1. **There are Four Packages:**Generic Component – Base. Class, ExcelReadWrite.Class  
   PageObject Component – PageObjectSearch.class, PageObjectCart.class  
   DataProvider Component – DataProviderSearch. Class, DataProviderBuy.class  
   Scenario Component – ScenarioSearch.class, ScenarioBuy.class - (Driver class)
2. **Details of the components:**

DataProvider – To collect the data from excel sheet and pass it to the script

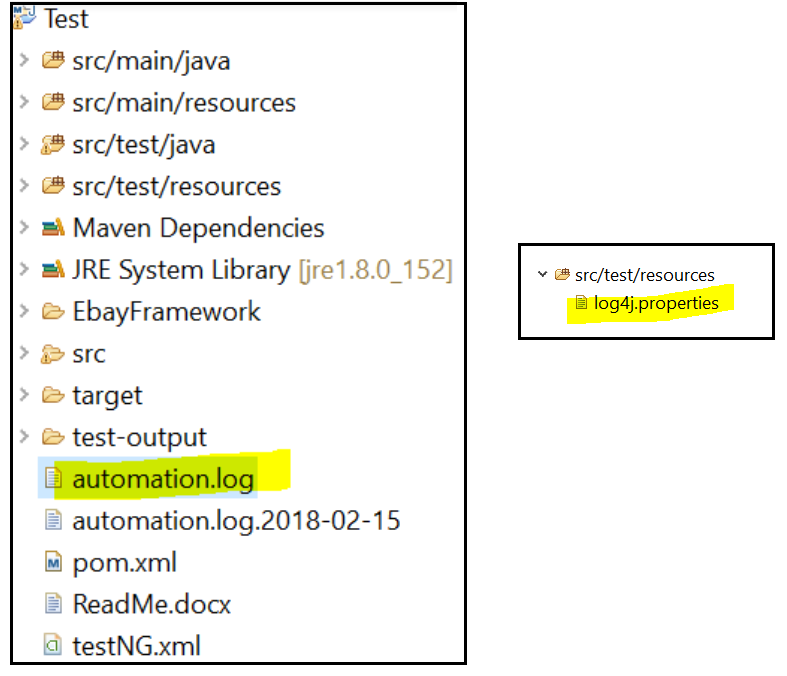
PageObject – To collect all the elements and put them in a method to perform action on those elements.

Scenario – This is the driver class, where the first line of execution takes place.

Generic – The classes in this will be used by other classes. This has the re-usable classes.

1. **Reports:**I have used Log4j report. And You can see the report in “automation.log” file

Also, you can see the Log4j properties under src/test/resource



1. **XPath:** I am using variables for the xpath. For example - The variable is in bold

@FindBy(xpath="//\*[@text='Deliveryaddress'][@index='0'][@class='android.view.View']")

**public** WebElement **deliveryAddress**;

//Another way to write the same code is

WebElement **deliveryAddress** = FindBy(xpath="//\*[@text='Deliveryaddress'][@index='0'][@class='android.view.View']");

1. **Avoid Hardcoding:**

I have just used in rotate() just to showcase that orientation can be handled.If I remove the sleep then it will happen very fast and the user will not be able to see the rotation. Rest of the places I have used Explicit Wait.

Note # for explicit wait I need some element so that I can check if that element is loaded but in such scenarios where the image is already loaded, I wouldn’t be able to use this condition to wait for it to load.

1. **Is the code compiled and executed:**

Yes, its compiled and executed without any errors.

You can see the log4j log which is named as automation.log.

I am attaching Appium logs as well, please have a look at **AppiumConsolelog** folder.

1. **Gestures: Swipe, Zoom, pinch, tap**

**Swipe** – Implemented  
  
**Tap** – It is a simple method therefore didn’t implement it -

driver.tap(int fingers,WebElement el, time);   
  
**Zoom & Pinch** – I have tried on eBay app but it doesn’t show on the screen.

It’s a simple scenario – ebay search textbox> enter search criteria and search for it> click on the desired result item > click on the small image to open the fullscreen > here capture the image and perform zoom & pinch.

But there is a known issue with Android that it either doesn’t find the element or just skips the step of locating the element when it comes to a full screen image and this is not yet fixed for Android for Commercial apps. I have tried with the other E-com apps like Flipkart and BuiltIn app like photo gallery and found the same result (i.e. it was not locating the element).

https://github.com/appium/appium/issues/9491

Then I tried it on an app which is a scale image app specifically to perform zoom and pinch using TouchAction and MultiTouchAction and my same code worked fine over there.

I am attaching my linear script separately called ZoomPinch.java for ebay app. Please have a look.

1. **Assertion:** It is implemented now. Try, catch and finally block added.