Workspace Set Up

May 9, 2016

Table of Contents

[Dictionary of Terms 3](#_Toc450564648)

[Introduction 4](#_Toc450564649)

[Tools and Components 4](#_Toc450564650)

[JDK1.8 (Ignore if already configured) 4](#_Toc450564651)

[Apache Maven 3 (Ignore if already configured) 4](#_Toc450564652)

[IntelliJ (Ignore if already configured) 5](#_Toc450564653)

[Selenium2 5](#_Toc450564654)

[Cucumber 5](#_Toc450564655)

[Serenity (Formerly Thucydides) 5](#_Toc450564656)

[ChromeDriver 5](#_Toc450564657)

[Firefox browser, Firebug, Firepath 5](#_Toc450564658)

[Final Configuration & Execution: 5](#_Toc450564659)

[Execution Through Jenkins: 7](#_Toc450564660)

[Serenity Reports 7](#_Toc450564661)

[Technology Stack 8](#_Toc450564662)

[Folder Structure 8](#_Toc450564663)

[Serenity.Properties file 9](#_Toc450564664)

[Mapping 9](#_Toc450564665)

# 

# Dictionary of Terms

|  |  |
| --- | --- |
|  |  |
| GitHub Project | https://github.com/YamStranger/serenity-demos/tree/master/cucumber-webtests |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

The objective of this document is to describe the workspace of test automation and how to set it up in a develop environment.

# Tools and Components

To set up your workspace for developing and implementing test cases in proposed POC, you need following:

* JDK 1.8
* [Apache Maven 3](#Maven)
* [Eclipse](#_Eclipse) as IDE
* [Selenium2](#_Selenium2)
* [Cucumber](#JBehave), as the methodology used in testing
* [Serenity](#_Thucydides) for reporting
* [Chromedriver](#_ChromeDriver)
* [Firefox browser, Firebug, Firepath](#_Firefox_browser,_Firebug,)

## JDK1.8 (Ignore if already configured)

Java is needed for this POC.

You can download JDK7 from [Java SE Development Kit 7u25](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html). (Or the latest version)

JDK is used in both development machine and JVM.

Create an environment variable “JAVA\_HOME”. This environment variable should have the path to the maven.

java version "1.8.0\_60"

## Apache Maven 3 (Ignore if already configured)

We use maven for building and executing test cases in POC. To install maven,

* Download the binary or zip file from [here](http://maven.apache.org/download.cgi)
* Install it.
* Create an environment variable called “M2\_HOME”. This environment variable should have the path to the maven.
* Configure the system path to include maven
* Open a terminal and enter *mvn –v*. It shows the version of maven, if it is installed and configured properly.

Following screenshots illustrates above steps.

Apache Maven 3.0.3 (r1075438; 2011-03-01 04:31:09+1100)

Maven home: C:\Program Files (x86)\maven

Java version: 1.8.0\_60, vendor: Oracle Corporation

## IntelliJ (Ignore if already configured)

Download and install the right version of eclipse from <http://www.eclipse.org/downloads/>.

Install maven plugin “m2eclipse”, if it is not installed. maven is used for building and executing Test Automation program. You can get m2eclipse from [here](http://eclipse.org/m2e/download/).

Or IntelliJ community Edition

**Plugins**:

Maven Integration

Cucumber for Java

Gherkin

## Selenium2

Selenium2 (WebDriver) is the main component for testing websites in the automated test system. There is no need to download Selenium2 as it will be downloaded by Thucydides and Jbehave plugins.

## Cucumber

Cucumber JVM is the test framework that supports behaviour driven development in BDPOC project.

These jar files will be downloaded automatically by maven.

## Serenity (Formerly Thucydides)

Thucydides (Thoo-SID-a-dees) is a tool designed to make writing automated acceptance and regression tests easier. It provides features that make it easier to organize and structure your acceptance tests, associating them with the user stories or features that they test. As the tests are executed, Thucydides generates illustrated documentation describing how the application is used based on the stories described by the tests.

All necessary jar files will be downloaded by POM.xml.

## ChromeDriver

You need chromedriver in your path file, when you want to test using chrome browser. Get the latest version of [chromedriver](http://code.google.com/p/chromedriver/downloads/list) and put it in C:\Windows\System32.

## Firefox browser, Firebug, Firepath

Install [Mozilla Firefox](http://www.mozilla.org/en-US/firefox/new/) then install Firebug and Firepath Add-ons.

## Final Configuration & Execution:

If this is the first time, start from here:

Extract the project from the Zip files and place it any location.

Change the extensions of run.txt and run\_Whisper.txt to .bat

Place the “settings.xml” file in C:\Users\<YourUserName>\.m2

Open the settings.xml and edit proxy information. Leave it if proxy is not required.

Open command prompt and go to the project location.

Execute the following command:

mvn clean install –Dwebdriver.driver=chrome –Dwebdriver.base.url=”http://au.whispir.com”

This will download all the requisite jars. If the build fails, resolve the issues and re-run the command until you get BUILD SUCCESS

If this is not the first time then start from here:

Open AcceptanceTestSuite in project folder and add the tags of the features that you want to run.

Open run\_Whisper.bat file and:

1. Set execution platform: (1 for execution and 0 for ignoring)

SET DESKTOP\_WHISPIR=1

SET BROWSERSTACK\_WHISPIR=0

1. Set Execution profile to SINGLETHREAD or MULTITHREAD

SET PROFILE=%SINGLETHREAD%

Note: Multithreaded execution is unstable when same steps are being used across different browser instances

1. Enter/change Base URL
2. Browser if required

For BrowserStack execution,

1. Set Proxy configuration (if required)

SET PROXY\_SET=true

SET PROXY\_HOST=

SET PROXY\_PORT=

SET PROXY\_USER=

SET PROXY\_PASSWORD=

1. Provide execution environment:

SET USERNAME=

SET AUTOMATE\_KEY=

SET BROWSERSTACK\_OS=

SET BROWSERSTACK\_OS\_VERSION=

SET BROWSER=

SET BROWSER\_VERSION=

Open command prompt, go to the location where the project is kept, run the run\_Whisper.bat file and execution will start.

## Execution Through Jenkins:

* Create a new job, give any name for the project, Select "Freestyle project" among the project types and hit ok
* You should be in the configuration page. Click Advanced button.
* Check "Select Custom Workspace" and provide the location of Cucumber-Serenity-POC in your system. (We can also configure to read the project from an SCM tool)
* Click "Add Build step" and select "Execute windows batch command"
* Enter the name of the batch file in the command section "run\_Whisper.bat" and click Apply/Save

Click on "Build Now" to start the execution.

## Serenity Reports

Serenity reports will be found in %PROJECT\_LOCATION%\target\site\serenity\index.html

# Technology Stack

* Serenity – Reporting and Requirements Management
  + Serenity Core
  + Serenity Junit
  + Serenity-rest-assured
  + Serenity-cucumber
* Maven – Dependency and build management
* Cucumber – BDD
* Bintray repositories for serenity maven plugins
* SLF4J - The Simple Logging Facade for Java (SLF4J) serves as a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at deployment time.
* Groovy - A powerful, dynamic language for the JVM
* JUnit – Unit testing framework
* Lambdaj - main purpose of lambdaj is to partially eliminate the burden to write (often nested and poorly readable) loops while iterating over collections
* AssertJ - provides a rich and intuitive set of strongly-typed assertions to use for unit testing (either with JUnit or TestNG).

# Folder Structure

Src

* Test

- java

- package

- JunitTest folder

- JunitTest.java

- Model folder

- getter setters , enumerations etc

- Pages

- Page object files

- steps

- Step definition files / per feature

- cucumber runner files (Test Suites)

- resources

- features

- featurefolder

- .feature file

# Serenity.Properties file

#firefox, chrome, phantomjs or iexplorer  
webdriver.driver=chrome  
  
#webdriver.base.url  
serenity.project.name = LinkedIn Project using Serenity and Cucumber  
  
serenity.browser,height = 1200  
serenity.browser,width = 1200  
# Run through the steps without actually executing them.  
serenity.dry.run=false  
# Set this to run all web tests in a single browser  
serenity.use.unique.browser = false  
  
#serenity.test.root=net.thucydides.showcase.cucumber  
serenity.requirement.types = epic,feature  
#tag.failures = "true"  
#linked.tags = "issue"  
#logging = "NORMAL"  
serenity.requirements.dir = src/test/resources/features  
  
# How long does Serenity wait for elements that are not present on the screen to load  
webdriver.timeouts.implicitlywait = 5000  
  
#FOR\_EACH\_ACTION, BEFORE\_AND\_AFTER\_EACH\_STEP, FOR\_FAILURES  
serenity.take.screenshots=AFTER\_EACH\_STEP  
serenity.report.show.manual.tests=false  
# Keep the Thucydides session data between tests. Normally, the session data is cleared between tests.  
serenity.maintain.session=true  
serenity.store.html.source=true

**Complete list of serenity properties:**

<http://go-gaga-over-testing.blogspot.com.au/2015/11/complete-list-of-serenity-properties.html>

# Mapping

Feature folder = Epics

Feature = Feature

Scenario = Acceptance Tests

Feature File

Steps - Step Definition (Given/When/Then)

Steps=>Serenity – UserSteps

Step Declarations @Step

Pages – Page Objects

Custom methods for Page

# Run Tests

To run without deleting previous results:  
mvn verify  
  
mvn clean verify  
  
By default, the tests run with PhantomJS, so you will need this installed. Otherwise, if you prefer Firefox, modify the serenity.properties file or run the tests like this:  
```  
mvn clean verify -Dwebdriver.driver=firefox  
```  
  
The reports will be generated in ***`target/site/serenity`***.

**To run only individual feature, run a specific tag say @wip**

Work In Progress tag @wip

mvn verify -Dcucumber.options="--tags @wip"

mvn verify -Dcucumber.options="--tags @wip @debug2"

More -DCucumber.options here: <http://cucumber.github.io/api/cucumber/jvm/javadoc/cucumber/api/CucumberOptions.html#tags-->

**To run only individual feature, run a specific runner class**

@RunWith(CucumberWithSerenity.class)  
@CucumberOptions(features = "src/test/resources/features/search/search\_by\_keyword.feature")  
public class SearchByKeyword {  
}