Introduction & Configuration

Video

Serenity Intro Practice.

The objective of this practice is to guide you through the installation process of Serenity BDD, in order to use it for test automation and behavior-driven development (BDD).

As a prerequisite, you will need to have installed Java Development Kit (JDK) 8 or higher, along with your preferred IDE.

Start with creating on your IDE a new Java Maven project with the name "SerenityBDDIntro". Add the following Maven dependencies to work with Serenity BDD:

```
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-core</artifactId>
    <version>2.5.12</version> <!-- Replace with the latest version -->
</dependency>
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-junit</artifactId>
    <version>2.5.12</version> <!-- Replace with the latest version -->
    <scope>test</scope>
</dependency>
```

Feel free to modify the version of the dependencies according to your needs. After the dependencies are added, make sure to install them.

Now we are ready to create our first Serenity BDD Test:

Start by creating a new SerenityTest class in your project, and write a simple test using the @Narrative and @WithTags annotations. Your class should look like this (Don't forget to add the imports for each required element)

```
@Narrative(text = {"This is a sample Serenity BDD test."})
@WithTags({@WithTag("sample")})
public class SerenityTest {
    @Steps
    SampleSteps sampleSteps;
```

```
    @Test
    public void verifySerenityBDDSetup() {
        sampleSteps.performSomeAction();
        sampleSteps.verifyResult();
    }
}
```

As the test is already created, then you can create your own step definitions, on a new SampleSteps Java class. Your SampleSteps Class should look like this:

```
public class SampleSteps {
    @Step
    public void performSomeAction() {
        // Implement your test steps here
    }
    @Step
    public void verifyResult() {
        // Implement assertions or verifications here
    }
}
```

When the class is created, add code in the performSomeAction() method to sum two numbers received as parameters, and assign the result to a SampleSteps class attribute. Then, add an assertion on the verifyResult() method to check if the value of the previously assigned attribute matches the result of the sum of the two given numbers.

Run your test using your IDE, and then, review the Serenity BDD report generated in the target/site/serenity directory. It provides detailed information about the test execution.

Congratulations! You have successfully completed the installation of Serenity BDD and created a simple Serenity BDD test project.

Additional Resources

What is Serenity

Configuring Selenium

Serenity Page Object, Web Elements

Videos

You will find that in this section there is no loaded exercise to put your new acquired skills into practice. This is because you will integrate this content with the following. Remember to lean on your team to move forward and resolve any doubts that may arise.

Additional Resources

Serenity Page Elements

Interacting with Web Pages

JUnit Tutorial

JUnit Tutorial with example

Junit Annotations, tags and assertions

Video

Serenity Page Object and JUnit Practice.

In this practice, we are going to create our first automation framework using Serenity and Junit. In order to achieve this, start by creating a maven java project including the latest version of Serenity BDD and Junit.

Now we are almost ready to start, check the following link in order to get familiarized with basic Serenity BDD actions. **https://serenity-bdd.github.io/docs/tutorials/first_test**

Now, we are going to create our own Selenium scripts. Use the methods and interactions seen on in the previous example to implement, using Page Objects and Junit Annotations, the following actions:

1. Navigate to the webpage (**https://wikipedia.org**)
2. Write on the search box in the web page the text “Microsoft”
3. Click on the search button aside the search box
4. Wait for the new webpage to load
5. Get the title of the new webpage and check if it matches with “Microsoft”, print a message in the console indicating if it matches or not.
6. Get all the anchor links on the web page (hint: find by HTML tag “a”), and for each one of them, print in the console its respective text.

Note: For steps a, b, c, d, e, implement those actions on only one Test. For step f, implement another separate Test.

First Framework Overview

Video

As in the previous meeting, we have an exercise that transcends this topic and must be resolved in this meeting and the next two. You will be able to view it but remember to repeat it so you can add the new topics studied.

Serenity Interactions and Actors Practice.

You are automating the **https://www.saucedemo.com/** web application.

Create a new Serenity BDD project, and configure it to launch and test the required webpage on chrome browser. Feel free to use the Serenity base framework found at **https://github.com/serenity-bdd/serenity-junit-starter**. Implement on your framework the following scenarios, each one as a different test:

- Log In: Use the standard credentials displayed on the webpage.
- Purchase a product: Follow the complete buyflow of the page, selecting a random product, adding it to the cart, adding the personal data, and check you are successfully arriving at the “Thank you for your purchase” page.
- Removing elements of the shopping cart: Add 3 different elements to the shopping cart, enter to the cart page, remove them and check the shopping cart is empty.
- Logout: try to log out and check if you are correctly redirected to the login page.

For this implementation, follow the rules described next:

- Make sure to use Page Objects, Interactions and Actors as needed.
- Before annotations should be used to manage preconditions. Any other required annotations can be implemented as needed.

Additional Resources

Git Serenity Cucumber Starter

Your first Web Test

UI Interactions

Video

Remember that you must continue applying what you learned in this exercise.

Serenity Interactions and Actors Practice.

You are automating the **https://www.saucedemo.com/** web application.

Create a new Serenity BDD project, and configure it to launch and test the required webpage on chrome browser. Feel free to use the Serenity base framework found at **https://github.com/serenity-bdd/serenity-junit-starter**. Implement on your framework the following scenarios, each one as a different test:

- Log In: Use the standard credentials displayed on the webpage.
- Purchase a product: Follow the complete buyflow of the page, selecting a random product, adding it to the cart, adding the personal data, and check you are successfully arriving at the “Thank you for your purchase” page.
- Removing elements of the shopping cart: Add 3 different elements to the shopping cart, enter to the cart page, remove them and check the shopping cart is empty.
- Logout: try to log out and check if you are correctly redirected to the login page.

For this implementation, follow the rules described next:

- Make sure to use Page Objects, Interactions and Actors as needed.
- Before annotations should be used to manage preconditions. Any other required annotations can be implemented as needed.

Additional Resources

Serenity Step Libraries

Serenity Java

Scenarios and Actors

Video

Today is the last meeting to polish and improve your resolution of this exercise.

Serenity Interactions and Actors Practice.

You are automating the **https://www.saucedemo.com/** web application.

Create a new Serenity BDD project, and configure it to launch and test the required webpage on chrome browser. Feel free to use the Serenity base framework found at **https://github.com/serenity-bdd/serenity-junit-starter**. Implement on your framework the following scenarios, each one as a different test:

- Log In: Use the standard credentials displayed on the webpage.
- Purchase a product: Follow the complete buyflow of the page, selecting a random product, adding it to the cart, adding the personal data, and check you are successfully arriving at the “Thank you for your purchase” page.
- Removing elements of the shopping cart: Add 3 different elements to the shopping cart, enter to the cart page, remove them and check the shopping cart is empty.
- Logout: try to log out and check if you are correctly redirected to the login page.

For this implementation, follow the rules described next:

- Make sure to use Page Objects, Interactions and Actors as needed.
- Before annotations should be used to manage preconditions. Any other required annotations can be implemented as needed.

Additional Resources

Web Testing with Serenity Screenplay

Reporter

Video

Serenity Reporter Practice.

You are automating the **https://www.saucedemo.com/** web application.

Create a new Serenity BDD project, and configure it to launch and test the required webpage on chrome browser. Feel free to use the Serenity base framework found at **https://github.com/serenity-bdd/serenity-junit-starter**. Implement on your framework the following scenarios, each one as a different test:

- Log In: Use the standard credentials displayed on the webpage.
- Purchase a product: Follow the complete buyflow of the page, selecting a random product, adding it to the cart, adding the personal data, and check you are successfully arriving at the “Thank you for your purchase” page.
- Removing elements of the shopping cart: Add 3 different elements to the shopping cart, enter to the cart page, remove them and check the shopping cart is empty.
- Logout: try to log out and check if you are correctly redirected to the login page.

For this implementation, follow the rules described next:

- Make sure to use Page Objects, Interactions and Actors as needed.
- Before annotations should be used to manage preconditions. Any other required annotations can be implemented as needed.

After this implementation is complete, configure the following options in order to modify the report generated by Serenity:

- Add 2 Custom Fields to your reporter properties, to include the Application Version and the environment (you can add them on the serenity properties file). Feel free to assign any values to these fields. You can use the Custom Fields Serenity Documentation in order to understand better how to modify these properties. Make sure this information is displayed on your report.
- Use the sysinfo properties in order to add additional information to your report. the report should include in a Test Run Section the name of the person who executed, and the java version used. Make sure this information is displayed on your report.
- Add a Readme.md file into the folder including your scenarios in order to explain the requirements needed for all of your tests to run, including any needed information.

Make sure this information is shown into the requirements section of your generated report.

Additional Resources

The Serenity Reference manual

Screenplay Pattern Intro

Video

In this exercise you will be able to apply what you learn in this meeting and also in tomorrow's meeting. If you don't finish it, don't worry, in the next meeting you can continue.

Screenplay Pattern Practice.

It's time to start creating your first framework using Serenity BDD and the Screenplay Pattern. In order to achieve this, access to this link: **https://github.com/serenity-bdd/serenity-articles/blob/master/screenplay-tutorial/screenplay-tutorial.adoc**

On the link you will find initial documentation and overview of how a Screenplay framework should look like. Make sure you look at all the information there in order to understand better how the pattern works. Once you have reviewed from point 1 to 3, you are ready to start with your implementation.

Start implementing your own framework creating a new Maven - Serenity Framework (As explained in points 1 to 3) and then, follow the instructions from point **4. Your first Serenity Screenplay test** , make sure to follow all steps for your framework to run correctly.

Note: The webpage to automate on this exercise is available at: **https://example.cypress.io/todo**

Additional Resources

Screenplay Fundamentals

Actors, Abilities, Tasks, Questions

Video

Screenplay Pattern Practice.

It's time to start creating your first framework using Serenity BDD and the Screenplay Pattern. In order to achieve this, access to this link: **https://github.com/serenity-bdd/serenity-articles/blob/master/screenplay-tutorial/screenplay-tutorial.adoc**

On the link you will find initial documentation and overview of how a Screenplay framework should look like. Make sure you look at all the information there in order to understand better how the pattern works. Once you have reviewed from point 1 to 3, you are ready to start with your implementation.

Start implementing your own framework creating a new Maven - Serenity Framework (As explained in points 1 to 3) and then, follow the instructions from point **4. Your first Serenity Screenplay test** , make sure to follow all steps for your framework to run correctly.

Note: The webpage to automate on this exercise is available at: **https://example.cypress.io/todo**

Additional Resources

Serenity and the Screenplay Pattern

Live coding session

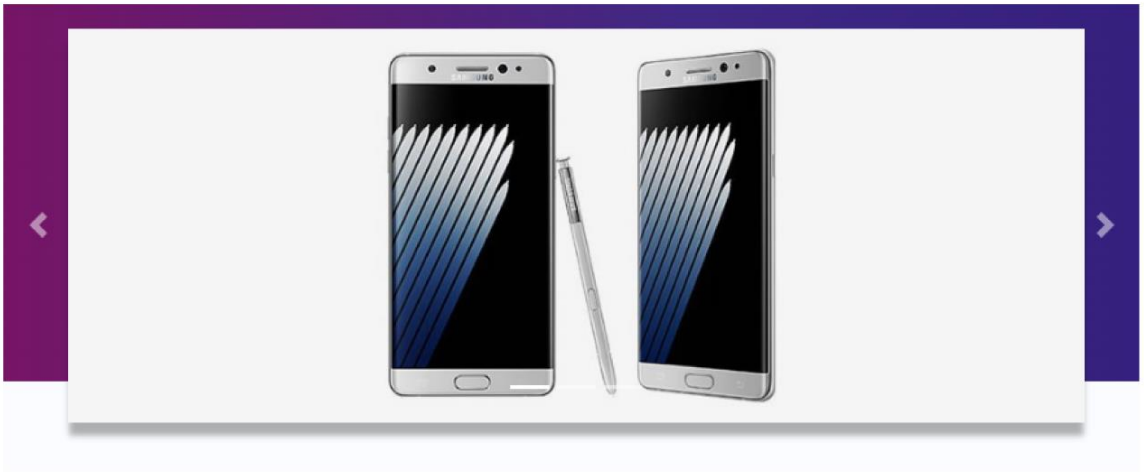
Video

Screenplay Final Practice.

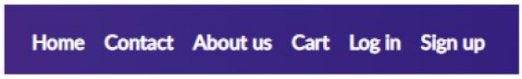
For this practice, we need to create a framework from scratch using serenity and Gherkin Language cucumber to test the web page: **https://www.demoblaze.com/**. Feel free to use the oficial Screenplay Cucumber Template found at **https://github.com/serenity-bdd/serenity-cucumber-starter** , and the documentation at: https://serenity-bdd.github.io/docs/tutorials/cucumber-screenplay

The framework should include test cases for the following functionalities:

1. Test the featured items carousel. It should be in a feature including at least 3 scenarios.



2. Test Navigation to all header links. It should be in a feature including at least 3 scenarios.



3. Test the categories selection, product selection, adding product to cart, buying cart, removing element from cart. All of the defined scenarios for these functionalities should be on a feature.

Keep in mind the following instructions and tips:

- The framework should be organized using best practices and following what suits best to the software under test.
- All defined scenarios should be tagged in any way you decide.
- The framework should include the use of the screenplay pattern.

- Test cases to implement should be designed following best practices and correct organization. Tip: test cases should be fully independent one from another.
- The use of cucumber is mandatory.

Additional Resources

Screenplay Pattern approach using Selenium and Java