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

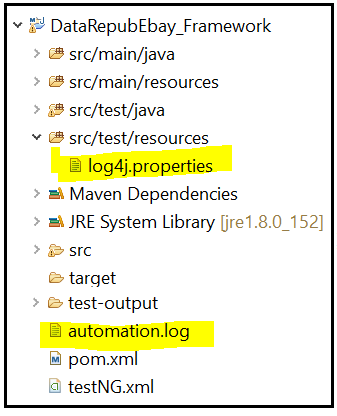
DataProvider – To collect the data from excel sheet and pass it to the script

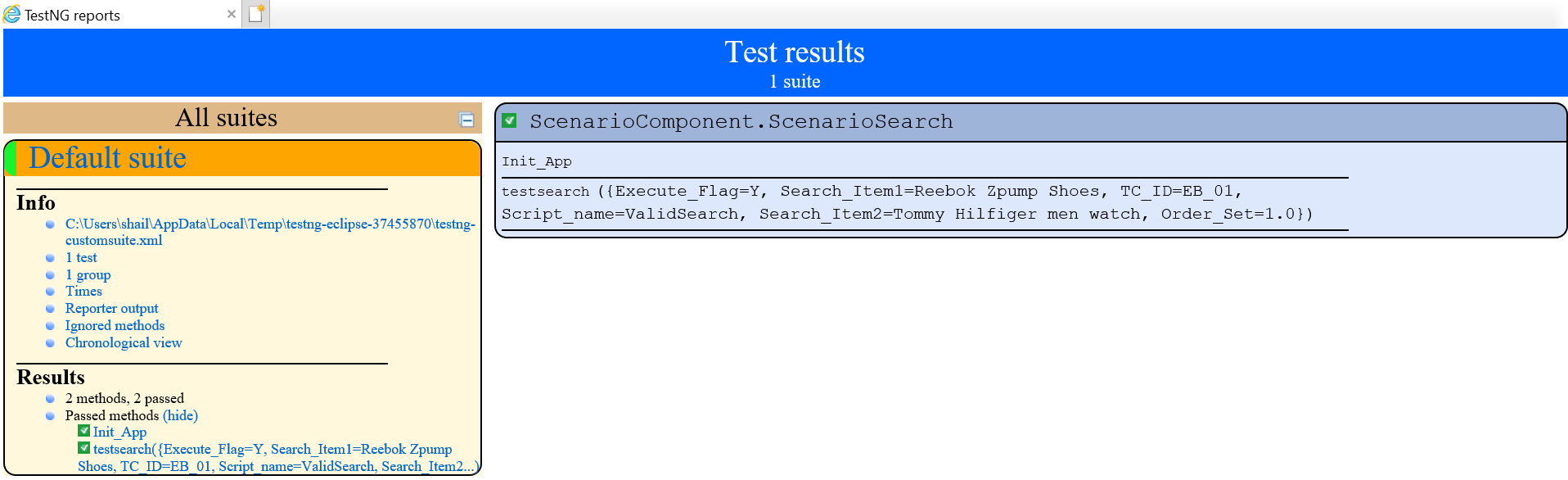
PageObject – To collect all the elements and put them in a method to perform action on those elements.

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:**a. **Log4J 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



b. **TestNG Reports** – Please go to Reports folder inside the folder named DataRepubEbayFrameworkFolder and click on index.html to view the summary of test run.  


1. I have attached the working code for single device. The parallel testing code launches the browser in both the connected device but its running the testcase only on one device and for other device it fails to find the element. I have time crunch to fix this code but I will explain here how I do the parallel testing.  
     
   **A. Below mentioned snippet is the changes required in Init\_App() which is in Base class-** @BeforeTest
2. @Parameters({"deviceID"})
3. public static void Init\_App(String deviceID) throws MalformedURLException, InterruptedException
4. {
6. DesiredCapabilities capabilities= new DesiredCapabilities();
8. //device details
9. //device details
11. if(deviceID.equals("d1"))
12. {
13. System.out.println("Executing in Device1");
15. capabilities.setCapability("deviceName", "Redmi");
16. capabilities.setCapability("platformName", "Android");
17. capabilities.setCapability("platformVersion", "7.0");

20. }
22. else if(deviceID.equals("d2"))
23. {
24. System.out.println("Executing in Device2");
26. capabilities.setCapability("deviceName", "Samsung Galaxy S7 edge");
27. capabilities.setCapability("platformName", "Android");
28. capabilities.setCapability("platformVersion", "6.0.1");
30. }

33. //browser details
34. capabilities.setCapability(CapabilityType.BROWSER\_NAME,"Chrome");
35. //appium server details
36. driver= new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);
38. driver.get("https://www.ebay.com.au/");
39. Thread.sleep(8000);
40. }

**B. In class SearchScenario class, comment or remove the method call Init\_App()**

**C. Get the selenium grid up and running by running this command on command Line  
java -jar selenium-server-standalone-2.52.0.jar -role hub  
Now check on the browser if this URL is up -** [**http://localhost:4444/grid/console/#**](http://localhost:4444/grid/console/)**.**

**D. Register Appium server/ Selenium grid for Device 1 – Run from main.js  
cd \Appium\resources\app\node\_modules\appium\build\lib**

**C: node main.js -a localhost -p 4567 --udid f2366a250803 -bp 5556 --nodeconfig "D:\d1.json" -g "D:\appium\_1.log" --session-override**

**E. Register Appium server for Device 2 – Run from Appium.js  
cd Appium\node\_modules\appium\bin**

**C :**

**node appium.js -a localhost -p 4568 --udid ce06160633c8e60701 -bp 5557 --nodeconfig "D:\d2.json" -g "D:\appium\_2.log" --session-override**

**F. Refresh URL on browser -** [**http://localhost:4444/grid/console/#**](http://localhost:4444/grid/console/)**.  
 User will see both the devices on the screen once they are registered.**

**G. Add the Jason files to the project for device1 and device2**

**F. Add the parallel execution TestNG file in the project and run from this file - testng\_parallel\_exec.xml**