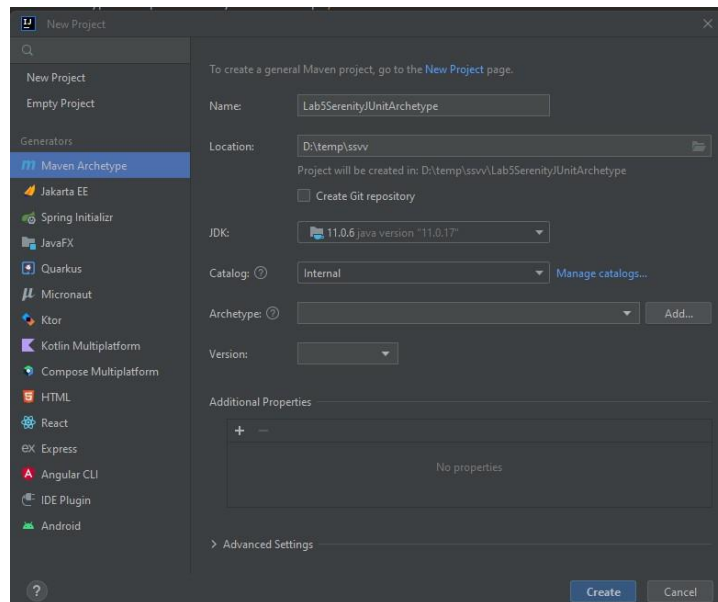


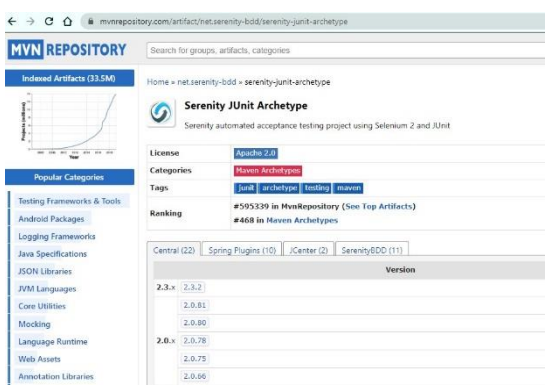
## Tutorial Web UI Automation serenity-junit-archetype Serenity + JUnit

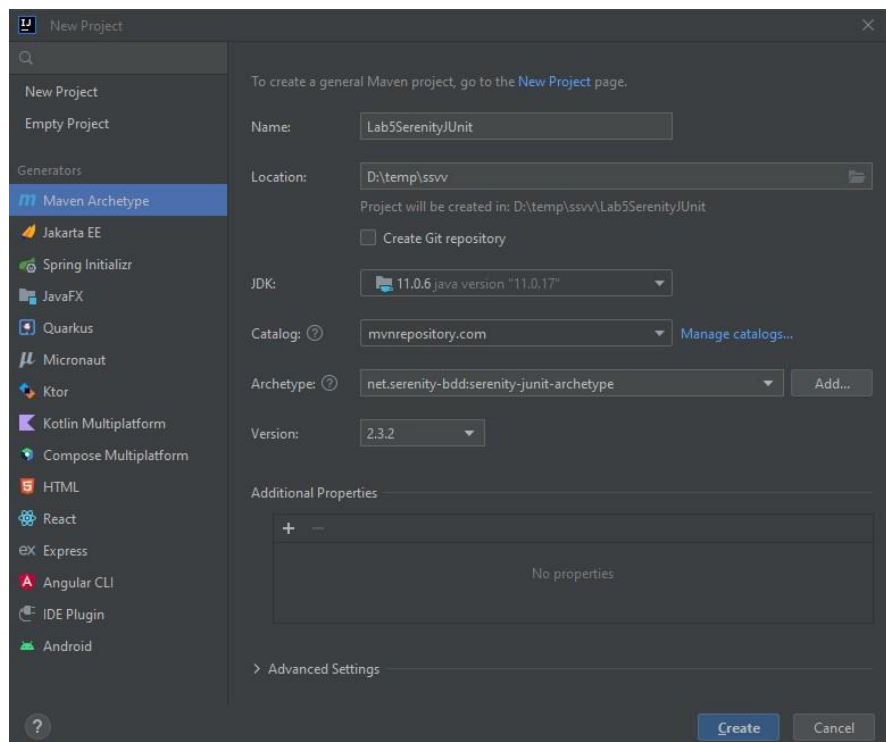
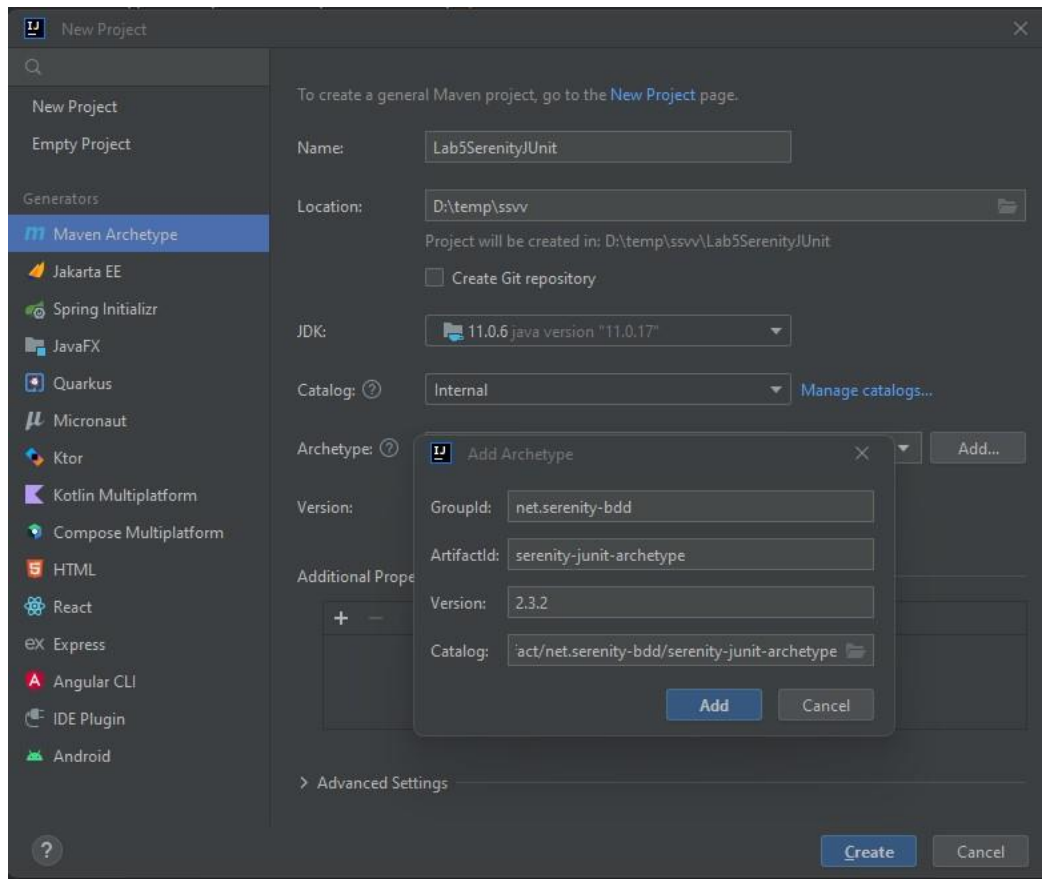
### 1. Create Maven Project (serenity-junit-archetype)

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

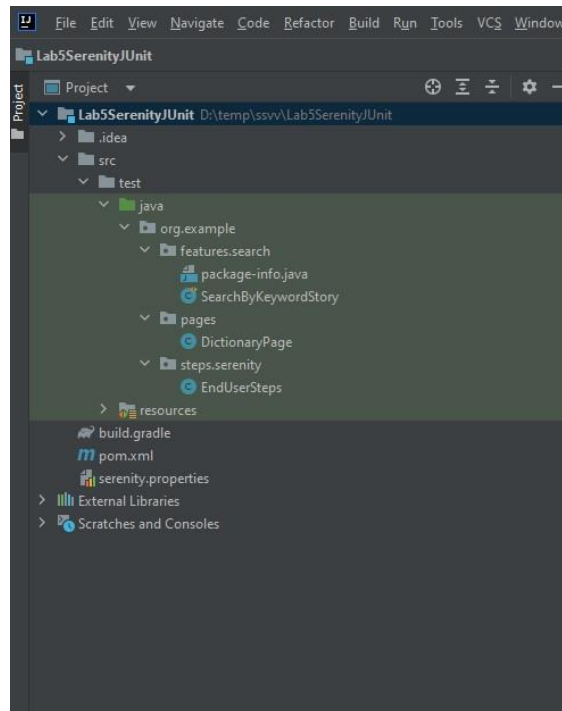


<https://mvnrepository.com/artifact/net.serenity-bdd/serenity-junit-archetype/2.3.2>





- The Maven project will have the following structure



## 2. Setting to run with headless mode to false

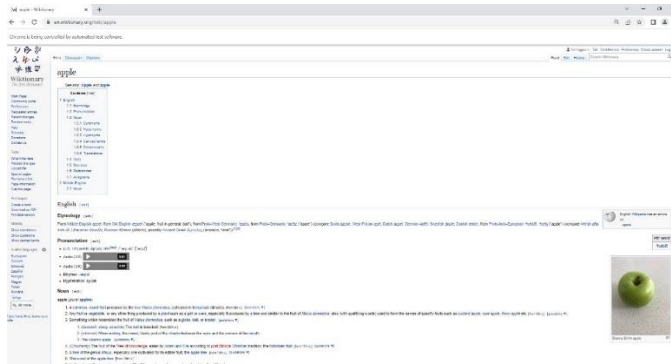
In file serenity.properties change the headless.mode= true to false. Run as JUnit test

- 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.

**Remarks:** If it does not work, download the chromedriver.exe and place it in a directory at your choice and specify it in the file serenity.properties in the project.

```
# Appears at the top of the reports
serenity.project.name = Demo Project using Serenity and JUnit
webdriver.autodownload = false
webdriver.driver = chrome
headless.mode = false
```

```
drivers {
    windows {
        webdriver.chrome.driver =
src/test/resources/webdriver/windows/chromedriver.exe
    }
}
```



### 3. Obtaining the documentation for the executed test cases

[Option 1]

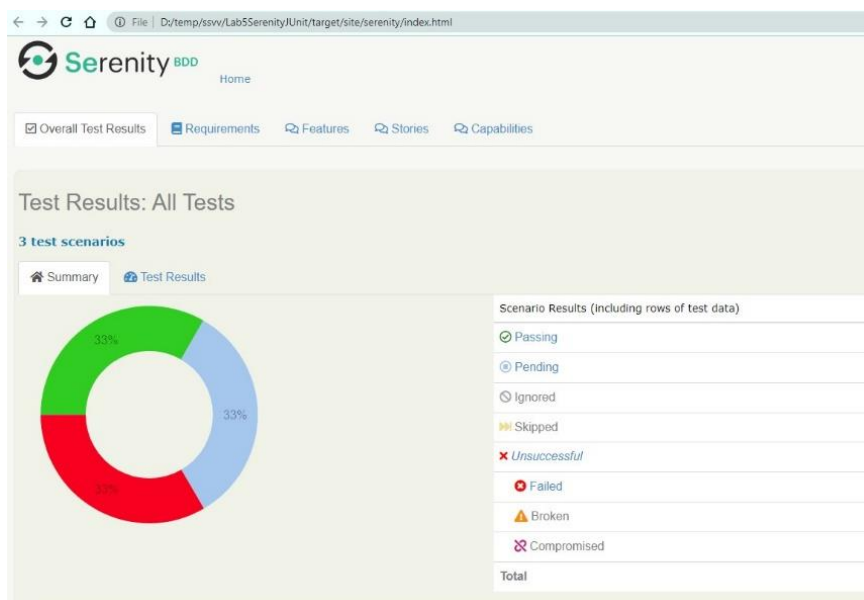
- **IntelliJ → View → Tool Windows → Maven**
- Select *Lifecycle* → the “verify” option
- the generated report will be saved in the project folder in `\target\site\serenity`;
- open `index.html`

[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`

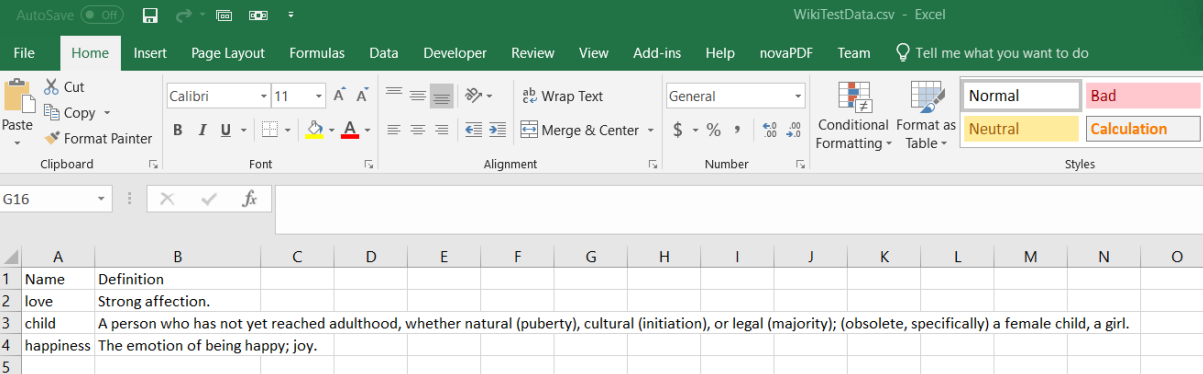
### 4. Viewing the Serenity report

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



## 5. 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.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Definition													
2	love	Strong affection.													
3	child	A person who has not yet reached adulthood, whether natural (puberty), cultural (initiation), or legal (majority); (obsolete, specifically) a female child, a girl.													
4	happiness	The emotion of being happy; joy.													
5															

- 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 org.example.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.Steps;
import net.thucydides.junit.annotations.Qualifier;
import net.thucydides.junit.annotations.UseTestDataFrom;
import org.example.steps.serenity.EndUserSteps;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.openqa.selenium.WebDriver;

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

    @Managed(uniqueSession = true)
    public WebDriver webdriver;

    @Steps
    public EndUserSteps endUser;

    public String name;
    public String definition;

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

    @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**