**Assignment 1: API Automation**

**How to Execute-**

* Download the project named “HelloFreshAssignment-master” from GitHub as zip. Extract the zip file. Kindly note that sometimes after unzipping the file, the “src” folder contents go missing. Hence, make sure that the “src” are intact.
* Import the RestAssured project from local directory
* Right click on “testng.xml”->Run As->TestNG Suite

The main code logic is in “ExecutionClass” class, where 3 methods are created to meet below requirements.

* Get all countries and validate that US, DE and GB were returned in the response – Implementation method = getAllCountries()
* 2. Get each country (US, DE and GB) individually and validate the response – Implementation method = getEachCountryAndValidate()
* 3. Try to get information for inexistent countries and validate the response – Implementation method = getEachCountryAndValidate()
* 4. This API has not a POST method at the moment, but it is being developed. Write a test that would validate new country addition using POST (it will not work now, but no worries).
* Example of json body for POST is below:

  {  
     name: "Test Country",  
     alpha2\_code: "TC",  
     alpha3\_code: "TCY"  
  }

– Implementation method = addNewCountry()

The current output is:



**Assignment 2: UI Automation**

**How to Execute-**

* Download the project from GitHub as zip. Extract the zip file. Kindly note that sometimes after unzipping the file, the “src” folder contents go missing. Hence, make sure that the “src” are intact.
* Import the SeleniumWebDriver project from local directory
* Right click on “testng.xml”->Run As->TestNG Suite

**Description of the project divided in folders and packages-**

1. **Src folder**
2. com.webelements

This package consists of WebElements class which stores all the element locators for the project.

1. common.Config

This package consists of BaseConfiguration class which is used to invoke the properties files (data.properties/config.properties/log4j.properties)

1. common.Resources

This package consists of all the property files (data.properties/config.properties/log4j.properties)

1. common.UIAutomationTest

This package consists of the main java code for all the UI tests. It consists of 3 main classes:

ExecutionClass class: This class is the test class and is invoked for execution by TestNG(testing.xml). It calls test methods along with the before and after methods implemented in WebTestLogic class.

GenerateRandom class: This class is used to generate the email IDs, passwords and mobile numbers used for testing randomly.

WebTestLogic class: This class is the main class consisting of all the code logic. The methods belonging to this class are called by ExecutionClass.

1. **Drivers folder**

This folder can be used to place all the browser drivers required to invoke different browsers for execution. For this project, chrome and firefox drivers have been placed.

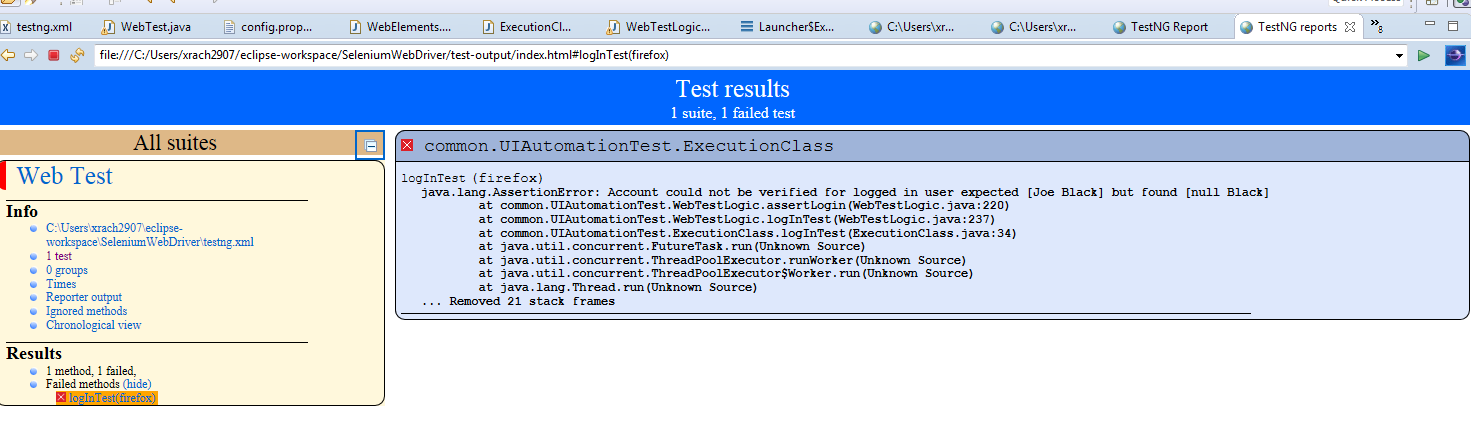
1. **Screenshots folder**

This package consists of all the collection of screenshots post any test/operation failure. Ex. Screenshots will be generated in case of failure of either tests-signUp, login, checkout

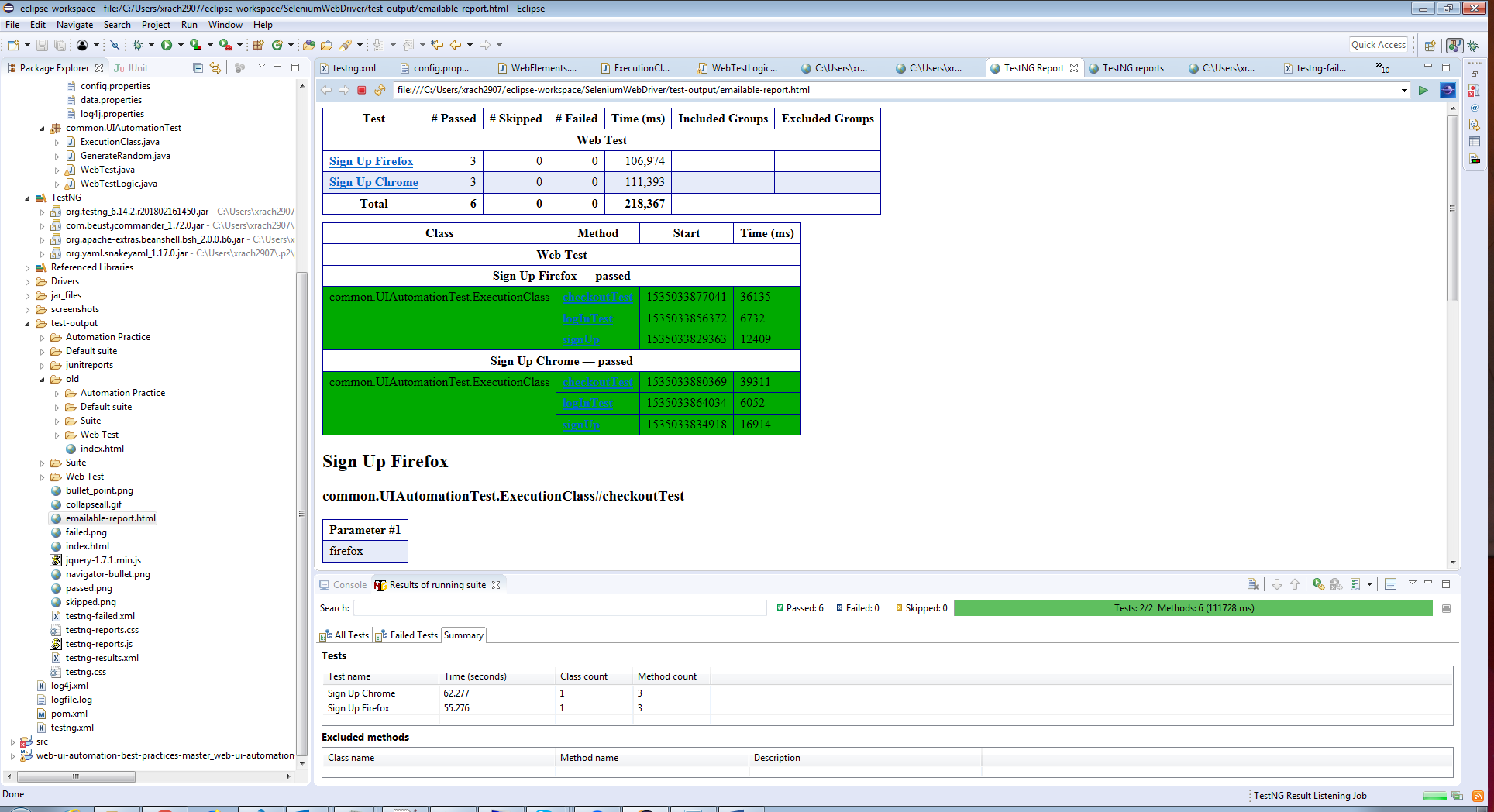
1. **test-output**

This folder can be used to collect the reports generated post execution. It generates 2 html reports – index.html and emailable-report.html

Index.html :



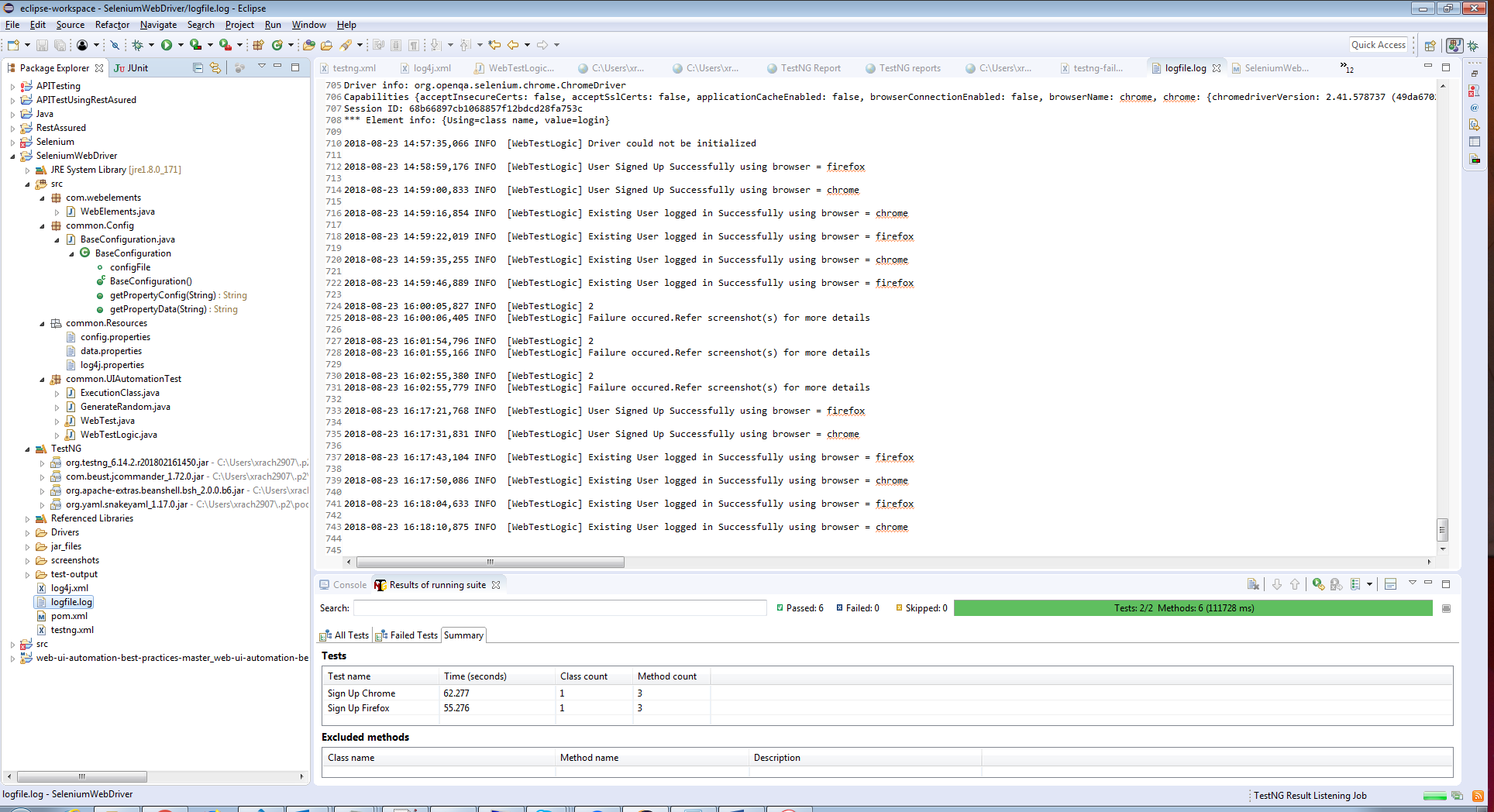
emailable-report.html



1. **.xml, .log files:**

log4j.xml: This consists of all the dependencies required for logging INFO messages into logfile.log

logfile.log: This file gets appended during every execution with INFO messages post test case success/failure

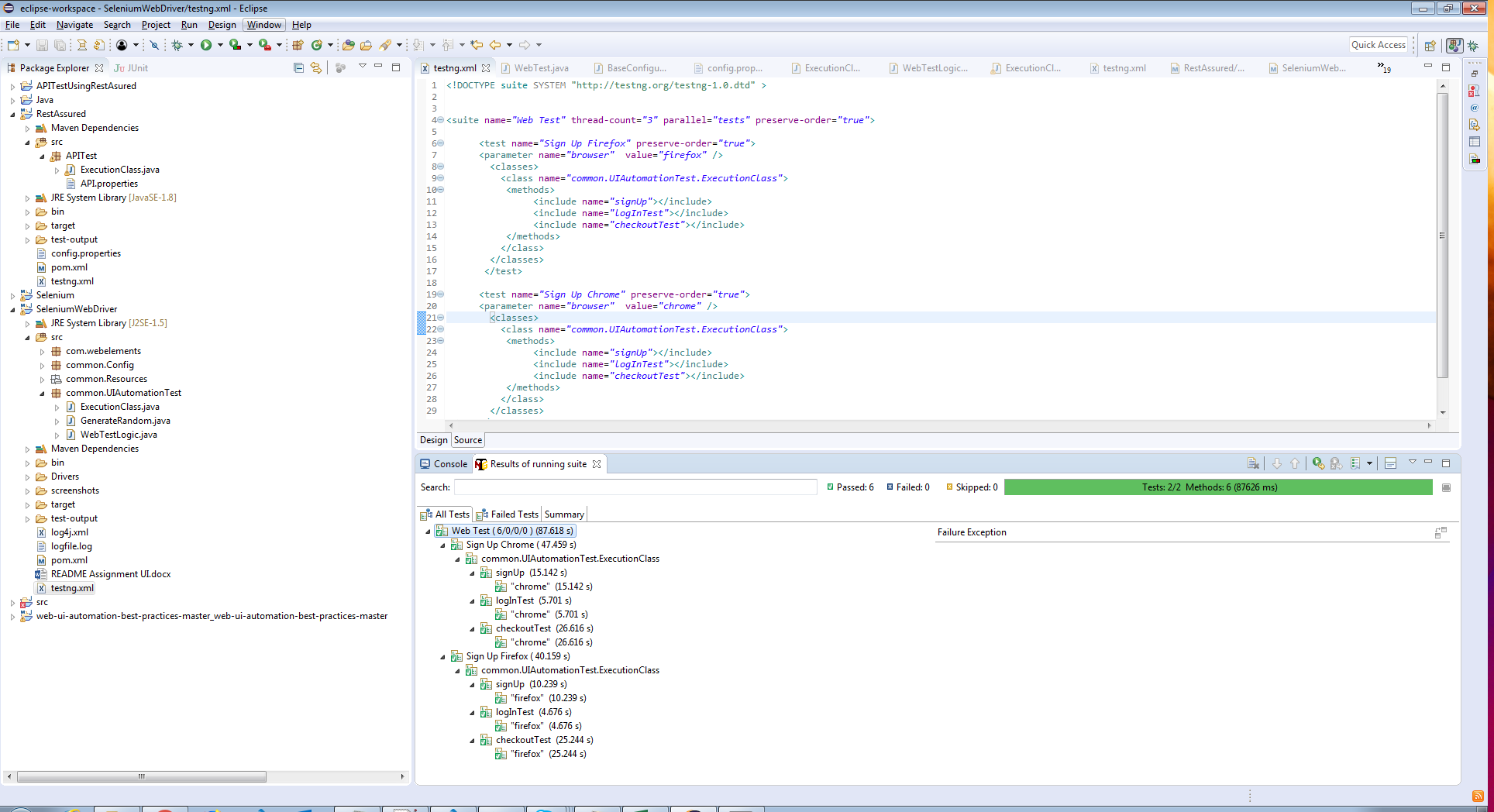


pom.xml: This consists of all the dependencies and versions of different jars used in the project.

testng.xml: This is the xml file that is used for one-click execution. All the execution requirements like browser details, methods to execute, classes to execute, parallel execution/sequential execution etc., can be provided here.



**Current Test Output:**



\*\*\*\*\*\*\*End Of Document\*\*\*\*\*\*