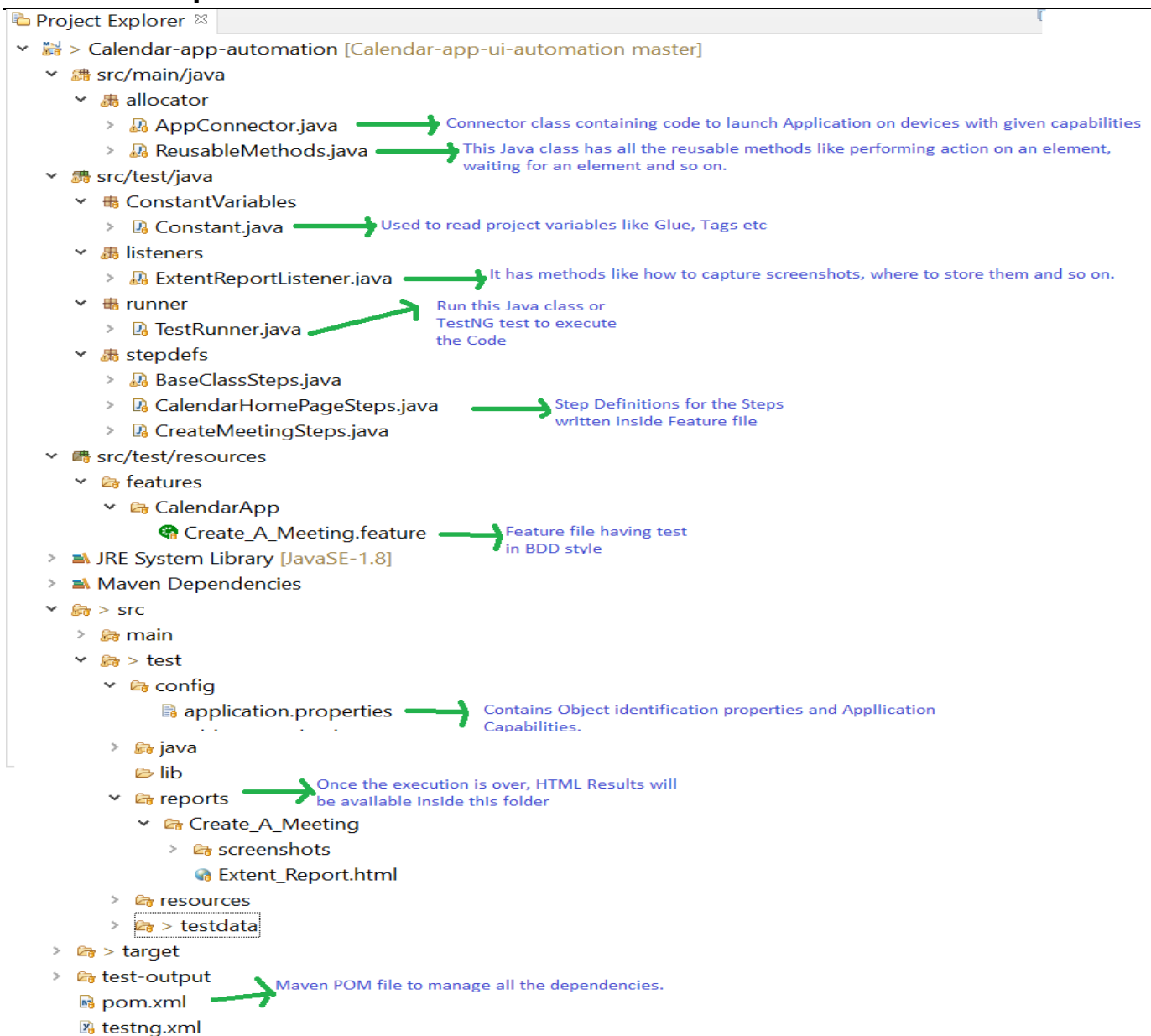# Background

I wrote UI automation tests for creating and verifying a meeting in Calendar application.

Tools/ Library Used

➢ Appium Java Client library

➢ Appium Server

➢ Extent report plugin

➢ My Samsung- Android phone

Solution Components

Project Explorer ⊠

```
✓ 🖳 > Calendar-app-automation [Calendar-app-ui-automation master]
  ✓ 🐂 src/main/java
    ✓ 🐂 allocator
      > 🗓 AppConnector.java ─────────→ Connector class containing code to launch Application on devices with given capabilities
      > 🗓 ReusableMethods.java ─────→ This Java class has all the reusable methods like performing action on an element,
                                         waiting for an element and so on.
  ✓ 🐂 src/test/java
    ✓ 🐂 ConstantVariables
      > 🗓 Constant.java ─────────→ Used to read project variables like Glue, Tags etc
    ✓ 🐂 listeners
      > 🗓 ExtentReportListener.java ─────→ It has methods like how to capture screenshots, where to store them and so on.
    ✓ 🐂 runner
      > 🗓 TestRunner.java ─────→ Run this Java class or
                                    TestNG test to execute
                                    the Code
    ✓ 🐂 stepdefs
      > 🗓 BaseClassSteps.java
      > 🗓 CalendarHomePageSteps.java ─────→ Step Definitions for the Steps
      > 🗓 CreateMeetingSteps.java              written inside Feature file
  ✓ 🐂 src/test/resources
    ✓ 📂 features
      ✓ 📂 CalendarApp
          🌐 Create_A_Meeting.feature ─────→ Feature file having test
                                               in BDD style
  > 🔺 JRE System Library [JavaSE-1.8]
  > 🔺 Maven Dependencies
  ✓ 📂 > src
    > 📂 main
    ✓ 📂 > test
      ✓ 📂 config
          📄 application.properties ─────→ Contains Object identification properties and Application
                                            Capabilities.
      > 📂 java
        📂 lib
      ✓ 📂 reports ─────→ Once the execution is over, HTML Results will
                           be available inside this folder
        ✓ 📂 Create_A_Meeting
          > 📂 screenshots
              🌐 Extent_Report.html
      > 📂 resources
      > 📂 > testdata
  > 📂 > target
  > 📂 test-output
    📄 pom.xml ─────→ Maven POM file to manage all the dependencies.
    📄 testng.xml
```

How to Execute?

Please follow below steps:

1. Ensure that Java and Maven is available on the machine.
2. Extract all the files from ZIP.
3. Launch Command Prompt.
4. Navigate to the project directory in CMD.

   C:\WINDOWS\system32\cmd.exe

   ```
   Microsoft Windows [Version 10.0.17763.1397]
   (c) 2018 Microsoft Corporation. All rights reserved.

   C:\Users\sonra3>cd C:\Automation\Calendar-app-automation

   C:\Automation\Calendar-app-automation>
   ```

5. Once done, run the below Maven command to trigger execution:
   - ***mvn clean test***

```
C:\Automation\Calendar-app-automation>mvn clean test
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------< com.samsung.android:calendar >--------------------
[INFO] Building calendar-app 1.0
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
------------------------------------------------------
 T E S T S
------------------------------------------------------
Running runner.TestRunner
Configuring TestNG with: org.apache.maven.surefire.testng.conf.TestNG652Configurator@5577140b
Thread Id  | Scenario Num      | Step Count
Extent Report location initialized . . .
System Info. set in Extent Report
Sep 06, 2020 8:02:18 AM io.appium.java_client.remote.AppiumCommandExecutor$1 lambda$0
INFO: Detected dialect: W3C
###........CONNECTION SUCCESSFULLY ESTABLISHED WITH NODE DEVICE.......###
It is a working day
isWorkingDate? = true
DURATION - 24657
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 25.591 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:14 min
[INFO] Finished at: 2020-09-06T08:02:24+10:00
[INFO] ------------------------------------------------------------------------

C:\Automation\Calendar-app-automation>
```
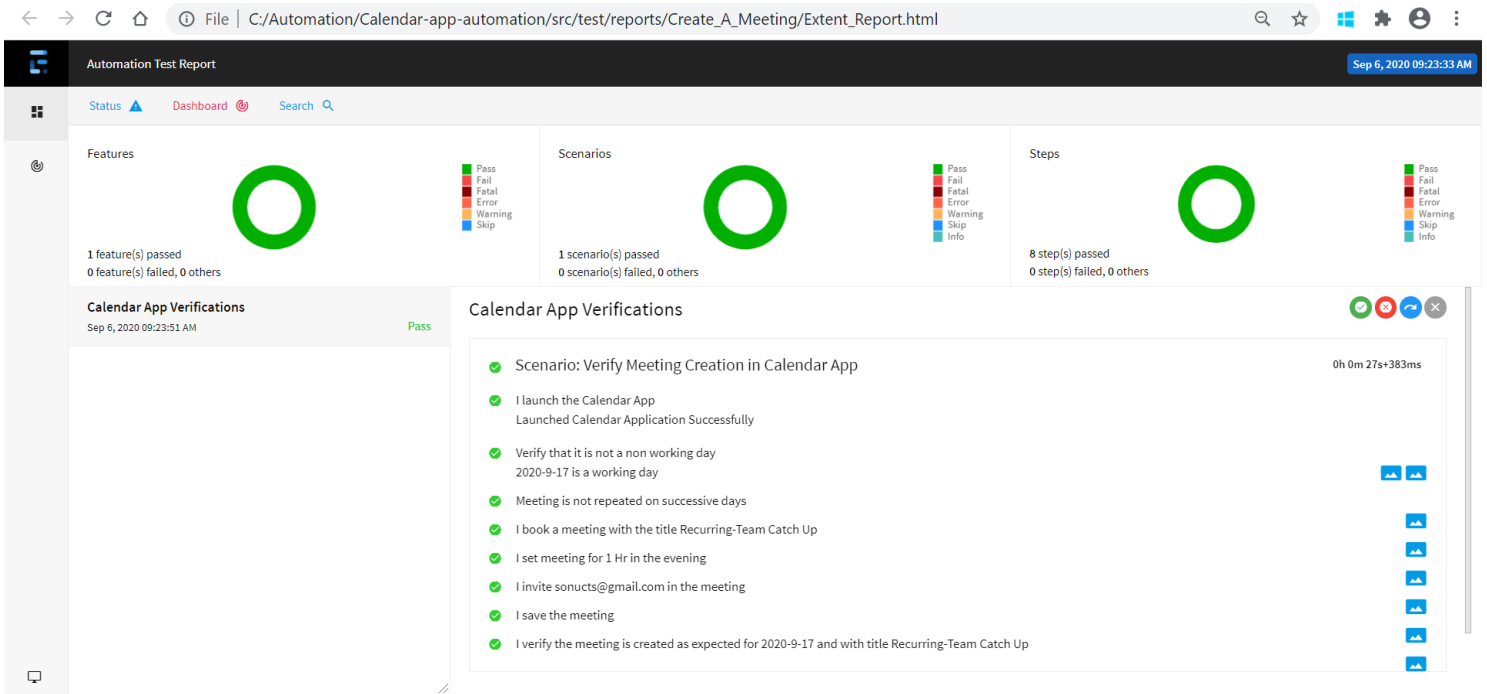
7. Once the execution is over, open the Extent_Report.html file from below path:
     .\Calendar-app-automation\src\test\reports\Create_A_Meeting\Extent_Report.html

Execution Results

Once execution is completed successfully, HTML report will be generated, and will look like below:

Suggested Enhancements

Some of the ideas which is coming in my mind,

1. Currently the solution is Single Threaded, in future we can enhance this to multithreading.
2. Integration with CI/ CD tool.
3. Integration with Jira, so that Jira ticket will be updated by automation test itself.
4. Containers Implementations.
5. Can be integrated with tools like JACOCO or Sonarqube for Code Analysis.
6. We can write Groovy to capture execution logs from Jenkins console and have a summarized HTML page.
7. We can also work little more on Reporting to remove some of the print statements.