

Automating the SwagLabs application using Cucumber & Maven

Table of Contents

<i>Introduction.....</i>	<i>- 2 -</i>
<i>Problem Statement.....</i>	<i>- 2 -</i>
<i>Solution Details</i>	<i>- 3 -</i>
<i>Project Structure.....</i>	<i>- 4 -</i>
<i>Snapshots</i>	<i>- 4 -</i>
<i>Conclusion</i>	<i>- 6 -</i>

Introduction

- This is the project report for the Project 1: Automating the SwagLabs application using Cucumber for “Get started with Functional Testing, TDD, and DevOps Integration” course of Simplilearn.
- Project submitted by : Krishnaveni Rajan
- Email Id: krishnaveni07.rajan@gmail.com
- Submission Date: 25-July-2021
- Language/Tools Used : Core Java, Cucumber, Gherkin language with Selenium Web driver. Project written in Eclipse in Maven + Cucumber format and uploaded to GitHub
- CI/CD : Jenkins for GitHub and Maven

Problem Statement

To automate the SwagLabs to verify the Login and Product feature
Background of the problem statement:

Login.feature

Scenario outline

- 1) Verify login functionality with correct username and password
- 2) Verify error message "Epic sadface: Username and password do not match any user in this service", when incorrect username and password is provided.

Product.feature

Background step for login

- 1) Verify that user is able to add the "<single product>" to cart -> Scenario outline
- 2) Verify that user is able to add the "<multiple product>" to cart using datatable -> Scenario

The following requirements should be met:

- Hooks to be used
 - Before hook to initialize the driver
 - After hook to (1) Quit
- Use runner file for test case execution
- Run the features from command outline
- Create the Jenkins job to run the cucumber test

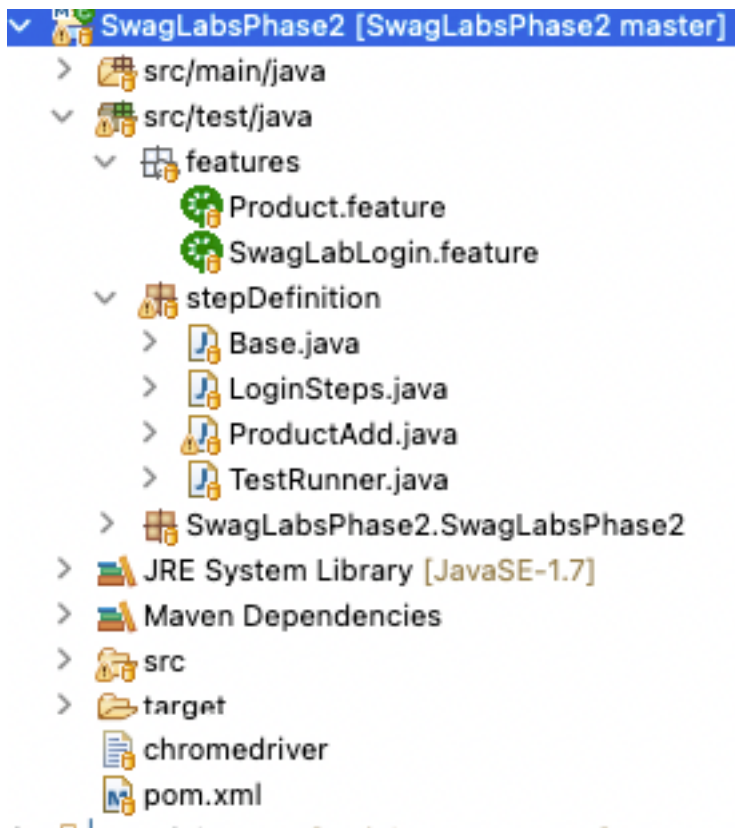
Solution Details

Solution to the product has been achieved using the Cucumber. The solutions involves following steps

- Created Maven project using 'maven-archetype-quickstart' arch type of apache maven and then configure to Cucumber type
- Create feature and stepsDefinition package.
- Login Feature
 - Add 'SwagLabLogin.feature' under feature
 - Add Base java class with Before and After hooks
 - Add LoginSteps java class to login for Positive and Negative scenarios
 - Add TestRunner java class
 - Execution done
 - Cucumber
 - Junit
 - Command Line – mvn test
 - Commit code to GitHub
 - Jenkins Execution
 - Mvn
 - Git Hub
- Product Feature
 - Add 'Product.featureunder feature
 - Add ProductAdd java class to login for Single product and Multiple product using Datatable scenarios
 - (I have used static xPath for time being, will try to do with dynamic and loops post submission)
 - Update TestRunner java class
 - Execution done
 - Cucumber
 - Junit
 - Command Line – mvn test
 - Commit code to GitHub
 - Jenkins Execution
 - Mvn

- Git Hub
- Push the code to GitHub @ <https://github.com/krajan07/SwagLabsPhase2>

Project Structure



Snapshots

Jenkins Link - <http://localhost:8080/>

S	W	Name	Last Success	Last Failure	Last Duration
✓	🔍	SwagLabsPhase2	25 min - #5	N/A	30 sec
✓	🔍	SwagLabsPhase2_Git	26 min - #9	1 hr 45 min - #3	2.4 sec

Jenkins

Dashboard > SwagLabsPhase2_Git >

[Back to Dashboard](#)

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Cucumber reports

Build History

trend ^

find

#9

Jul 25, 2021, 12:18 PM

#8

Jul 25, 2021, 12:15 PM

#7

Jul 25, 2021, 12:11 PM

#6

Jul 25, 2021, 12:09 PM

Project SwagLabsPhase2_Git

/Users/renu/ProjectSubmissions/SwagLabsPhase2

Cucumber reports

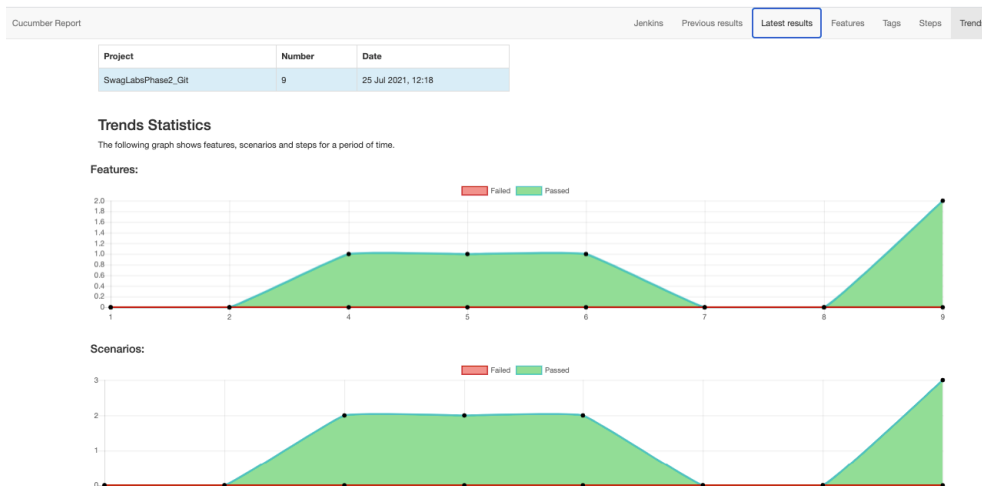
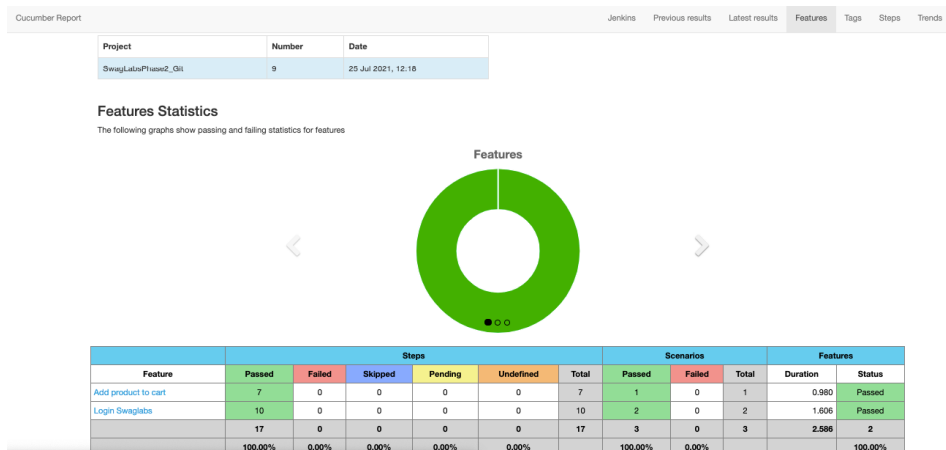
Workspace

Recent Changes

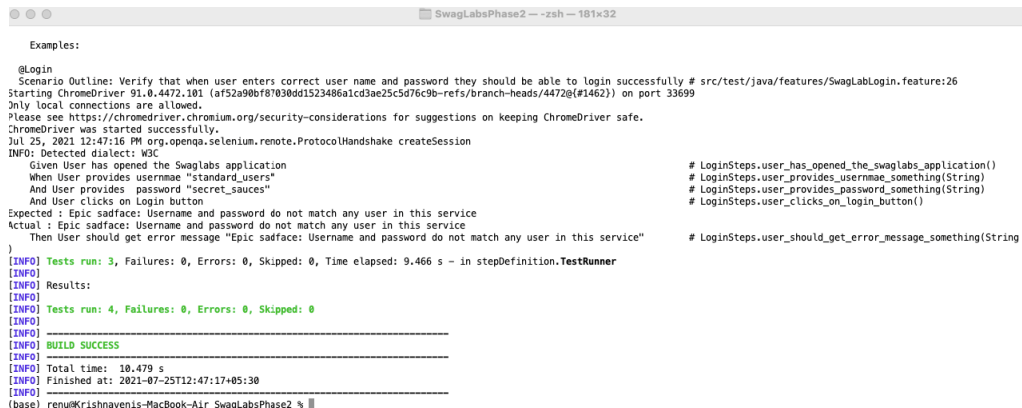
Permalinks

- Last build (#9), 27 min ago
- Last stable build (#9), 27 min ago
- Last successful build (#9), 27 min ago
- Last failed build (#3), 1 hr 45 min ago
- Last unsuccessful build (#3), 1 hr 45 min ago
- Last completed build (#9), 27 min ago

Cucumber Report



Command Line : mvn test

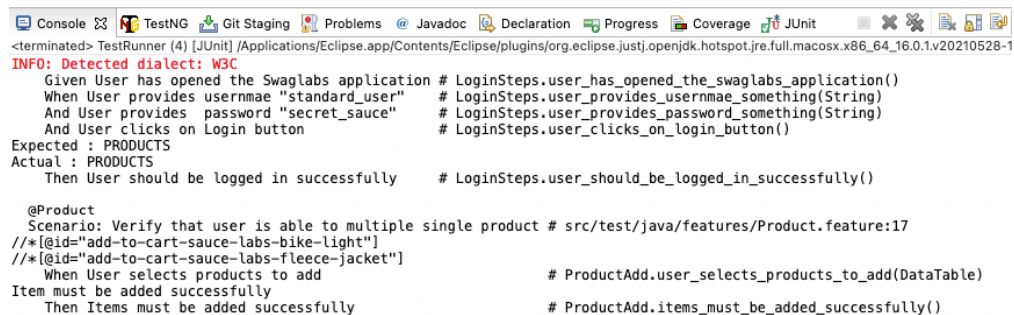


```
SwagLabsPhase2 -- -zsh -- 181x32

Examples:

@Login
Scenario Outline: Verify that when user enters correct user name and password they should be able to login successfully # src/test/java/features/SwagLabLogin.feature:26
Starting ChromeDriver 91.0.4472.101 (af52a9b8f87038dd1523486a1cd3ae25c5d76c9b-refs/branch-heads/4472@{#1462}) on port 33699
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 25, 2021 12:47:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Given User has opened the Swaglabs application
When User provides usermae "standard_users" # LoginSteps.user_has_opened_the_swaglabs_application()
And User provides password "secret_sauces" # LoginSteps.user_provides_usermae_something(String)
And User clicks on Login button # LoginSteps.user_provides_password_something(String)
# LoginSteps.user_clicks_on_login_button()
Expected : Epic sadface: Username and password do not match any user in this service
Actual : Epic sadface: Username and password do not match any user in this service
Then User should get error message "Epic sadface: Username and password do not match any user in this service" # LoginSteps.user_should_get_error_message_something(String)
)
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.466 s - in stepDefinition.TestRunner
[INFO] Results:
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO] BUILD SUCCESS
[INFO] Total time: 10.479 s
[INFO] Finished at: 2021-07-25T12:47:17+05:30
[INFO]
(base) renu@Krishnavenis-MacBook-Air: SwagLabsPhase2 %
```

Eclipse Console OutPut



```
<terminated> TestRunner (4) [JUnit] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_16.0.1.v20210528-1
INFO: Detected dialect: W3C
Given User has opened the Swaglabs application # LoginSteps.user_has_opened_the_swaglabs_application()
When User provides usermae "standard_user" # LoginSteps.user_provides_usermae_something(String)
And User provides password "secret_sauce" # LoginSteps.user_provides_password_something(String)
And User clicks on Login button # LoginSteps.user_clicks_on_login_button()
Expected : PRODUCTS
Actual : PRODUCTS
Then User should be logged in successfully # LoginSteps.user_should_be_logged_in_successfully()

@Product
Scenario: Verify that user is able to multiple single product # src/test/java/features/Product.feature:17
/**[id="add-to-cart-sauce-labs-bike-light"]
/**[id="add-to-cart-sauce-labs-fleece-jacket"]
When User selects products to add # ProductAdd.user_selects_products_to_add(DataTable)
Item must be added successfully # ProductAdd.items_must_be_added_successfully()
Then Items must be added successfully
```

Conclusion

Cucumber is an Behaviour Driven Development. With the simple English like language it is easy to build the feature well ahead of development. Jenkins is useful to have the builds along with multiple options like mvn test, git code and reports. I have learnt to incorporate the learnings of Phase 1 like Java code, XPath, Web driver along with the mvn and Cucumber.