**Allure report integration to java maven project**

To generate your first report you will need to go through just a few simple steps:

Downloading and installing Allure commandline application suitable for your environment.

Locating test execution data that you have to build a report on.

**Manual installation**

1.Download the latest version as zip archive from Maven Central(Link address : **<http://repo.maven.apache.org/maven2/io/qameta/allure/allure-commandline/>**)

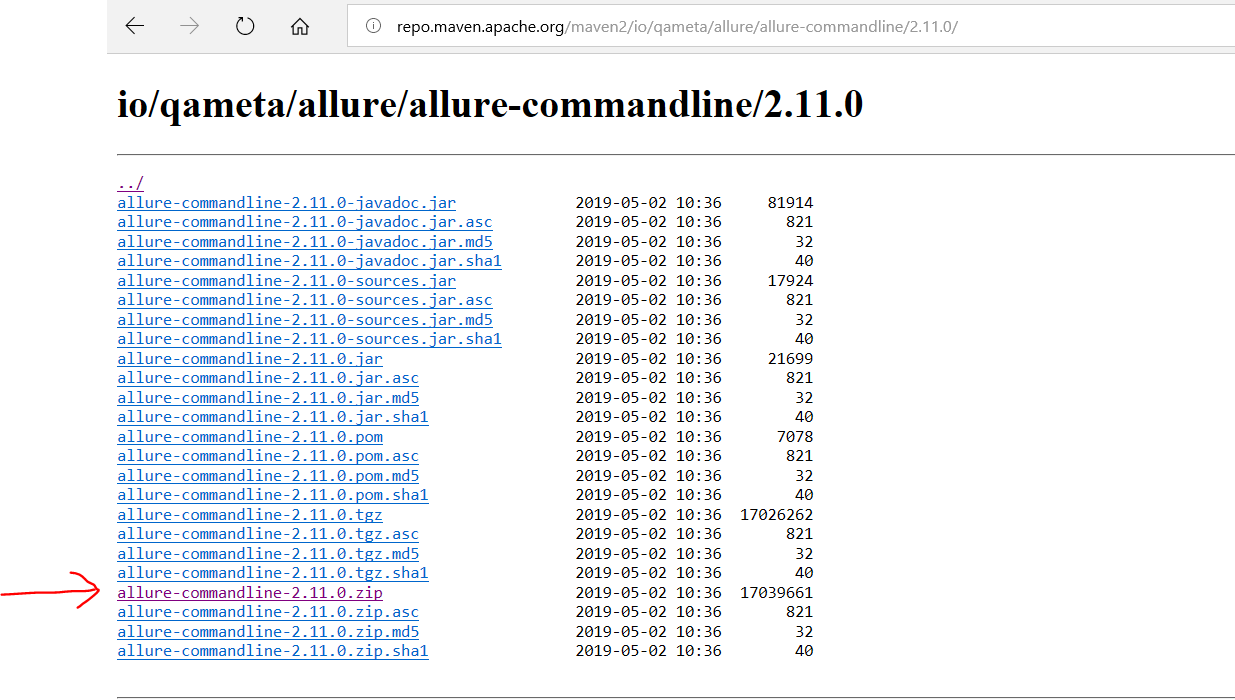
2.Unpack the archive to allure-commandline directory.

3.Navigate to **bin** directory.

4.Use **allure.bat** for Windows or **allure** for other Unix platforms.

5.Add **allure** to system PATH.

6.Execute **allure --version** in console to make sure that allure is now available.



**Modify Maven pom.xml**

For Allure integration, you need to add below maven dependencies into your ****pom.xml file****.

### ****Properties****

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<aspectj.version>1.8.10</aspectj.version>

</properties>

### ****Allure TestNG Dependency****

<dependency>

<groupId>io.qameta.allure</groupId>

<artifactId>allure-testng</artifactId>

<version>2.12.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/ru.yandex.qatools.allure/allure-java-annotations -->

<dependency>

<groupId>ru.yandex.qatools.allure</groupId>

<artifactId>allure-java-annotations</artifactId>

<version>1.5.4</version>

</dependency>

### ****Build Section****

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.20</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>LoginTest\_testng.xml</suiteXmlFile>

</suiteXmlFiles>

<argLine>

-javaagent:"${settings.localRepository}/org/aspectj/aspectjweaver/${aspectj.version}/aspectjweaver-${aspectj.version}.jar"

</argLine>

<systemPropertyVariables>

<allure.results.directory>${project.build.directory}/allure-results</allure.results.directory>

<allure.link.issue.pattern>https://example.org/browse/{}</allure.link.issue.pattern>

<allure.link.tms.pattern>https://example.org/browse/{}</allure.link.tms.pattern>

</systemPropertyVariables>

<dependencies>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>${aspectj.version}</version>

</dependency>

</dependencies>

<reporting>

<excludeDefaults>true</excludeDefaults>

<plugins>

<plugin>

<groupId>io.qameta.allure</groupId>

<artifactId>allure-maven</artifactId>

<version>2.8</version>

</plugin>

</plugins>

</reporting>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-source-plugin</artifactId>

<executions>

<execution>

<id>attach-sources</id>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<fork>true</fork>

<executable>C:\Program Files\Java\jdk1.8.0\_201\bin\javac.exe</executable>

</configuration>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.7.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-site-plugin</artifactId>

<version>3.7.1</version>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.0.0</version>

</plugin>

</plugins>

<resources>

<resource>

<directory>src/main/resources</directory>

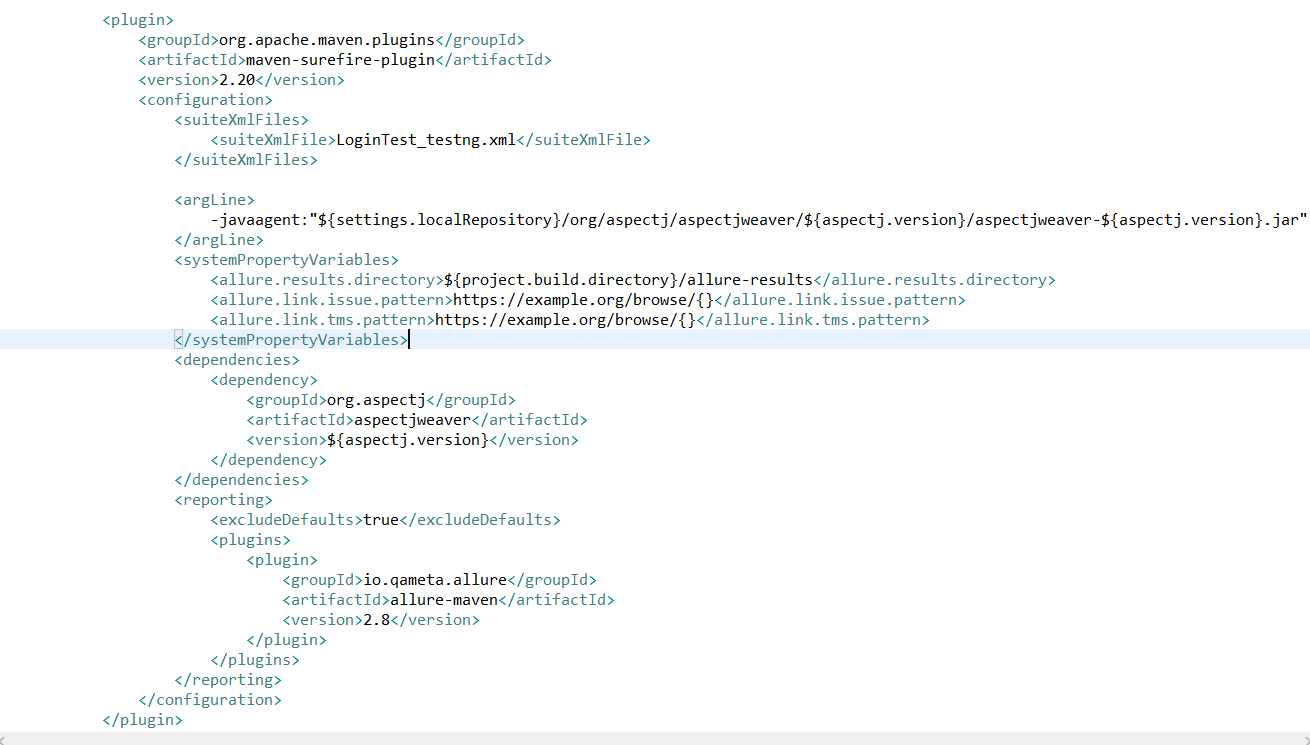
<filtering>true</filtering>

</resource>

</resources>

</build>

**Please see below screenshot to add build Tag in POM.XML**



### ****Reporting****

**<reporting>**

**<excludeDefaults>true</excludeDefaults>**

**<plugins>**

**<plugin>**

**<groupId>io.qameta.allure</groupId>**

**<artifactId>allure-maven</artifactId>**

**<version>2.10.0</version>**

**<configuration>**

**<reportVersion>${allure.version}</reportVersion>**

**</configuration>**

**</plugin>**

**</plugins>**

**</reporting>**

**Please see below screenshot to add Reporting Tag in POM.XML**



## ****Configure Allure Features and Modify Test Codes****

In order to make our test report more understandable, I used ****description**** property of ****@test**** annotation.

@Test (priority = 0, description=”Invalid Login Scenario with wrong username and password.”)

Also, you can add an additional description with ****@Description**** annotation:

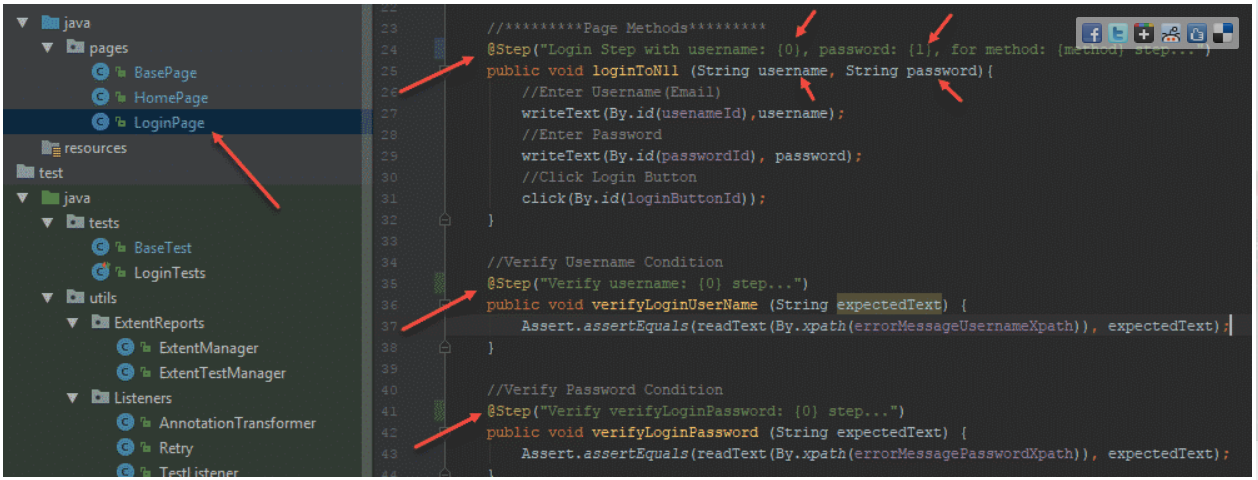
@Description(“Test Description: Login test with wrong username and wrong password.”)

The result will be shown as below:



### ****Steps****

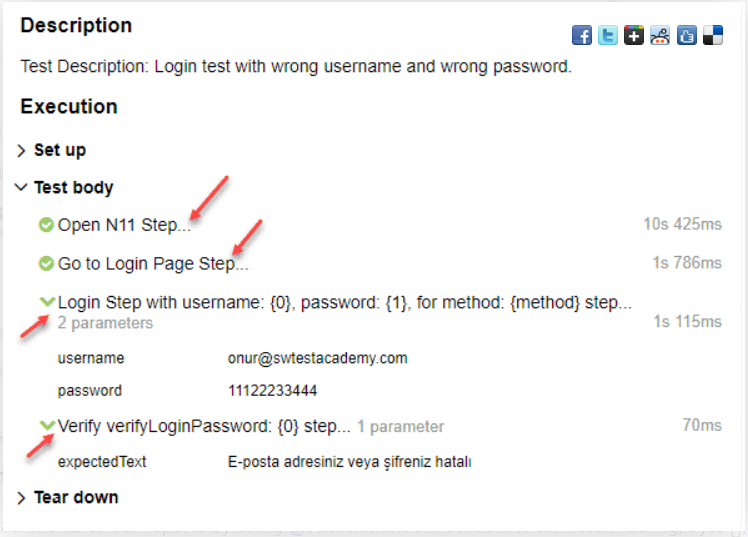
Steps are test actions in our test scenarios. They can be used for any testing scenario. Thus, we need to define steps in a generic place in our test project. In order to define a step, we need to use ****@Step**** annotation. In our project, steps are defined at our page classes. LogintoN11, verifyLoginUserName, verifyLoginPassword, all of these methods are our test steps. Therefore, we need to add ****@Step**** annotation above these methods as shown below. Here, we can also ****pass parameters with {} syntax****. For example, at first method, ****{0} is first parameter – username****, ****{1} is second parameter – password****.



****HomePage Steps:****



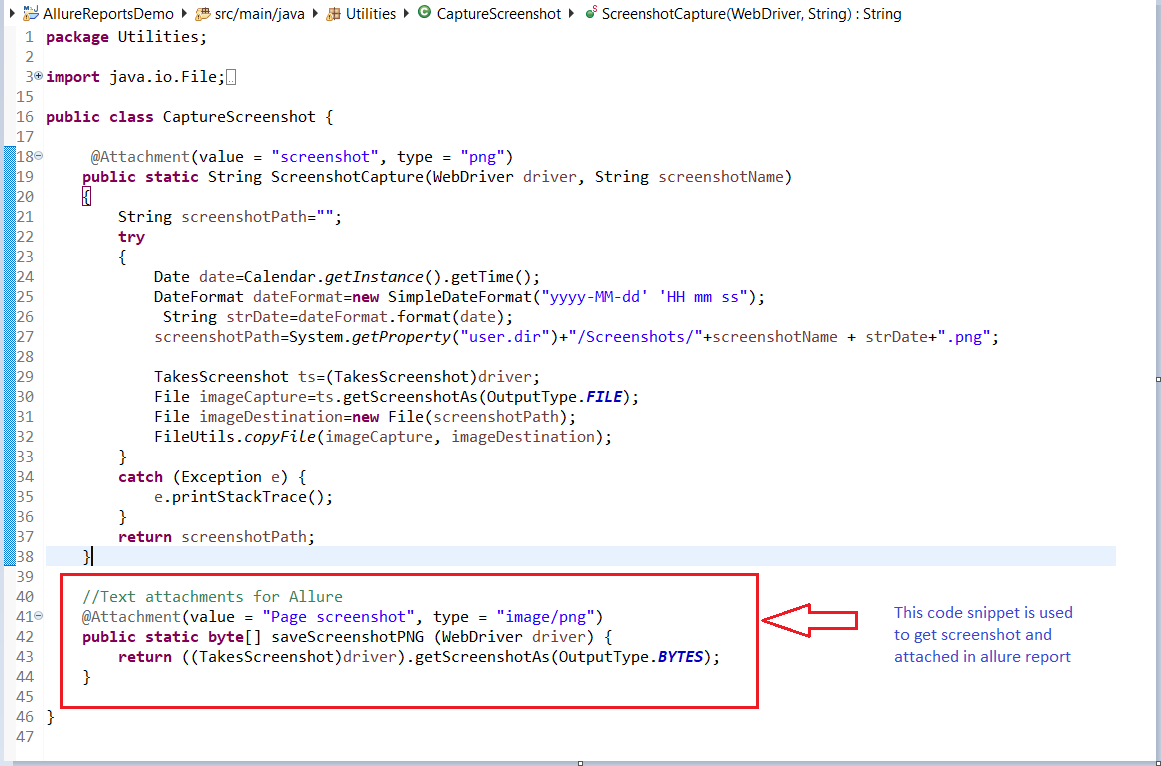
Here are the results. They will look on the report as like below image.

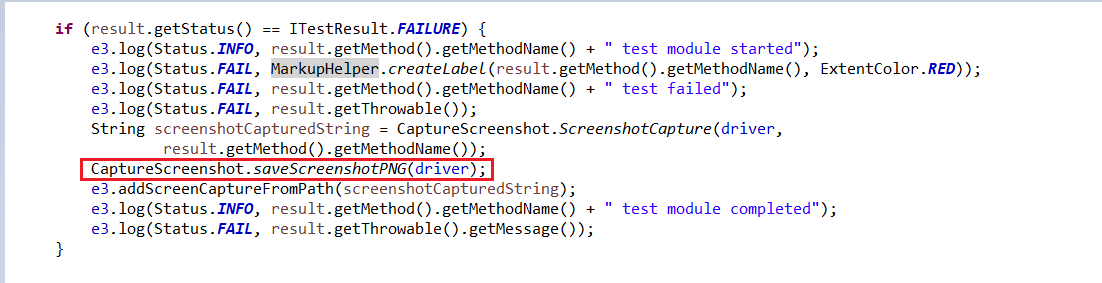


### ****Feature-3: Attachments****

We can add attachments to our reports by using ****@Attachment**** annotation. It can return String, byte [], etc. For example, if we want to attach a screenshot we should return byte[].

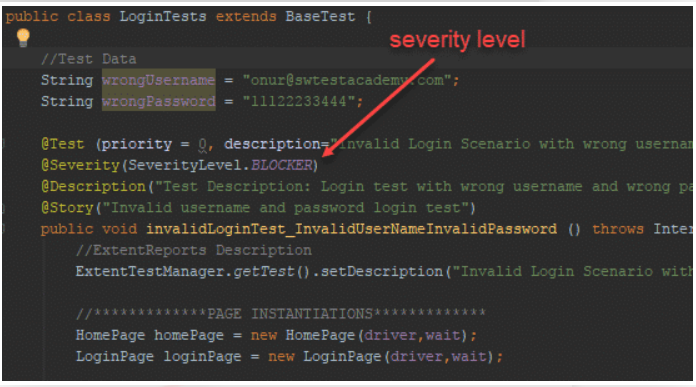
So add below piece of code where your test case failed. In my case I have added below piece of code in extentReportTest failed condition to generate the screenshot. Please go through below.



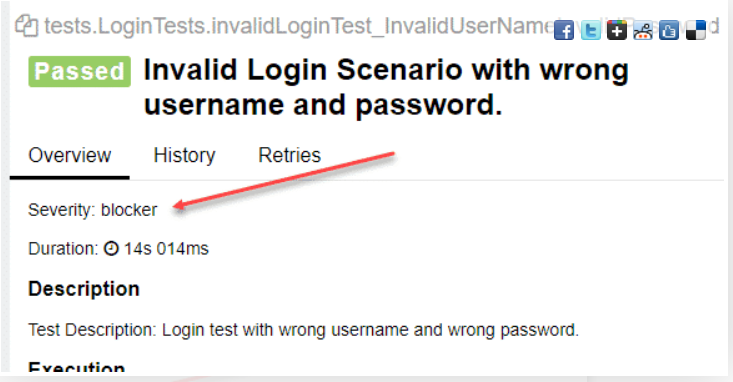


### ****Severity****

We can order test by severity by using ****@Severity**** annotation. I used this feature in tests as shown below.

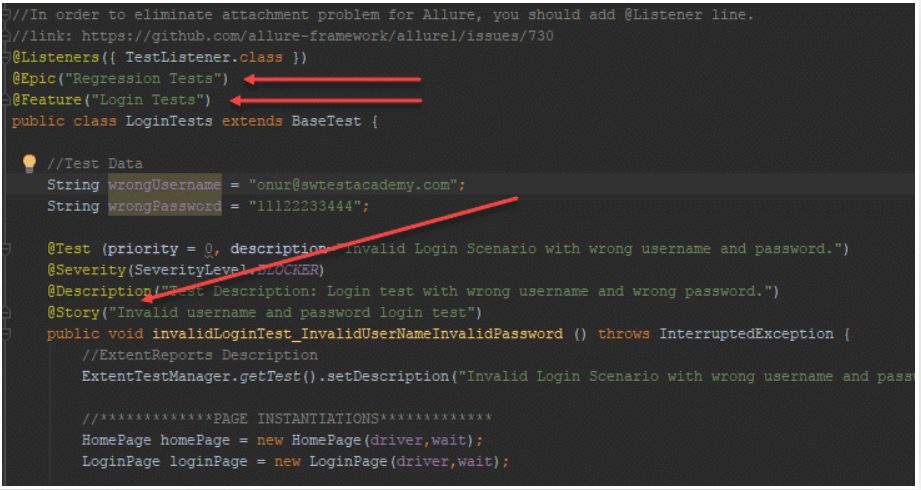


and the result will be like that.

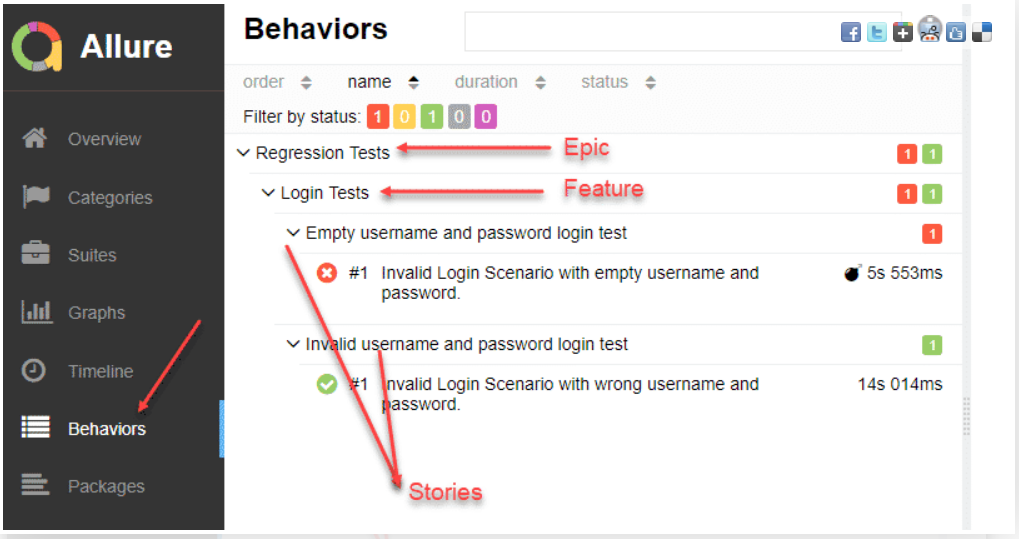


### ****Behaviour Driven Reporting (Features and Stories)****

We can group tests with ****@Epic****, ****@Feature****, and ****@Stories**** annotations.

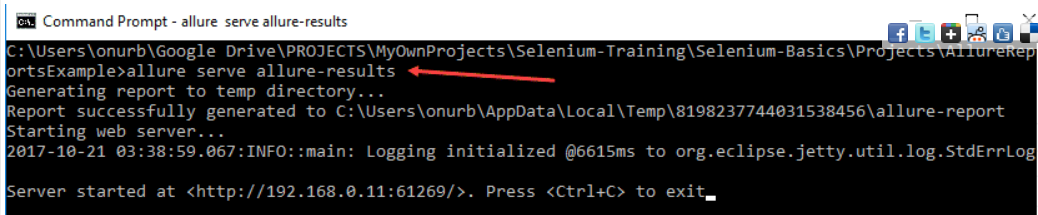


Here is the report result in ****Behaviors Section****.



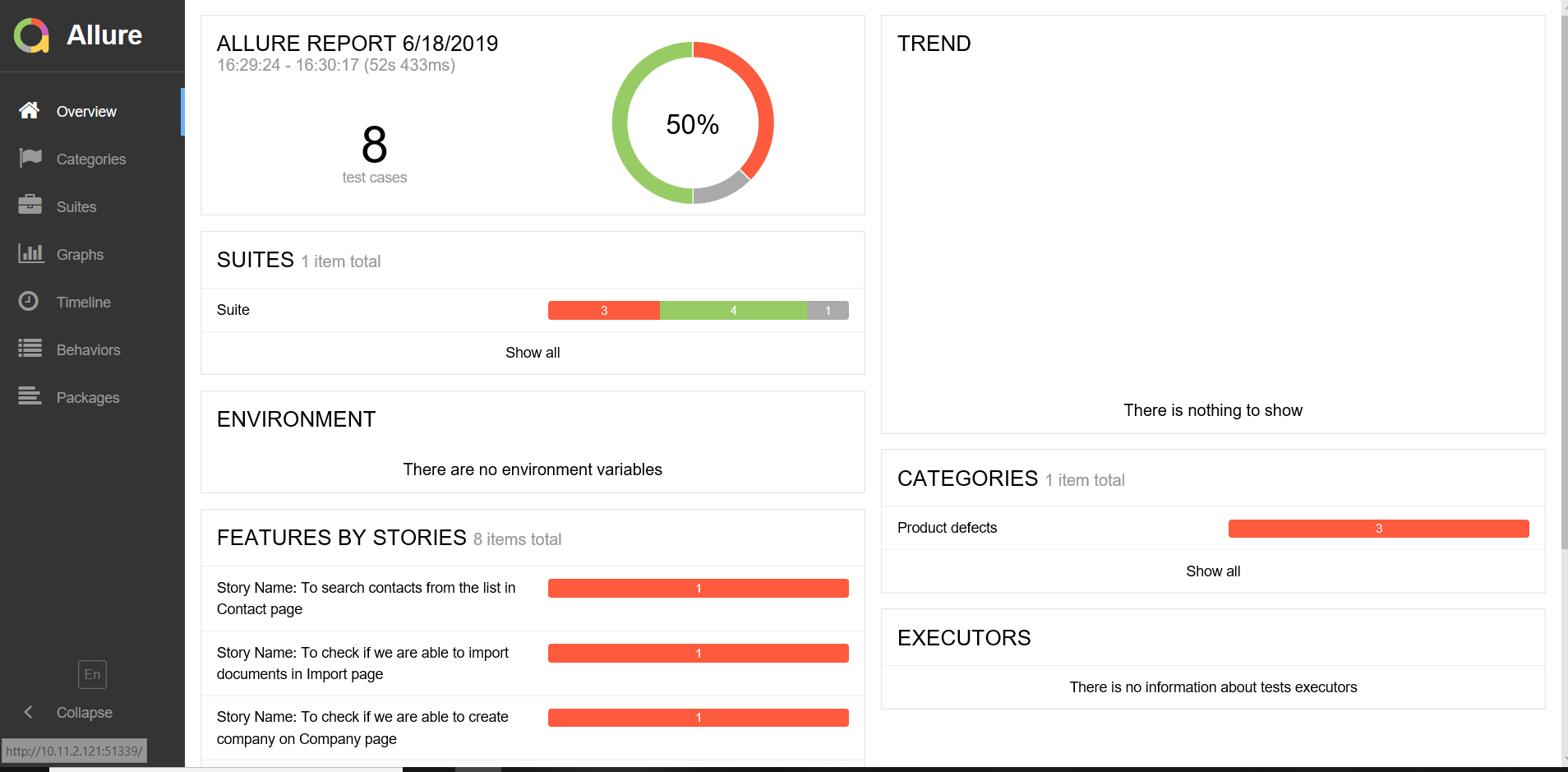
and finally, open a command prompt screen, go to the project directory, and write below command!

****allure serve allure-results****

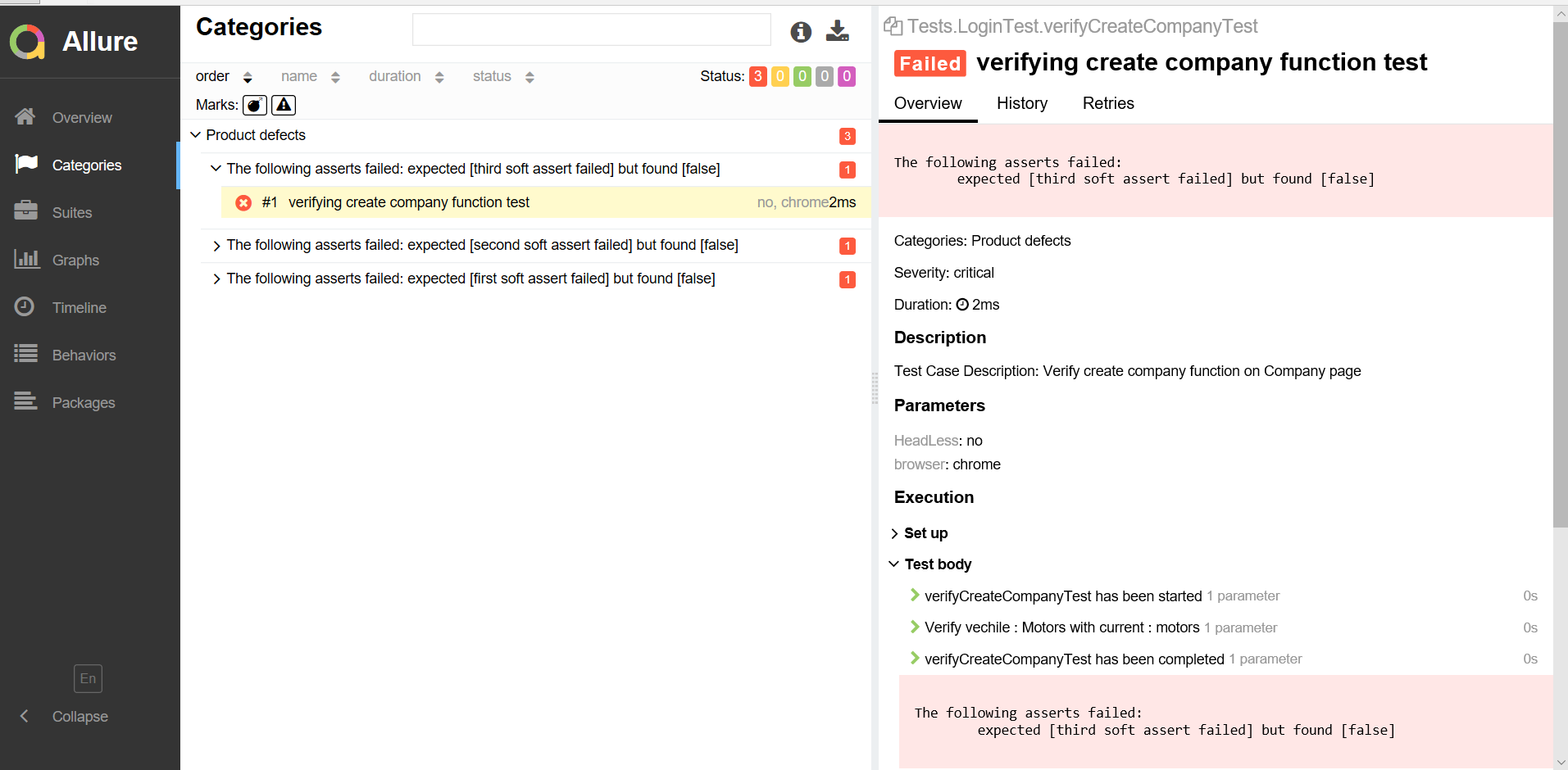


and, you will see the beautiful Allure Test Report as shown below.

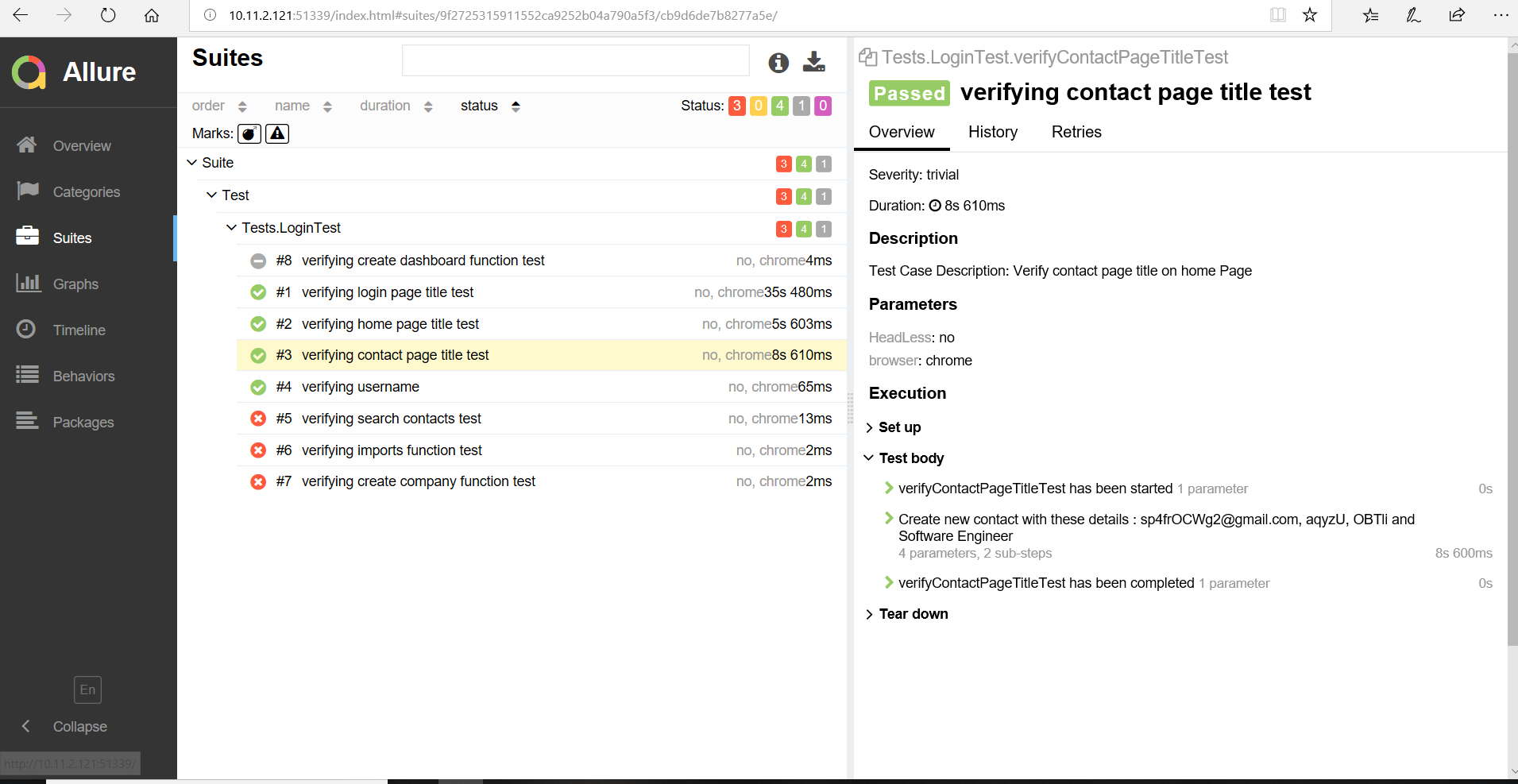
### ****Dashboard****



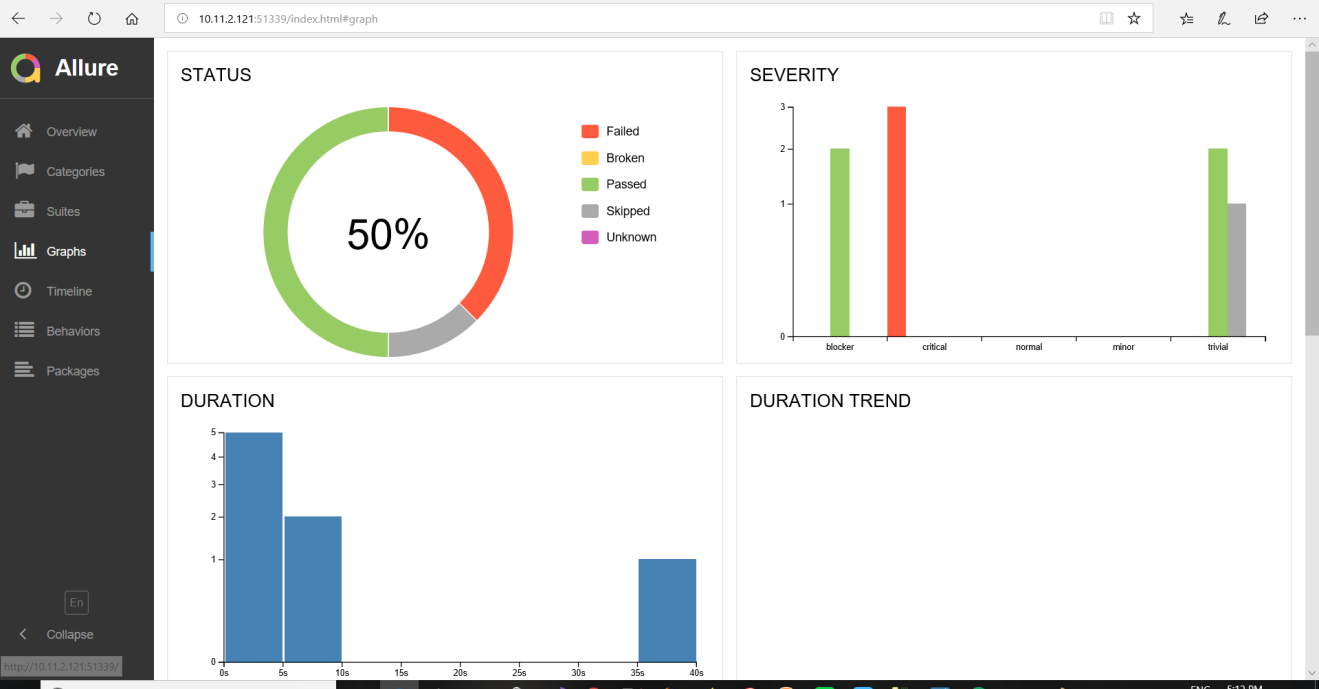
### ****Categories****



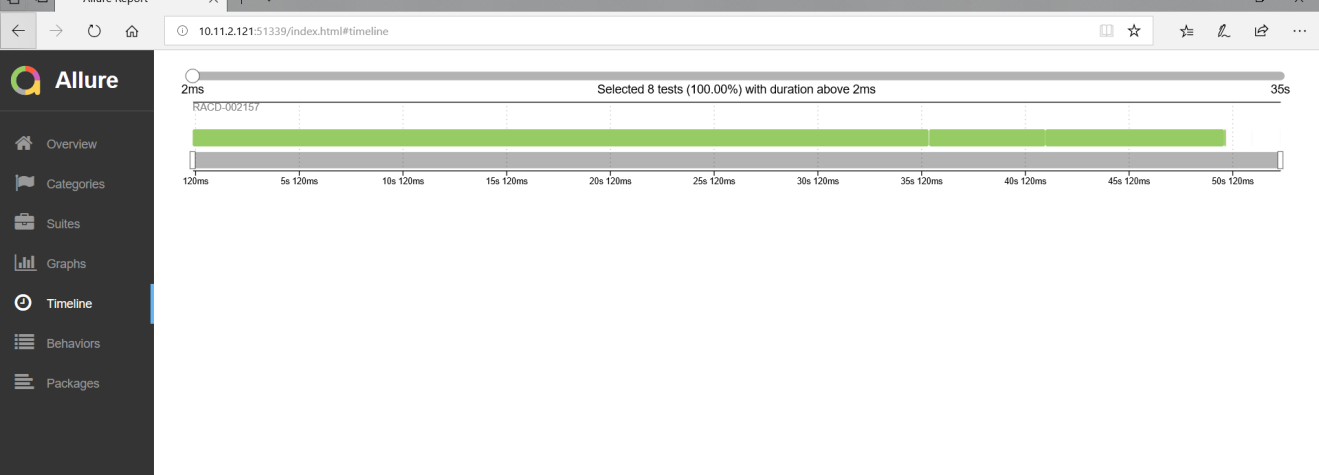
### ****Suites****



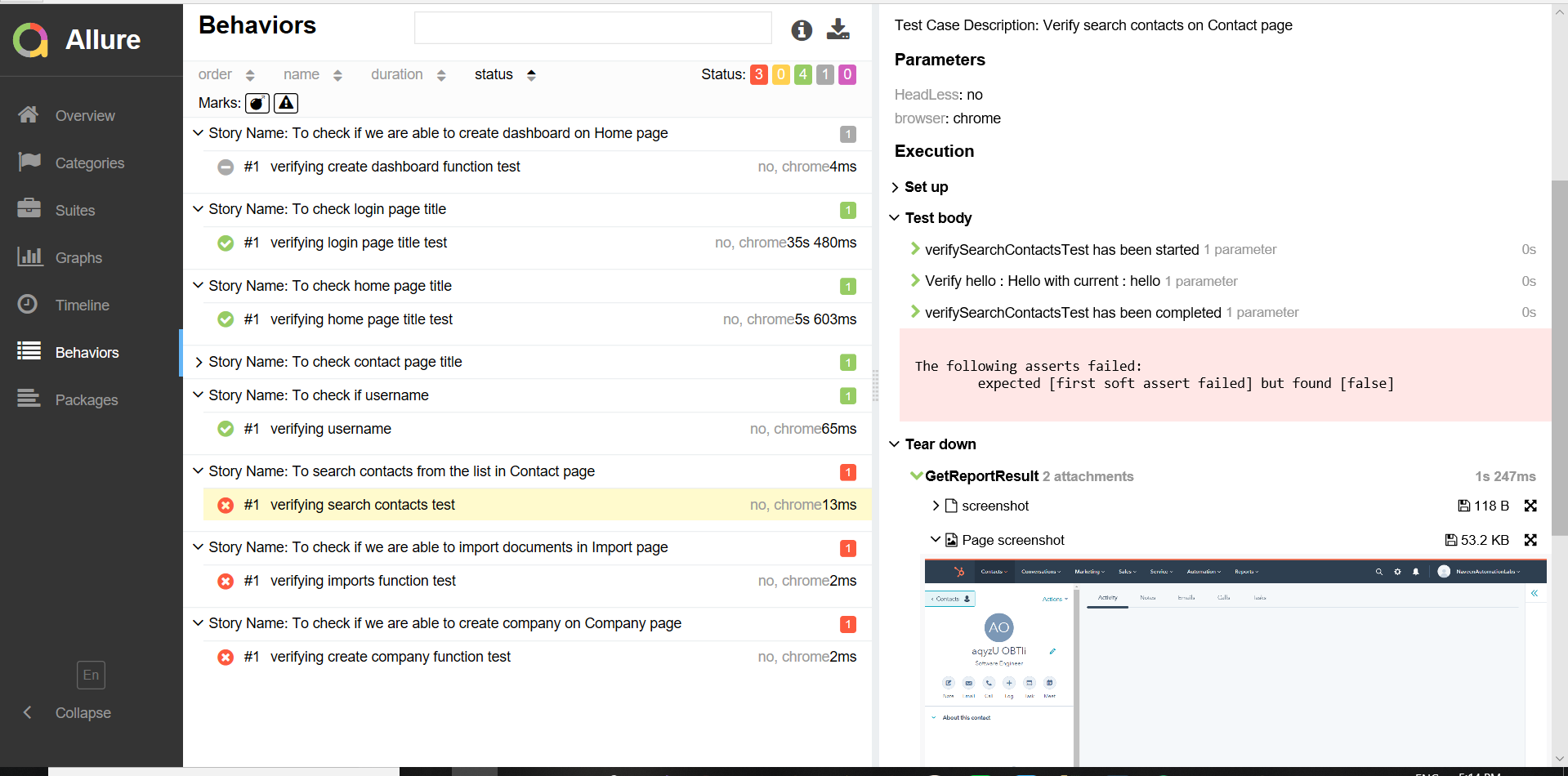
### ****Graphs****



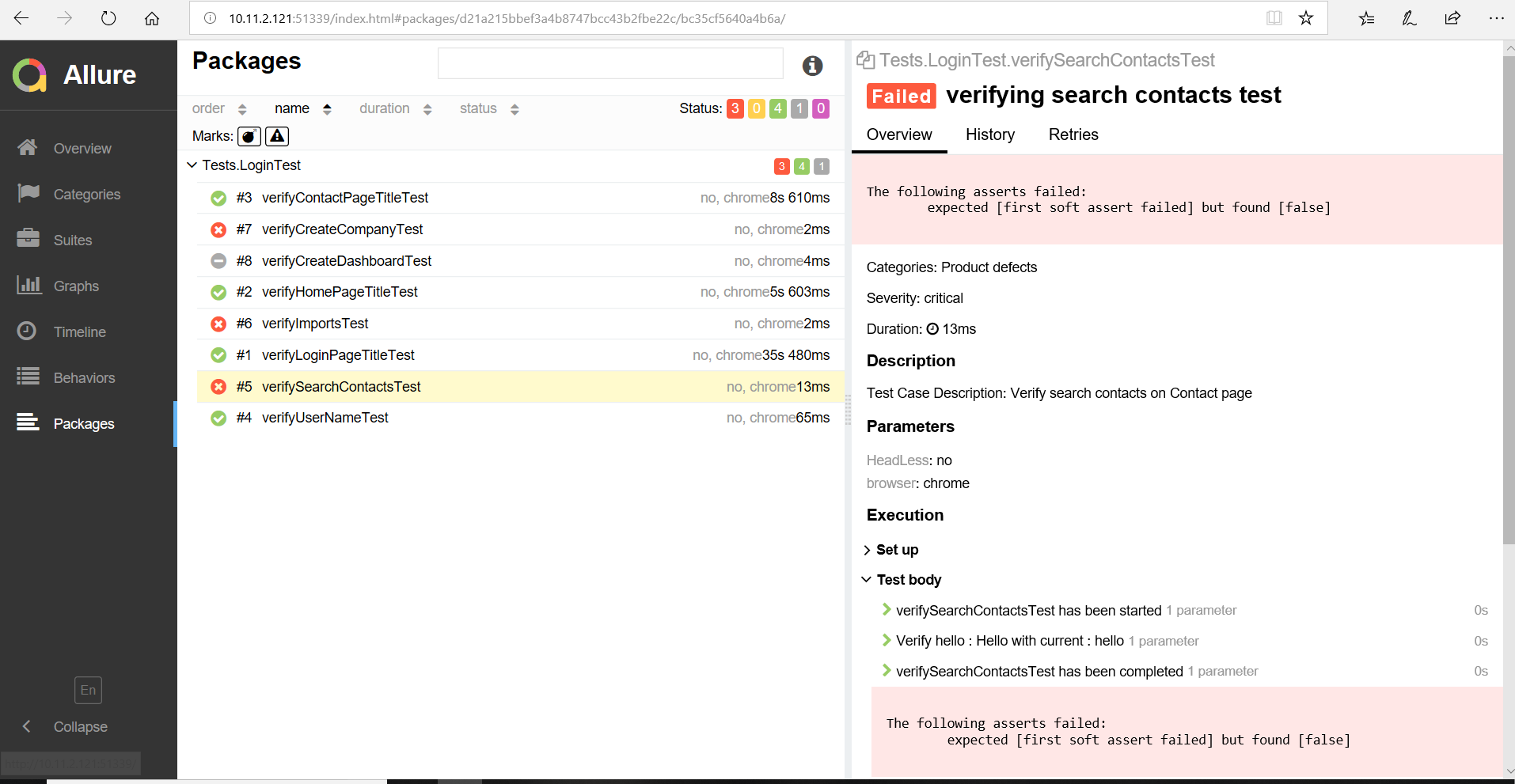
### ****Timeline****



### ****Behaviors****



### ****Packages****



### To set up the Environment

To add information to Environment widget just create **environment.properties** (or **environment.xml**) file to **allure-results** directory before report generation.

**Please see below screenshot :-**

