**Automation Framework – Hybrid for XE Currency Converter**

*Document Ver. 1.0*

*Created By- Pranit Kurdukar (*[*Pranit.kurdukar@gmail.com*](mailto:Pranit.kurdukar@gmail.com)*)*

*01/11/2020*

# Tools and PlugIns

The following tools and plugins are required to use automation framework

1. Eclipse IDE
2. JDK 1.8 or beyond
3. Apache Maven 3.6.3

Setup environment variables after downloading above tools/plugsin.

# Download the framework from GITHUB public repository

Download .zip version of Maven Project from github shared repository to your local machine.

GitHub URL:

<https://github.com/pranitkurdukar/XE-Currency-Converter.git>

## Overview of framework:

1. The following tables describes the framework folder structure and its significant

|  |  |  |
| --- | --- | --- |
| Package/folder | Class | Significance |
| Src/test/java/com.XE.base | TestBase | This class has the setup and tear methods. In setup method, webdriver is initialised for different types of browsers, and tear method is used to close the driver.  In addition to this, objects of other classes are declared in this class. E.g. explicit wait, extent report, log4j etc |
| Src/test/java/ com.XE.bizComponents | XECurrencyConverterPage.class | This class contains action methods specific to web application pages. These methods are invoked in testsuite class. |
| Src/test/java/ com.XE.listeners | CustomListeners.class | Testng listeners methods are implemented in this class. In onTestFailure method, screenshots of failed step will be taken and stored in the folder *\XE-CURRENCY-CONVERTER\XE-CURRENCY-CONVERTER\target\surefire-reports\html\* |
| Src/test/java/ com.XE.testsuites | CurrencyConverter.class | This class contains a single test which is parameterized using testng data provider. This test verifies exchanged amount using multiple currencies. Test Data from testdata sheet is used for this test. |
| Src/test/java/ com.XE.utilities | ExcelReader.class | In this class, reusable methods are created to retrieve test data from excel sheet. e.g. getcelldata, getrowcount, getcolcount etc. These methods are invoked/used in testsuite |
|  | ExtentManager | In this extent report object is created and also configuration of extent report is loaded. This class is used to generate extend report. These reports are generated in the folder *\XE-CURRENCY-CONVERTER\XE-CURRENCY-CONVERTER\target\surefire-reports\html\* |
|  | TestUtil.class | This class contains re-usable methods specific to selenium. E.g. click, type, isElementPresent, getObject etc.. These methods are used in bizComponents\ XECurrencyConverterPage.class |
|  | Log4j.properties | This is property file. It contains the configuration for generating log files for each action. |
| Src/test/resources/ | excel /Testdata.xlsx | It contains test data which is used in testsuite |
|  | executables | In this folder, all web browser driver files are kept. These executables files are called from TestBase class while initialising driver for browses chrome, ie, edge, firefox |
|  | Extentconfig/ReportsConfig.xml | This is configuration file, it contains html code to format the extent repot |
|  | Logs | .log files are generated in this folder. |
|  | Properties | In this folder, two property files are created, OR.property & Config.property.  In OR.property file, all locators are stored.  And URL of application, wait timeout period are stored in the Config.property |
|  | Runners | In this folder, testng.xml files are created for each type of browser. These files are used to run test suite in different browsers |
| /target/surefire-reports/html | Index.html | This folder will be generated after running first test. Test Report is generated in this folder. Refer Index.html file.  Also screenshots of failed test step are stored in same folder |
| /testoutput/html | Index.html | This folder will be generated after running first test. In this folder, testng reports are stored. Refer Index.html |
| pom.xml |  | In pom.xml files, the following dependencies are resolved.   1. SELENIUM-JAVA 2. WEBDRIVERMANAGER 3. extentreports 4. reportng 5. guice 6. log4j 7. poi 8. poi-ooxml 9. xmlbeans 10. dom4j 11. poi-ooxml-schemas 12. maven-surefire-plugin 13. testng |

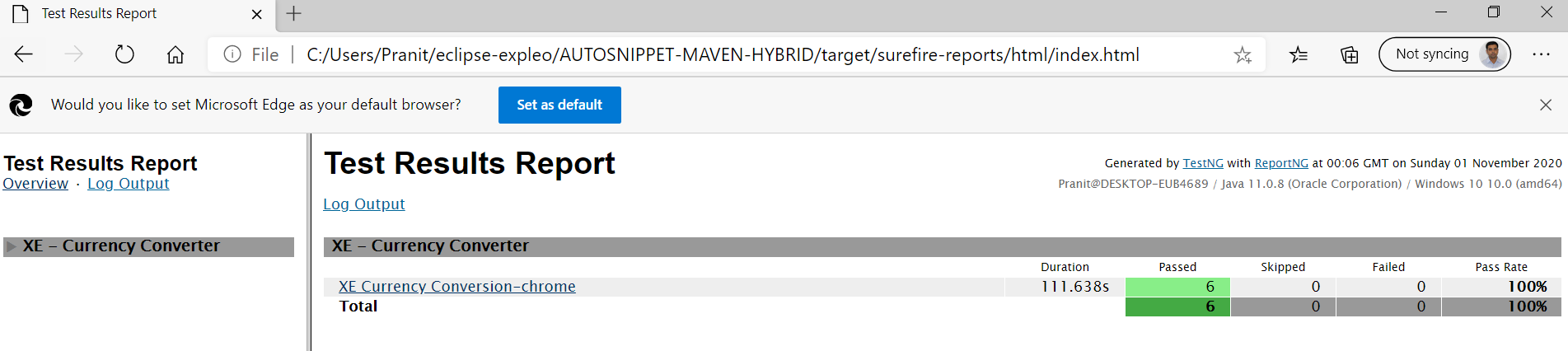
## Run the test suite in Eclipse IDE

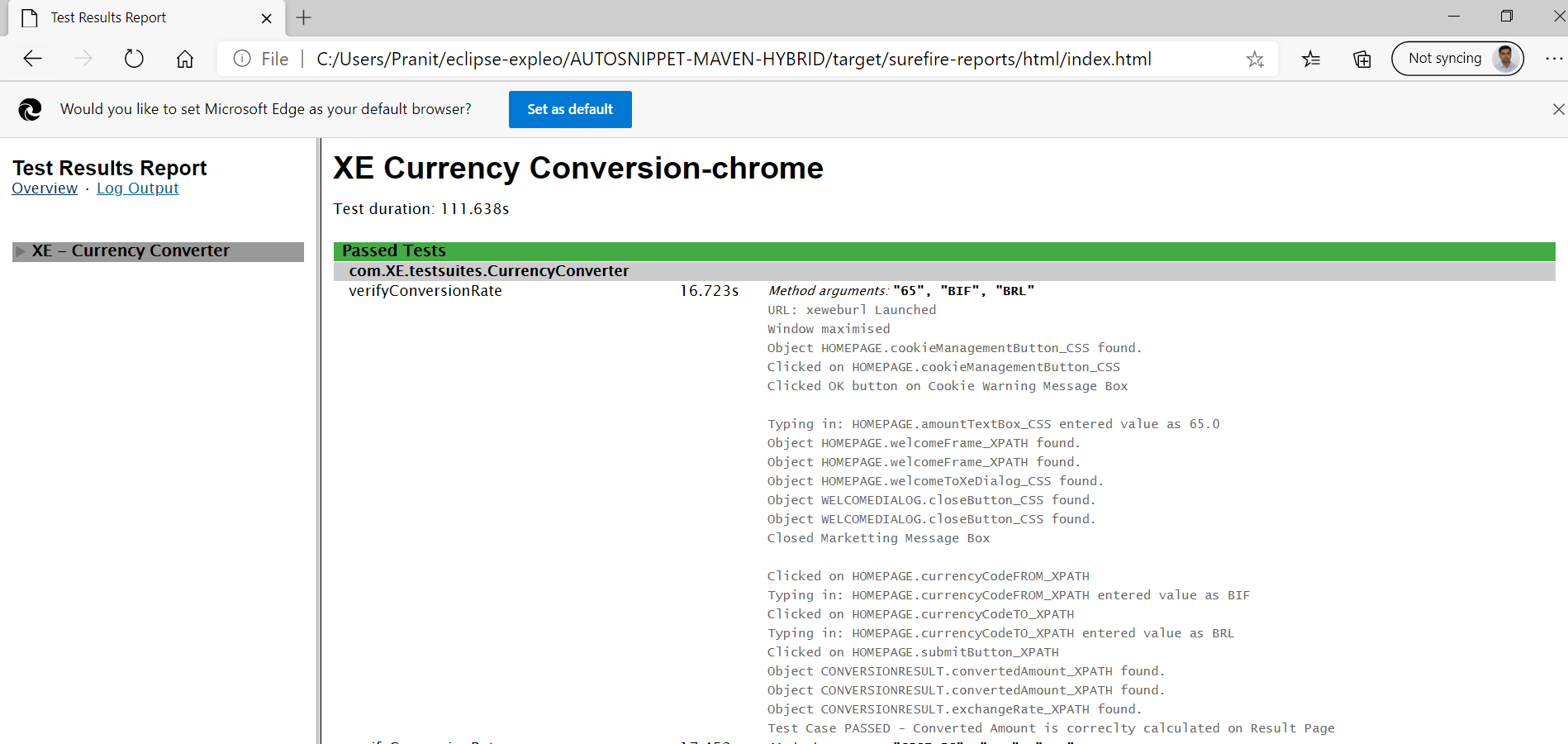
Follow the below steps to run the test case.

1. Right click on Maven Project – *XE-CURRENCY-CONVERTER* and Click Update Project (Alt + F5)
2. After updating Maven Project, right click on project name and Run it as Maven Install to download all dependencies at local machine
3. Update the test data in testdata sheet which is kept in the folder *XE-CURRENCY-CONVERTER\XE-CURRENCY-CONVERTER\src\test\resources\excel\testdata.xlsx.* Make sure type of cell should be TEXT before entering test data in the excel.
4. Save all files
5. Right Click on Maven Project, Run\clean
6. Right Click on Maven Project, Run\test
7. Test will run on chrome browser
8. To run the same test on different browser, uncomment reference of testng xml files in pom.xml.
9. Extent Report will be generated in the folder *\XE-CURRENCY-CONVERTER\XE-CURRENCY-CONVERTER\target\surefire-reports\html\Index.html* . Test Execution Sample Report is given below section.

## Sample Test Report:

\target\surefire-reports\html\index.html





## Run Test Suite on Windows Command Prompt

1. Open windows command prompt
2. Navigate to project folder using command

c:

cd\

cd <<project folder path>>

1. Run the following command, after entering below command, test script will start to run in chrome browser.

mvn clean test -DsuiteXmlFil=src/test/resources/runners/testng.XE\_Chrome.functional.xml

1. Similar command can be used to run the test suite in ie and firefox browser

mvn clean test -DsuiteXmlFil=src/test/resources/runners/testng.XE\_IE.functional.xml

mvn clean test -DsuiteXmlFil=src/test/resources/runners/testng.XE\_Firefox.functional.xml