# UserManual – SQS BDD Cucumber Framework

Contents

[UserManual – SQS BDD Cucumber Framework 1](#_Toc16006676)

[Cucumber BDD Framework 1](#_Toc16006677)

[Languages 1](#_Toc16006678)

[Tools 2](#_Toc16006679)

[Setup & Installation 2](#_Toc16006680)

[Required Softwares 2](#_Toc16006681)

[Framework Setup 2](#_Toc16006682)

[Configuration 2](#_Toc16006683)

[Test Execution 3](#_Toc16006684)

[Maven TestRunner: 3](#_Toc16006685)

[Excel Execution: 3](#_Toc16006686)

[Framework Structure 3](#_Toc16006687)

[ *Config* 3](#_Toc16006688)

[ *Features* 3](#_Toc16006689)

[ *TestData* 4](#_Toc16006690)

[ *PageObjects* 4](#_Toc16006691)

[ *Result* 4](#_Toc16006692)

[ScreenPrint References: 4](#_Toc16006693)

# Cucumber BDD Framework

Cucumber based Behavior Driven Development framework intended to provide tool & technology agnostic test automation solutions

### Languages

* Java
* Gherkin
* Bash
* yml
* json

### Tools

* Selenium – Web Application Automation
* Appium – Mobile application automation
* API – REST/ API Services automation

## Setup & Installation

### Required Softwares

* Java JDK – Install JDK & JRE and add JAVA\_HOME JDK path in system variable
* Maven - Install latest and set environment variable – M2\_HOME, M2, MAVEN\_OPTS and add M2 to System Path variable.
* Git – Install Git bash & Git GUI
* Eclipse IDE– Install or package the latest IDE and install the following plugins
  + Cucumber Eclipse Plugin
    - <https://cucumber.io/cucumber-eclipse/update-site>
  + Natural – Cucumber Syntax highlighter
    - <http://rlogiacco.github.com/Natural>
  + ADT Plugin – Android Development Tools , AVD Manager, SDK Manager
  + M2e  – Enable Maven Build features
  + JunitHelper – Junit Test Runner
  + TestNg – TestNg Runner Helper
  + Git – GIT Version Control System
* Appium - Install Appium Desktop , npm & nodeJS
* Android SDK Configured – Install the Android Studio/ ADT bundler, AVD manager, SDK Managers and following in the environment variables
  + ANDROID\_HOME, platform-tools, tools

### Framework Setup

* Get Access to the SQS Azure Devops and Clone the project into your local workspace
  + URL - <https://sqsglobal.visualstudio.com/BDD%20Framework/_git/Cucumber_BDD>
* Import the project in the Eclipse/Intellij IDE
* Reload the maven modules for the project
* Build the project and check for the successful compilation

## Configuration

1. Update the project name in Config/MainWeb.properties
   1. current.project.name=<projectName>
2. Create project specific properties in Config folder
   1. <projectName>.properties and fill in the basic details
   2. **excelreport.file**=**.\\Result\\** <**projectName>.xlsx**

*#Feature File location path =src/test/resources/Features*

* 1. **feature.path**=**src/test/resources/Features**

*#Object repository path=src/main/resources/PageObjects*

* 1. **object.repository**=**PageObjects***# Application Type (eg: nativeapp,hybridapp or MobileWeb or DeskTopWeb)*
  2. **application.type**=**DeskTopWeb***#application.URL*
  3. **application.url**=**https://demo.cyclos.org***# Browser name (eg: chrome , Firefox ,IE )*
  4. **desktop.browser.name**=**chrome**

1. Device Specification : In Order to do mobile automation, the list mobile devices or emulators details need to be given in Config/Devices.csvfile

## Test Execution

Test execution from SQS BDD framework can be done in two ways.

* Excel based Run Plan execution – regression tests, where multiple devices coverages across multiple applications (web/mobile) need to be executed
* Maven Test Runner – Features inputs that are given in runner file will be executed

### Maven TestRunner:

Input the feature details in TestRunner.java file and perform mvn test action

* 1. Scenarios specified in testrunner file will be executed ,
  2. Application specific run time configuration details will be picked from “<projectName>.property ” file

### Excel Execution:

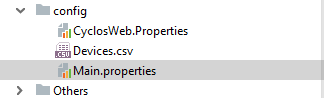
* Excell based execution , picks the run plan details and scripts will be executed in seperatly for each device coverage specified.
* To perform parallel execution on specified ‘N’ number of device configuration, trigger the sqsdriver run time time class for ‘N’ times device coverage specified, multiple

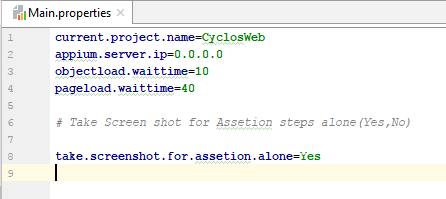
## Framework Structure

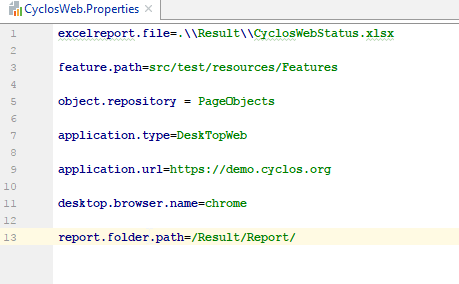
* *Config*: Configuration files are placed in “Config” folder. Any new projects implementation files are to be placed here
* *Features*: Feature file are to be placed in /src/test/resources/Features folder
  + Feature files can be placed in subfolders based on requirement
* *TestData*: TestData for the framework can be either given directly in features files via cucumber examples/ scenario data tables or from external files
  + JSON based test data parsing is handled.
  + All TestData related files are places in /src/test/resources/TestData folder
  + Tagging can be done, in order to pass specific test data file for a specific scenario,
    - For Example @data=FundTransfer
    - This will fetch the “FundTranser.json ” file contents from TestData folder to the specified scenario
    - TestData variables in scenarios can be either $ dollar appended or can be within quotes.
      * G**iven** As an user '**$MyUserID**' with credential '**MyPassword**'
      * During execution, any arguments that are either dollar appended or standalone, will be considered as test data variables and equivalent test data will be fetched from TestData folder
* *PageObjects*: Page object model is implemented and objects are json files
  + By Default the page objects are placed in “/src/main/resources/PageObjects” folder
* *Result*: Test execution reports and screenprints will be available here
  + Upon execution , a sub folder will be created based on project and run times details
    - For Example <projectName>\_<applicationType>\_<CurrentData\_Time>
    - Folder path /Result/Reports/CustomName
  + Extent Cucumber & Allure reports will be available on completion or execution

## ScreenPrint References:

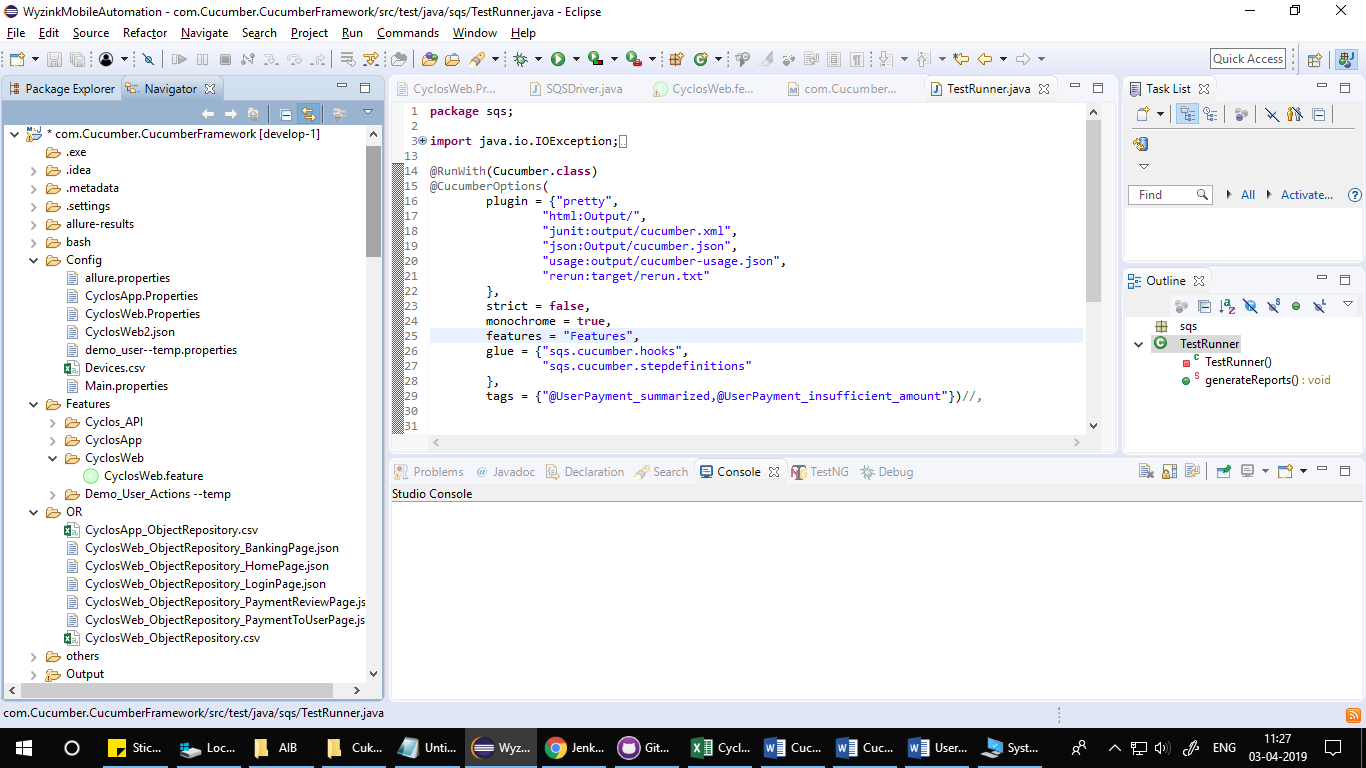
1. Config



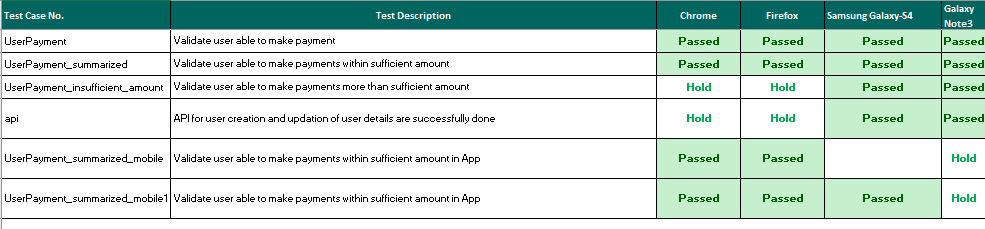




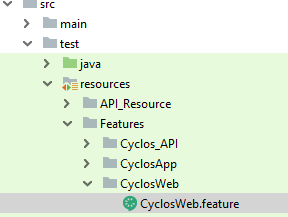
1. Execution - Maven TestRunner

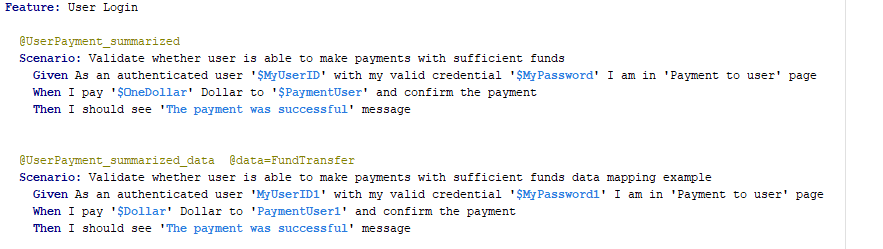


1. Execution - Excel Test Plan

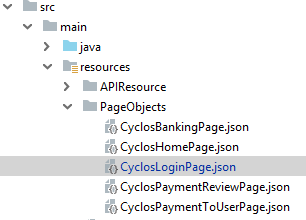


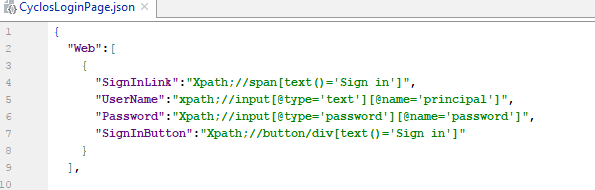
1. Features



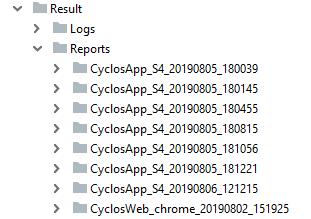


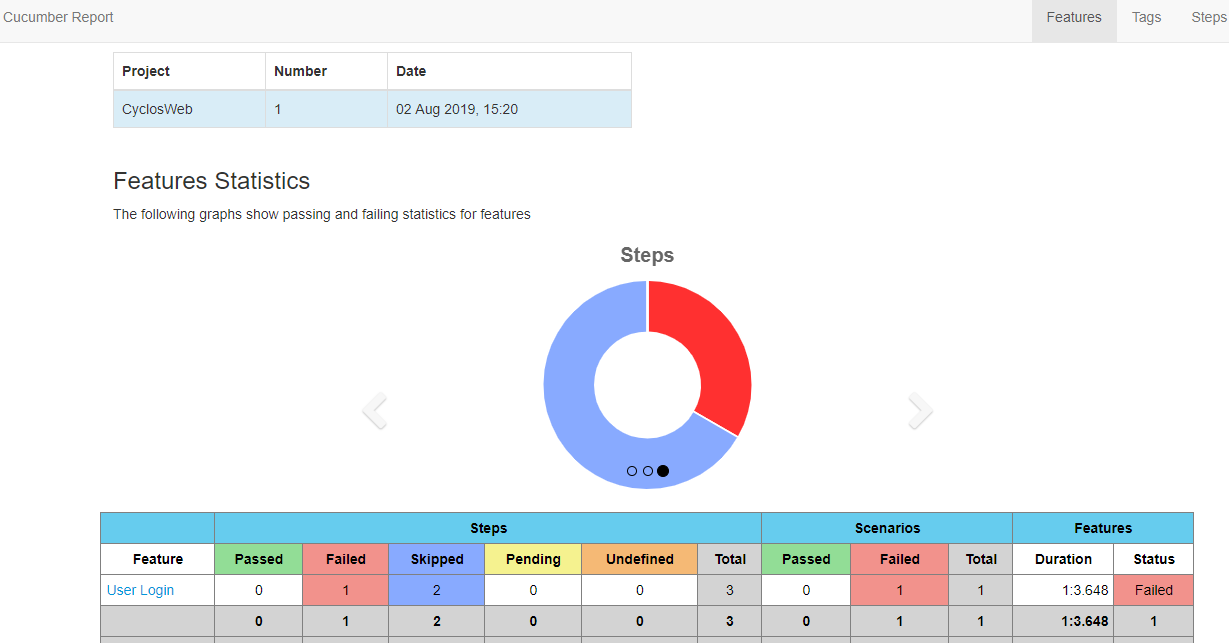
1. PageObjects:





1. Reports:





1. TestData:

