# Propine Assignment – Completion document

## Introduction

This document contains details related to project directory, instructions, environment details and other details as mentioned in assignment document.

## Tools used

Environment IDE – Eclipse

Project type – TestNG based on POM Framework and converted as Hybrid Framework to include excel based input file (we use SVN for version control, haven’t included them, created as a standalone project)

Tools – Selenium, Java, Selenium grid and TestNG

OS – Windows

Browser – Chrome (driver provided in project folder)

## Initialization

The project folder can be copied from google drive link –

<https://drive.google.com/drive/folders/1ES2xBtsn1x5HpU5ZvjTwc_B1JjhVUTHw?usp=sharing>

* There is a propine.zip file uploaded
* Unzip the contents to get 2 main folders – propine and PROPINE\_Assignment
* PROPINE\_Assignment contains all the documents where as propine is the folder which contains the actual project code

There are 3 initializations to be done before running the script

### Step 1

Config.properties file - There are 3 values which are in the properties file which can be configurable.

* Input url
* Excel input file name
* File sheet name

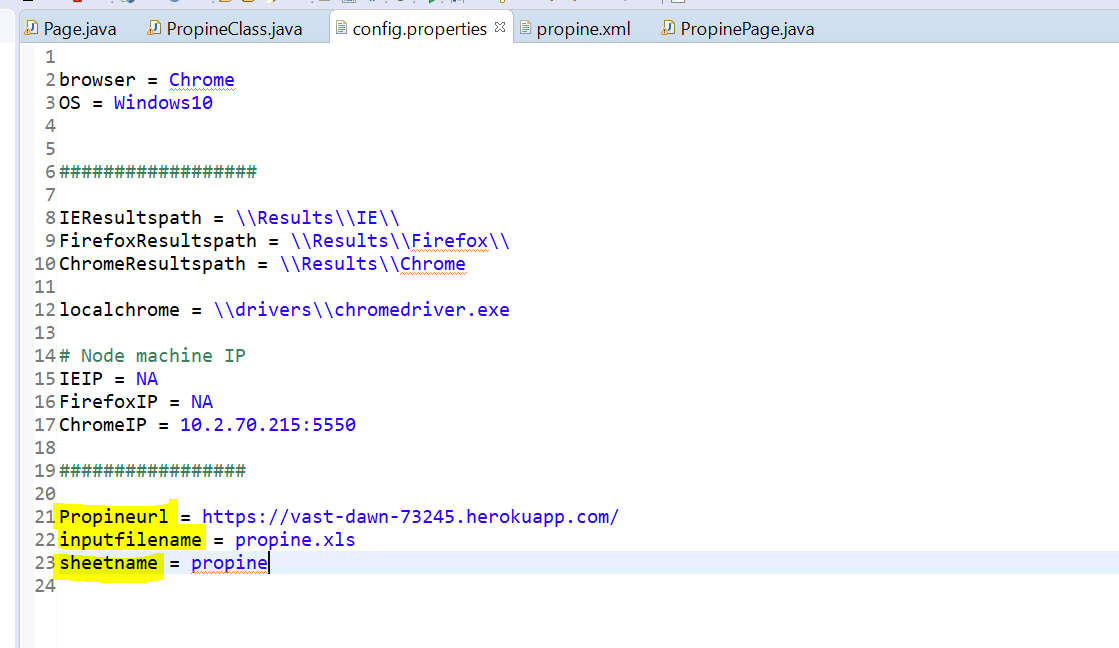


Figure 1: Config. Properties file

### Step 2

After copying the project file to the local folder. Open project in eclipse and configure the build path. All the supporting jar files are placed in the project folder

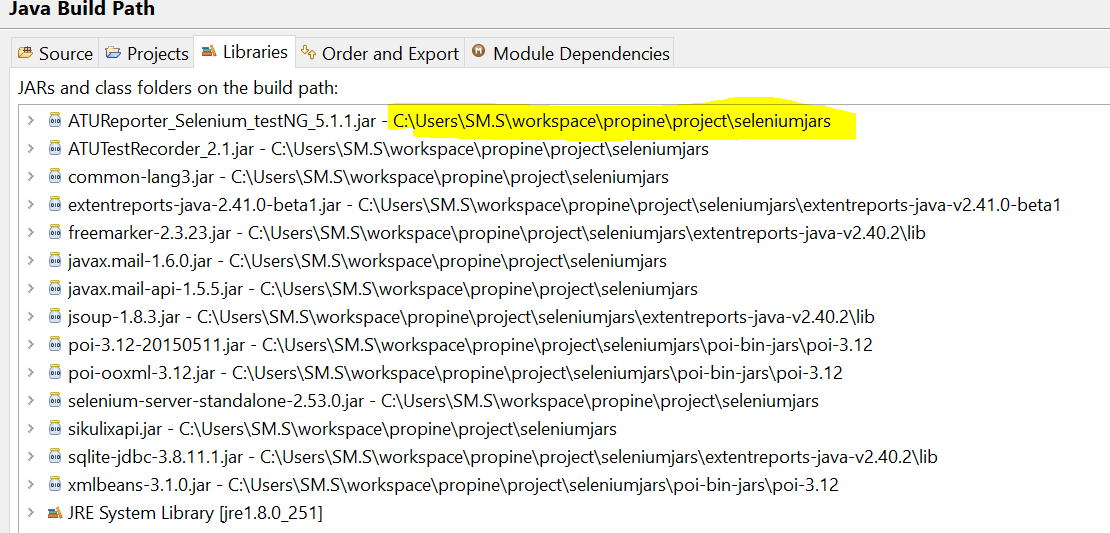


Figure 2: java build path of project, configure it based on local folder. Jar files are in project folder/ seleniumjars

### Step 3

Configuration of Selenium grid. This is based on where we copy the project folder, the selenium server standalone jar is placed in projectfoder/seleniumjars

* Open 2 command prompt windows, 1 for hub and 1 for node (chrome)
* Navigate to projectfolder/ selenium jars in both the command prompts
* In my case it is >C:\Users\SM.S\workspace\propine\project\seleniumjars
* Configure Hub
  + Copy paste in command prompt
  + >java -jar selenium-server-standalone-2.53.0.jar -role hub
  + Press enter to configure
  + This will activate the grid hub, the command prompt 1 will have the current IP of system which needs to be used for nodes or
  + Open another command prompt and verify >ipconfig to get the current ip
* Verify registered grid by opening the grid console
  + <http://localhost:4444/grid/console>
* Configure node
  + Open 2nd command prompt which is in folder project/seleniumjars
  + >java -jar selenium-server-standalone-2.53.0.jar -role webdriver -hub http://192.168.1.3:4444/grid.register -port 5550 -browser browserName="chrome",version=ANY,platform=VISTA,maxInstances=5 -Dwebdriver.chrome.driver=C:\Users\SM.S\workspace\propine\project\drivers\chromedriver.exe
  + Press enter to configure
  + Yellow – current IP of system taken from the selenium hub command prompt or from IP config
  + Green – current project directory\drivers\chromedriver.exe (latest chrome driver is placed in drivers folder in the project shared)

### Run configuration

Open the project in eclipse and open ‘propine.xml’ in source mode

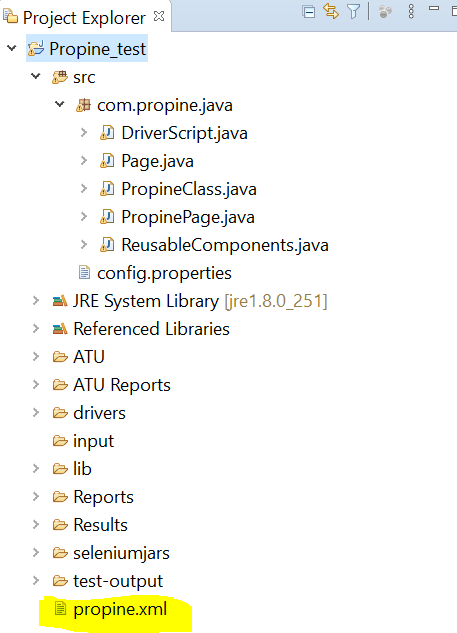


Figure 3: propine.xml

In the xml file we need to configure the node url which we have mentioned in GRID

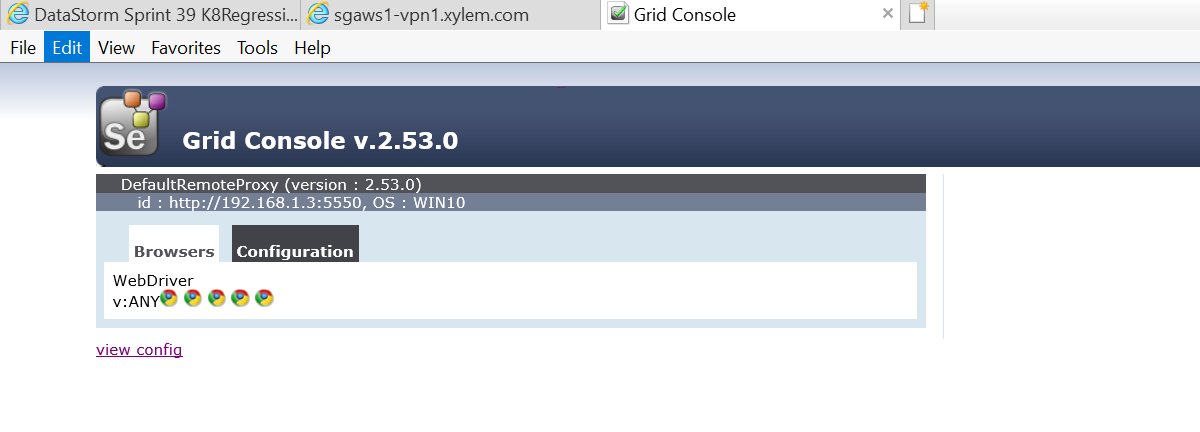


Figure 4: grid console view

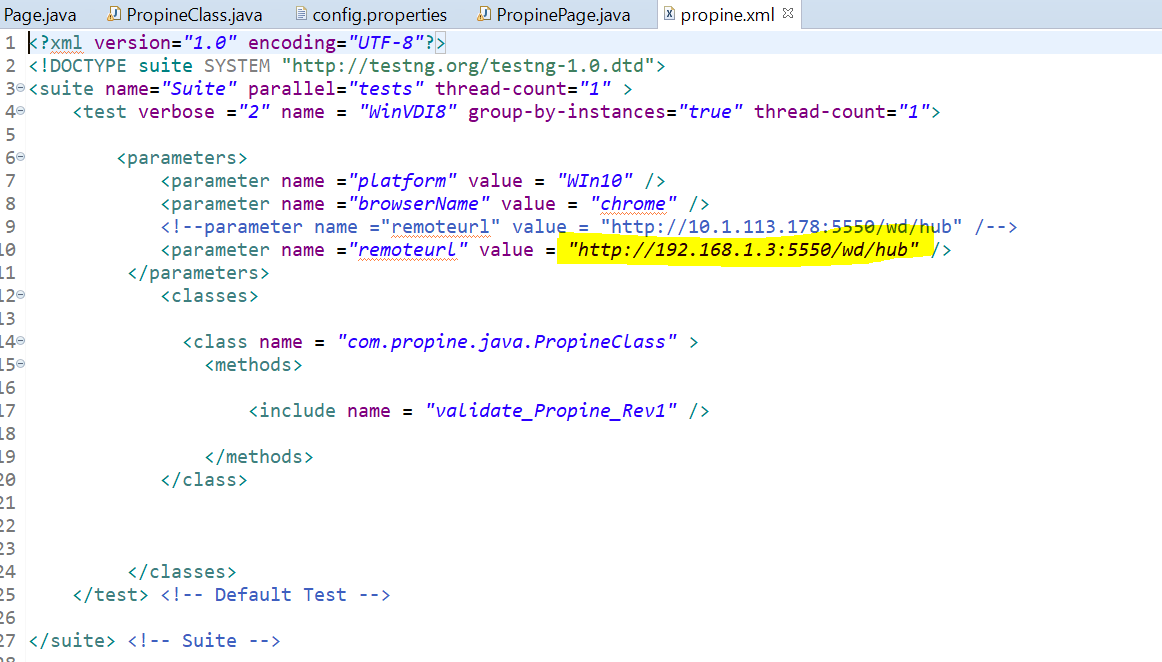


Figure 5: propine.xml view and configuration (change the IP address which is highlighted to the current IP of the system)

* Once updated with current IP address, save the ‘propine.xml’ file
* validate\_Propine\_Rev1

### Input file update

* navigate to project folder/ input to find propine.xls file ( which is mentioned in config.properties file)

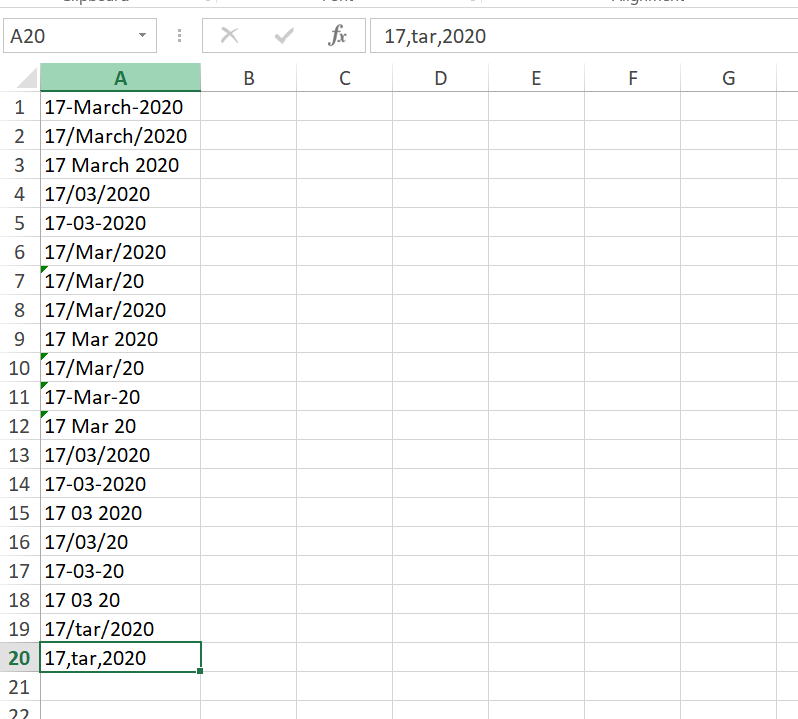


Figure 6: input excel sheet (propine.xls)

### Run

After all the configuration to run the script we have to run the ‘propine.xml’ as testngsuite

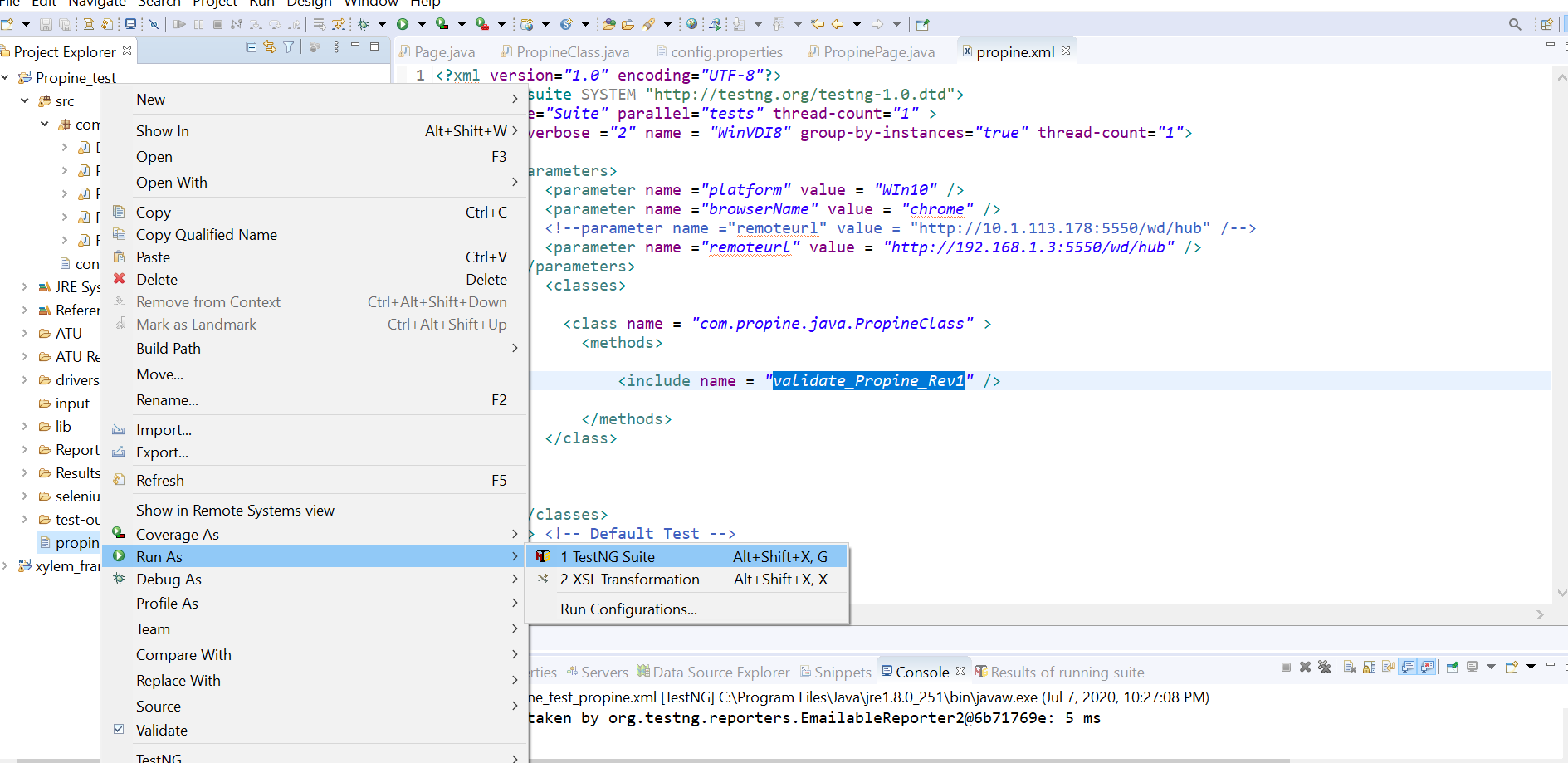


Figure 7: Run window view

### Result and Reporting

We can view the result in the reports folder under project folder (configurable)



Figure 8: DriverScript.java file, where report folder is configured

Sample run results can be found in project/Reports folder

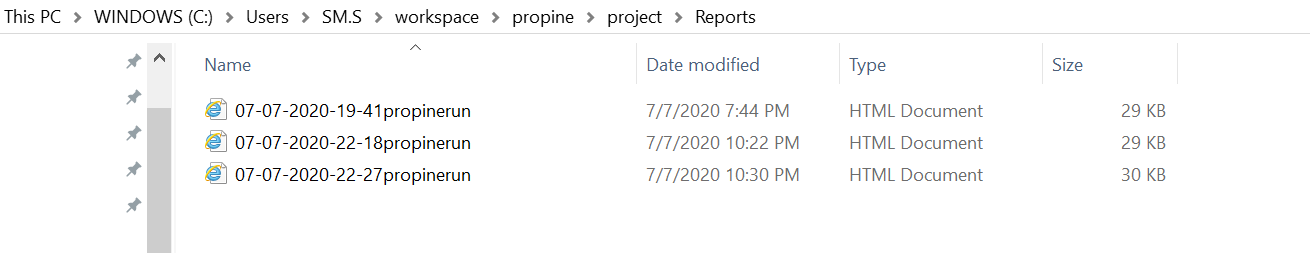


Figure 9: Reports folder

### Assignment Approach and observation

The script have been designed as to take input data from excel sheet and output via extent reporting bases reports which has all the data in them like pass, fail, screenshots etc…

Script approach is simple, the steps mentioned in each of the tests is followed and the observations are noted.

* Page object model (POM) is used with class and page functions in 2 different files.
* Class file consists of all the testng annotations.
* Operations to be reused are mentioned in separate file ‘reusablecomponents.java’.
* Values are not hard coded, instead have been mentioned in config.properties file, which can be changed accordingly
* Screenshots are added at crucial test steps.
* Reports are added with timestamps so that they can be used for further references and not over written.