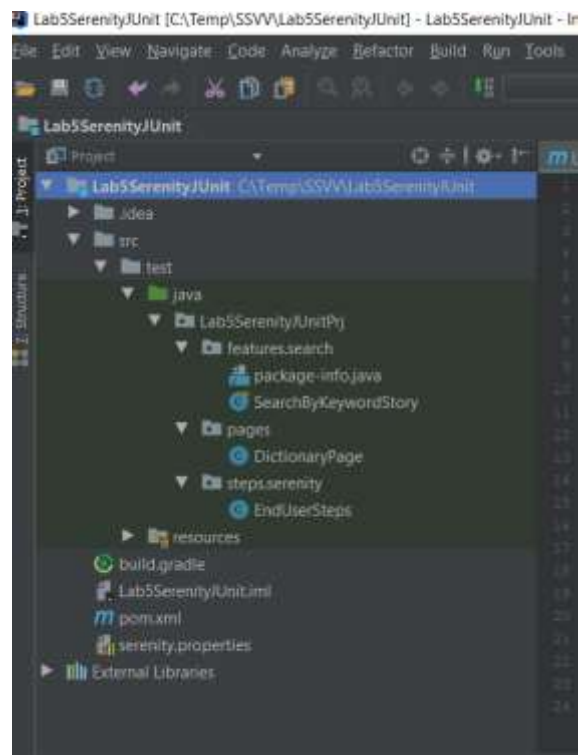
- IntelliJ → File → New → Project → Maven
 - o Create from archetype: **serenity-junit-archetype**



- After creating the project, check “**Enable Auto-Import**”



- The Maven project will have the following structure



2. Setting the web browser to run the tests

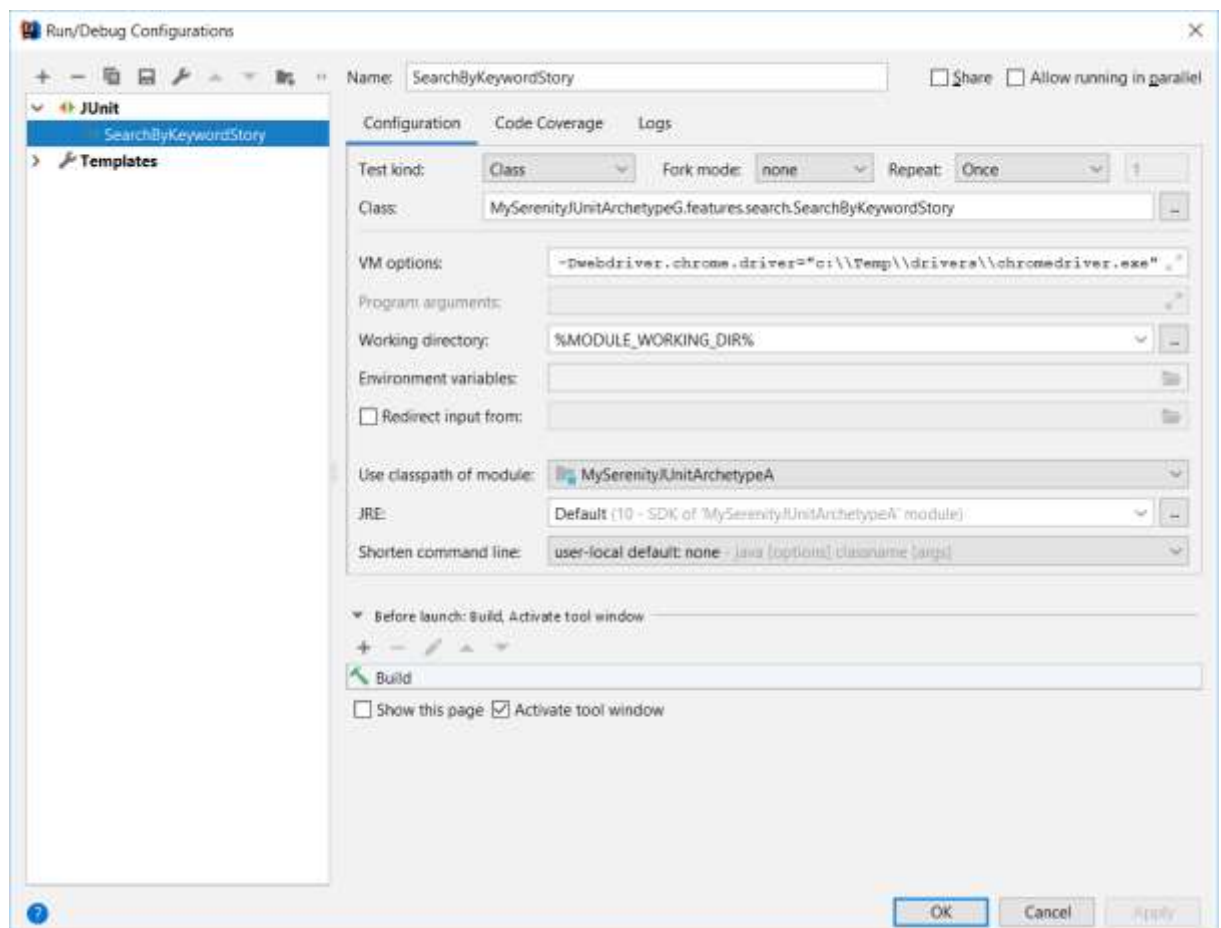
- In the pom.xml file Firefox driver is set implicitly
- Driver for chrome
 - o <https://sites.google.com/a/chromium.org/chromedriver/downloads>
- the drivers are downloaded, unzipped and saved to a folder, eg c:\Temp\drivers, from where they can be used later by any testing project;
- the path to the folder containing the web browser drivers, i.e. c:\Temp\drivers, is added to the Path environment variable
- to modify the browser used to run the test, the file type pom.xml changes to the driver, e.g., chrome

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <serenity.version>1.2.2-rc.1</serenity.version>
  <webdriver.driver>chrome</webdriver.driver>
</properties>
```

- add VM options:

- o Run->Edit configurations....-> VM options:

-Dwebdriver.chrome.driver="c:\\Temp\\drivers\\chromedriver.exe"



3. Run as JUnit test

- In ProjectExplorer- Right-click on a TestCase (e.g. **SearchByKeywordStory**) and select Run
- The Chrome browser is opened and the definitions for "pear" and "apple" are searched.



4. Obtaining the documentation for the executed test cases

- Add in the file "serenity.properties" file in the **IntelliJ** project

`webdriver.driver=chrome`

`webdriver.chrome.driver = C:\\Temp\\drivers\\chromedriver.exe`

[Option 1]

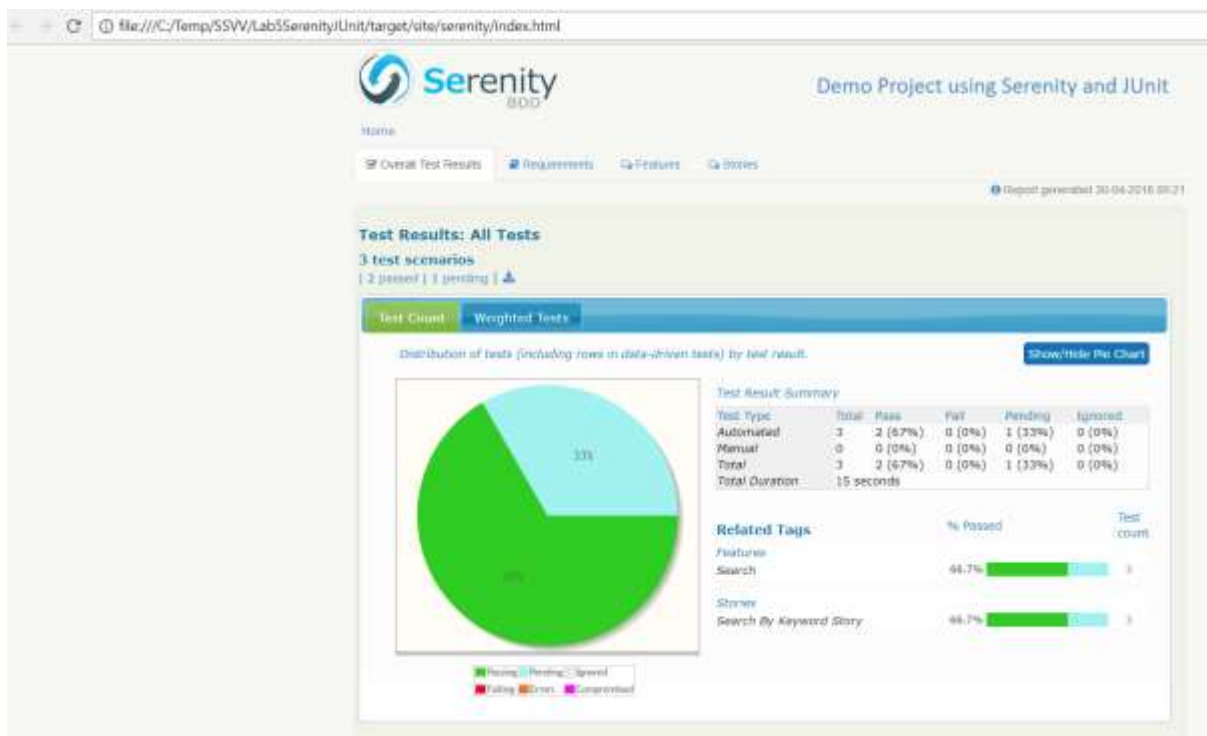
- **IntelliJ** → **View** → **Tool Windows** → **Maven Projects**
- Select *Lifecycle* → the "verify" option
- the generated report will be saved in the project folder in `\target\site\serenity;`

[Option 2]

- 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**

5. Viewing the Serenity report

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



6. 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.

The screenshot shows an Excel spreadsheet with the following data:

Name	Definition
love	Strong affection.
child	A person who has not yet reached adulthood, whether natural (puberty), cultural (initiation), or legal (majority); (obsolete, specifically) a female child, a girl.
happiness	The emotion of being happy; joy.

- add the csv file to the src/test/resources directory
- add a new class to run with Ddt with Parameterized Runner (see the class below)
- run (see Section 3 of the current document)
- obtaining documentation (see Section 4 of the current document)
- view the serenity report (see Section 5 of the current document)

```
• package WikiExampleGroup.features.search;

import net.serenitybdd.junit.runners.SerenityParameterizedRunner;
import net.thucydides.core.annotations.Issue;
import net.thucydides.core.annotations.Managed;
import net.thucydides.core.annotations.ManagedPages;
import net.thucydides.core.annotations.Steps;
import net.thucydides.core.pages.Pages;
import net.thucydides.junit.annotations.Qualifier;
import net.thucydides.junit.annotations.UseTestDataFrom;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.openqa.selenium.WebDriver;

import Lab5SerenityJUnitPrj.steps.serenity.EndUserSteps;

@RunWith(SerenityParameterizedRunner.class)
@UseTestDataFrom("src/test/resources/WikiTestData.csv")
public class SearchByKeywordStoryDdt {

    @Managed(uniqueSession = true)
    public WebDriver webdriver;

    @ManagedPages(defaultUrl = "http://en.wiktionary.org/wiki/Wiktionary")
    public Pages pages;

    public String name;
    public String definition;

    @Qualifier
    public String getQualifier() {
        return name;
    }

    @Steps
    public EndUserSteps endUser;

    @Issue("#WIKI-1")
    @Test
    public void searchWikiByKeywordTestDDT() {
        endUser.is_the_home_page();
        endUser.looks_for(getName());
        endUser.should_see_definition(getDefinition());
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDefinition() {
```

```
        return definition;
    }

    public void setDefinition(String definition) {
        this.definition = definition;
    }
}
```

Test case class that is parametrized using a csv file