**Data Driven Framework Using Appium**

Table of Contents

[1. Introduction 4](#_Toc488065928)

[1.1 Purpose 4](#_Toc488065929)

[2. Tools Used 4](#_Toc488065930)

[3. References 4](#_Toc488065931)

[4. Dependencies 4](#_Toc488065932)

[5. Procedure 4](#_Toc488065933)

[5.1 Setting up environment for creating Maven Project in Eclipse Neon: 4](#_Toc488065934)

[5.2 Creating Project 4](#_Toc488065935)

[5.3 Upload to Xamarin Test Cloud 7](#_Toc488065936)

[6. Troubleshoot 9](#_Toc488065937)

[6.1 Problem 9](#_Toc488065938)

[6.2 Solution 10](#_Toc488065939)

[6.3 Reference 10](#_Toc488065940)

# Introduction

## Purpose

Automate application with Appium in Eclipse using Data Driven Framework.

# Tools Used

* Eclipse Neon
* Appium (v1.4.16)

# Dependencies

* Java
* Maven
* Selenium
* TestNG
* Appium

# Procedure

## Setting up environment for creating Maven Project in Eclipse Neon:

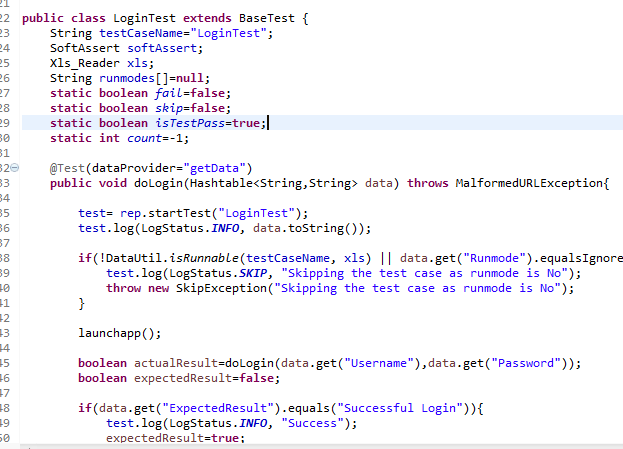
* Download Java 8 version and set environment variable for the same.
* Download [Apache Maven 3.5.0](https://maven.apache.org/download.cgi) (Binary tar.gz archive) and set environment variable in system for the same.
* Download and install Eclipse Neon IDE for creating project.
* Download and install Appium 1.4.16.
* Download Selenium Client & WebDriver Language Bindings from [seleniumhq.org](http://www.seleniumhq.org/download/) site.

## Creating Project

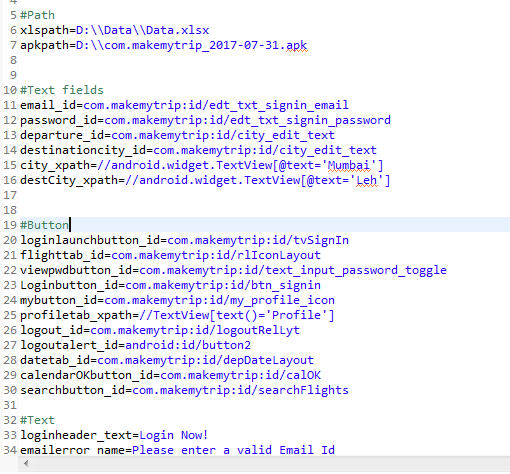
* Create Maven project in Eclipse Neon.
* Open command prompt-> Go to workspace path.
* Enter mvn archetype:generate command
* Click Enter for the version number
* Enter artifact id and group id.
* Click on enter twice.
* Type Y for Y/N
* The project will be created in the workspace folder which was given in the path.
* Now go to project path in command prompt.
* Enter mvn eclipse:eclipse
* Import the project in Eclipse.
* Add jar dependencies of excel and other as mentioned below in POM file.
* JAR dependencies 🡪 poi, poi-ooxml, poi-ooxml-schemas, xml-apis, dom4j,xmlbeans,selenium-java, testing, java-client,gson, httpclient, guava, cglib, commons-validator, javax.mail, extent reports
* In command prompt, enter the project path and type mvn compile
* All the jars will be downloaded and will take a bit longer to compile.
* Once the project is compiled, enter mvn eclipse:eclipse.
* Now in Eclipse, refresh the project.
* Verify that all the jars are displayed in the referenced libraries.
* Create test java files in the src/test/java folder.
* Create util package with XLS Reader, DataUtil and Extent Manager class.
* Create base and test package in src/test/java folder.
* Add the BaseTest class in base package with functions that are used commonly.
* Add AppFunctions class which will contain only the app related functions.



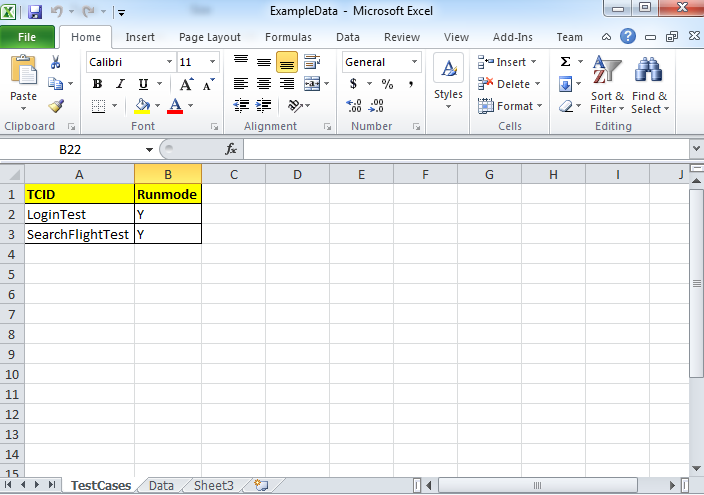
* Add the Test files which will be executed in the test package.

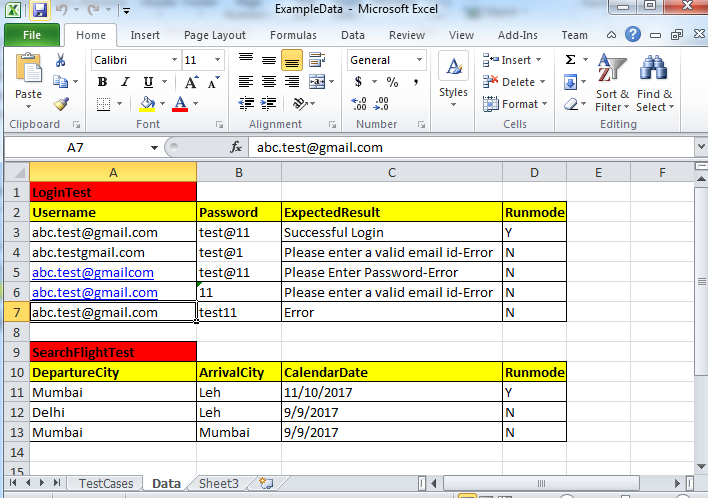


* Create source folder- src/test/resources and create config file in that folder.
* The projectconfig.properties file should contain all the path details.



* Create excel file with test cases and test data details.
* Example :





* After creating above Excel file with TestCases and Data tab, add the excel file path to the projectconfig.properties file.
* Create a testng.xml file. Add the suits and tests to be run through testng.xml.
* For Extent Reports, create and ExtentManager/ExtentReports class and xml file and described in [generate- extent-reports](http://www.softwaretestingmaterial.com/generate-extent-reports/) site.
* Create a Report folder in local machine drive and enter the path in ExtentReports class file.
* The project structure should look similar to below.

