

Automation of Expedia application in Java and integrating the same with Extent Reports.

Step 1: Prerequisites

a. Installing JDK and initialising \$JAVA_HOME:

1. Download JDK from the link: <https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html> and install it.
2. Now the environmental variable **JAVA_HOME** should be initialised.

Now initialising the \$JAVA_HOME variable. This can be done by executing the below commands.

```
nano .zprofile
export JAVA_HOME = "$(usr/libexec/java_home)"
```

Example

```
# Setting PATH for Python 3.9
# The original version is saved in .zprofile.pysave
export JAVA_HOME="/Library/Java/JavaVirtualMachines/jdk-11.0.10.jdk/Contents/Home"
```

b. Installing Appium Desktop App:

1. Download Appium Desktop app from the link: <https://appium.io/downloads.html> and install it.

This can be used to create the automation script.

c. Installing Code Editor:

1. Here, Visual studio code is used.
2. Download Visual Studio code from the link: <https://code.visualstudio.com/download> and install it.
3. You can configure VSC according to your needs such as changing themes, font style, size etc., Download the extension "Java extensions" and all the other extensions which support compiling of java code.

Any other code editor can also be used.

Once all the above steps are completed, the setup is ready.

Step 2: Adding dependencies:

1. Create a new maven project in the code editor. Once the project is created few dependencies should be added to pom.xml. The dependencies can be downloaded using the following links. Download the latest stable version available.
 - Selenium: <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java>
 - Appium: <https://mvnrepository.com/artifact/io.appium/java-client>
 - Web Driver Managar: <https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager>

- TestNG: <https://mvnrepository.com/artifact/org.testng/testng>
- Extent Reports: <https://mvnrepository.com/artifact/com.aventstack/extentreports>

```

<dependencies>

    <dependency>

        . . .

    </dependency>

    <dependency>

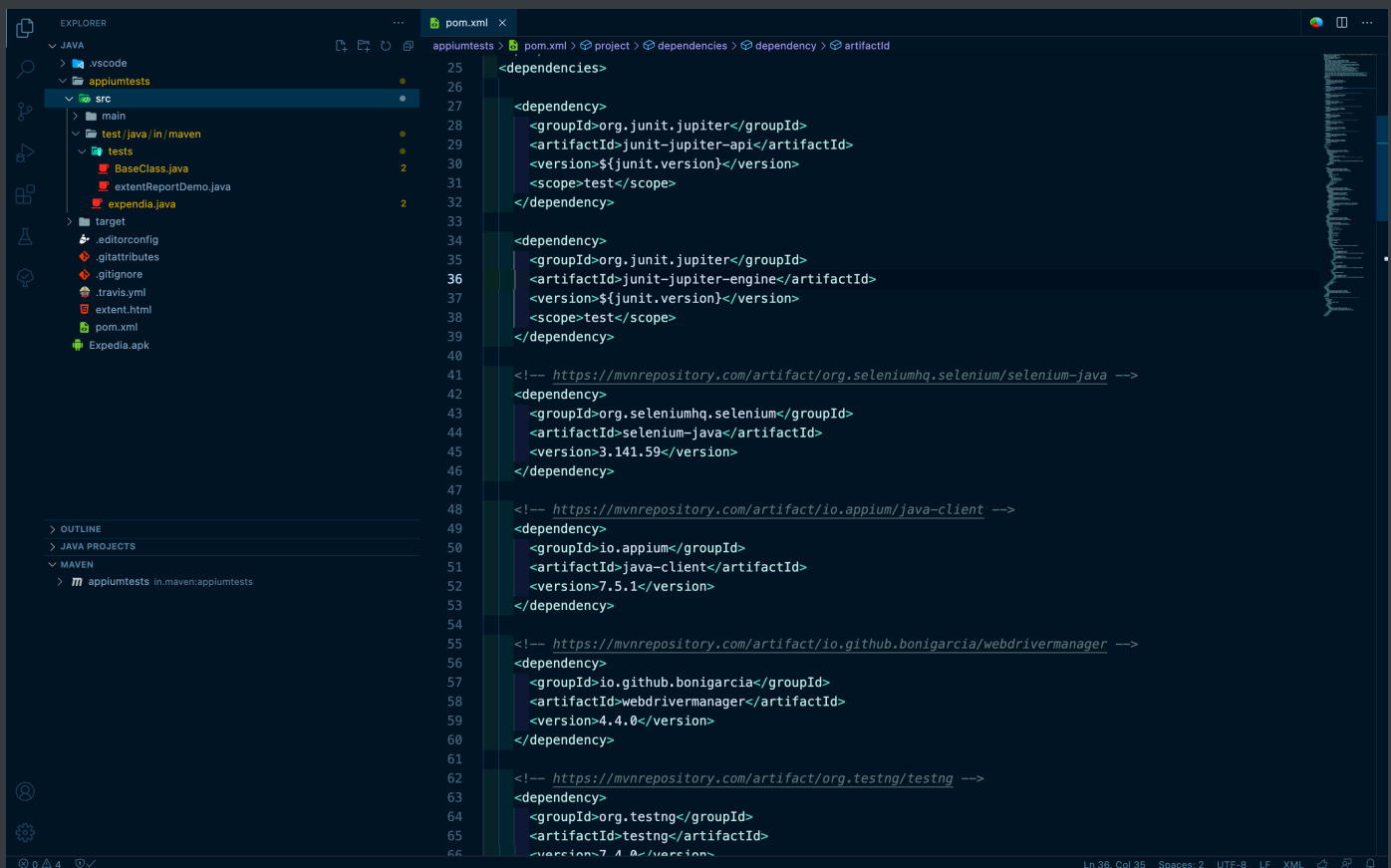
        ..

    </dependency>

</dependencies>

```

Note: All the dependencies should be added inside `<dependencies>` tag.

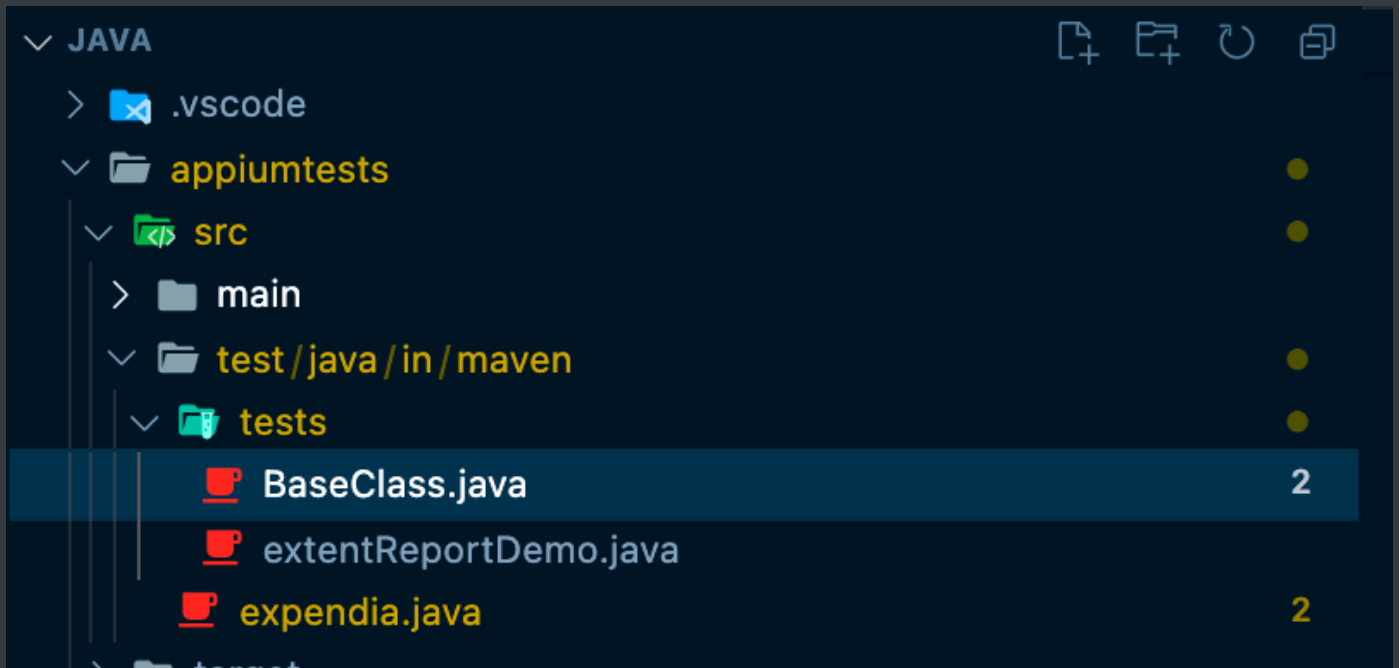


```

25 <dependencies>
26
27 <dependency>
28     <groupId>org.junit.jupiter</groupId>
29     <artifactId>junit-jupiter-api</artifactId>
30     <version>${junit.version}</version>
31     <scope>test</scope>
32 </dependency>
33
34 <dependency>
35     <groupId>org.junit.jupiter</groupId>
36     <artifactId>junit-jupiter-engine</artifactId>
37     <version>${junit.version}</version>
38     <scope>test</scope>
39 </dependency>
40
41 <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
42 <dependency>
43     <groupId>org.seleniumhq.selenium</groupId>
44     <artifactId>selenium-java</artifactId>
45     <version>3.141.59</version>
46 </dependency>
47
48 <!-- https://mvnrepository.com/artifact/io.appium/java-client -->
49 <dependency>
50     <groupId>io.appium</groupId>
51     <artifactId>java-client</artifactId>
52     <version>7.5.1</version>
53 </dependency>
54
55 <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
56 <dependency>
57     <groupId>io.github.bonigarcia</groupId>
58     <artifactId>webdrivermanager</artifactId>
59     <version>4.4.0</version>
60 </dependency>
61
62 <!-- https://mvnrepository.com/artifact/org.testng/testng -->
63 <dependency>
64     <groupId>org.testng</groupId>
65     <artifactId>testng</artifactId>
66     <version>7.4.0</version>

```

Once the dependencies are added, coding can be started. The code file should be saved in **src/test/java**.



Step 3: Integrating Extent Reports with automation code:

Create a new class for integration in the same directory where the test class is created.



1. Make sure you import the packages in Extent Report class.

```
import com.aventstack.extentreports.ExtentReports;  
import com.aventstack.extentreports.reporter.ExtentSparkReporter;
```

2. Code inside the Extent Report class

```
public class extentReportDemo {  
  
    public static ExtentReports report(){  
  
        //Creating the reports  
        ExtentReports extent = new ExtentReports();  
        ExtentSparkReporter spark = new  
ExtentSparkReporter("target/Spark.html");  
  
        extent.attachReporter(spark);  
  
        //Returning the created report  
        return extent;  
    }  
  
}
```

Note: Step 3 to Step 7 should be done in the Base class where the automation code is written.

3. Import the following packages.

```
// Importing Extent Reports packages  
import com.aventstack.extentreports.ExtentReports;  
import com.aventstack.extentreports.ExtentTest;  
import com.aventstack.extentreports.Status;
```

4. Inherit Extent Reports class

```
public class automationCode extends extentReportCode{  
    //your code goes here  
}
```

5. Declare the following variables globally

```
ExtentReports report;  
ExtentTest test;
```

6. Creating a report by calling the report() function from Extent Report class

```
report = extentReportDemo.report();
```

7. We can also write logs after a certain test is executed to obtain the result.

```
try{  
    // Creating the test  
    test = report.createTest( "Automation of Expedia - " +(count));  
    test.log( Status.INFO, "Test just started!" );  
  
    try{  
        // Test passes  
        test.log( Status.PASS, "PASSED!" );  
    }  
  
    catch( Exception e ){  
        // Test Fails  
    }  
}
```

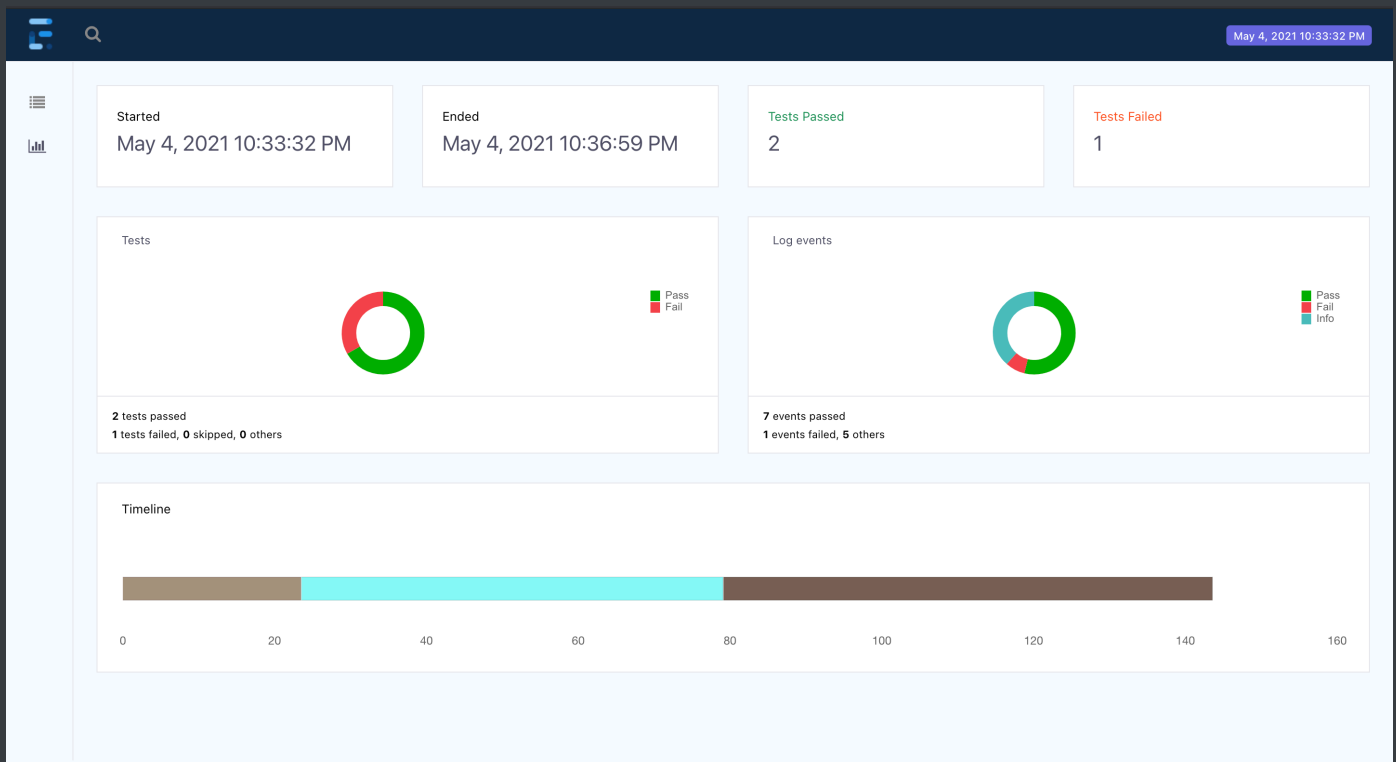
```

        test.log( Status.FAIL, "FAILED!" );
    }

    finally{
        //Keeping track of number of tests executed
        count = count + 1
    }
}

```

8. Now if you execute the automation test code, a HTML report will be created in the path you have specified in the above code. In my case the report will be generated in the same directory and will be named as **Spark.html**.



May 4, 2021 10:33:32 PM

Tests

Automation of Expedia - 1

10:33:33 PM / 00:00:23:533

Fail

Automation of Expedia - 2

10:34:32 PM / 00:00:55:613

Pass

Automation of Expedia - 3

10:35:42 PM / 00:01:04:488

Pass

Automation of Expedia - 1

05.04.2021 10:33:33 PM

05.04.2021 10:33:56 PM

00:00:23:533

#test-id=1

STATUS

TIMESTAMP

DETAILS

Info

10:33:33 PM

Test just started!

Pass

10:33:40 PM

Entered!

Fail

10:33:56 PM

Failed!

This will be the end product. This is how the report will be generated after the test is executed.