1. **What is the use of framework in automation testing**

Framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing. Advantages can be in different forms like ease of scripting, scalability, modularity, understandability, process definition, re-usability, cost, maintenance etc. Thus, to be able to grab these benefits, developers are advised to use one or more of the Test Automation Framework.

1. **What are things u consider/keep in framework**

1. What is the shelf of the framework?

2. What percentage of application is going to be automated?

3. How stable is the application under test (AUT).

4. Testing skill of the target audience.

5. Who is the target audience for using the test framework?

6. Return of Investment on the automation efforts.

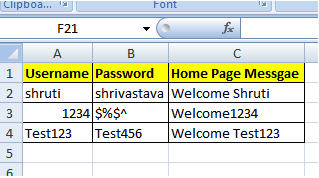
7. Easy availability of the automation tests.

8. How Would You Want Your Data In Framework

10. Environment to support the test framework.

1. **Different types of frameworks we have in selenium/qtp**
2. /\*Module Based Testing Framework
3. Library Architecture Testing Framework\*/
4. Data Driven Testing Framework
5. Keyword Driven Testing Framework
6. Hybrid Testing Framework
7. Behavior Driven Development Framework
8. **What is data driven frame work and when do we use it?**

While automating or testing any application, at times it may be required to test the same functionality multiple times with the different set of input data. Thus, in such cases, we can’t let the test data embedded in the test script. Hence it is advised to retain test data into some external data base outside the test scripts.



1. **What is keyword frame work and when do we use it?**

The Keyword driven testing framework is an extension to Data driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.

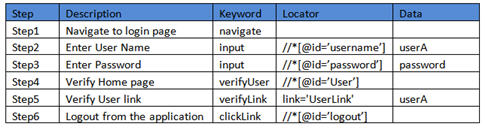
These set of code are known as Keywords and hence the framework is so named. Key words are self-guiding as to what actions needs to be performed on the application.

Ex:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Test-Automation-Frameworks-8.jpg)

1. **what is hybrid framework and when do we use it**

As the name suggests, the Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Test-Automation-Frameworks-10.jpg)

1. **What is POM and modular framework**

A **Project Object Model** or **POM** is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects

In most of the web application we have few set of actions which are always executed in the series of actions. Rather than writing those actions again and again in our test, we can club those actions in to a method and then calling that method in our test script. Modularity avoids duplicacy of code. In future if there is any change in the series of action, all you have to do is to make changes in your main modular method script. No test case will be impacted with the change.

1. **What is TestNG and Junit?**

**TestNG** is a testing framework for the Java programming language created by Cédric Beust and inspired by JUnit and NUnit. The design goal of **TestNG** is to cover a wider range of test categories: unit, functional, end-to-end, integration, etc., with more powerful and easy-to-use functionalities.

**JUnit** is a unit testing framework for the Java programming language.**JUnit** has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

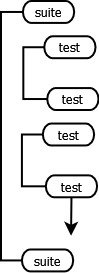
1. writing unit test cases with testng
2. **What are important elements in testng.xml**

Testng.xml is the main configuration file that defines the suite and tests. Suite is top level element in TestNG configuration file and is defined by one XML file.

If you need to have more suites then you need to define separate testng.xml file for each suite like database\_testng.xml, feature\_testng.xml, performance\_testng.xml etc.

Structure of testng.xml

suite is the first element of testng.xml. A suite contains one or more test elements.

[](http://www.javarticles.com/wp-content/uploads/2015/02/testng_main_structure1.png)

*Top level structure of testng*

1. **What are different annotations in testng and junit**

|  |  |  |
| --- | --- | --- |
| Feature | JUnit 4 | TestNG |
| test annotation | @Test | @Test |
| run before all tests in this suite have run | — | @BeforeSuite |
| run after all tests in this suite have run | — | @AfterSuite |
| run before the test | — | @BeforeTest |
| run after the test | — | @AfterTest |
| run before the first test method that belongs to any of these groups is invoked | — | @BeforeGroups |
| run after the last test method that belongs to any of these groups is invoked | — | @AfterGroups |
| run before the first test method in the current class is invoked | @BeforeClass | @BeforeClass |
| run after all the test methods in the current class have been run | @AfterClass | @AfterClass |
| run before each test method | @Before | @BeforeMethod |
| run after each test method | @After | @AfterMethod |
| ignore test | @ignore | @Test(enbale=false) |
| expected exception | @Test(expected = ArithmeticException.class) | @Test(expectedExceptions = ArithmeticException.class) |
| timeout | @Test(timeout = 1000) | @Test(timeout = 1000) |

SELENIUM WEBDRIVER FRAMEWORK,JUNIT & TESTNG DIFFERENCES

|  |  |  |
| --- | --- | --- |
| **Feature** | **Junit** | **TestNG** |
| test annotation | @test | @test |
| run before all tests in this suite have run | @beforeclass | @beforeclass |
| run after all tests in this suite have run | @afterclass | @afterclass |
| run before each test method | @before | @beforemethod |

1. what is group and suite and parallel execution in testng
2. **How to decide which test cases needs to be automated**

Good test cases for automation are ones that are run frequently and require large amounts of data to perform the same action.

You can get the most benefit out of your automated testing efforts by automating:

* Repetitive tests that run for multiple builds.
* Tests that tend to cause human error.
* Tests that require multiple data sets.
* Frequently used functionality that introduces high risk conditions.
* Tests that are impossible to perform manually.
* Tests that run on several different hardware or software platforms and configurations.
* Tests that take a lot of effort and time when manual testing.

1. **what are the steps in automation testing or when do we automation testing?**

write manual test cases (if they are not there)

out of all manual test cases we will pick test cases that are of regression testing

(test cases that needs to be executed)

put all positive scenarios in regression

we dont put any UI related changes in regression testing

**15) what is the use of dataprovider annotation**

It helps you to write data-driven tests, which essentially means that **same test method can be run multiple times with different data-sets.**

|  |
| --- |
| package com.howtodoinjava.test;    import org.testng.annotations.DataProvider;  import org.testng.annotations.Test;    public class SameClassDataProvider  {      @DataProvider(name = "data-provider")      public Object[][] dataProviderMethod() {          return new Object[][] { { "data one" }, { "data two" } };      }        @Test(dataProvider = "data-provider")      public void testMethod(String data) {          System.out.println("Data is: " + data);      }  } |
|  |

Now run above test. Output of above test run is given below:

|  |
| --- |
| Data is: data one  Data is: data two  PASSED: testMethod("data one")  PASSED: testMethod("data two") |

**16) Difference between junit and testng**

**Similarities Between JUnit and TestNG**

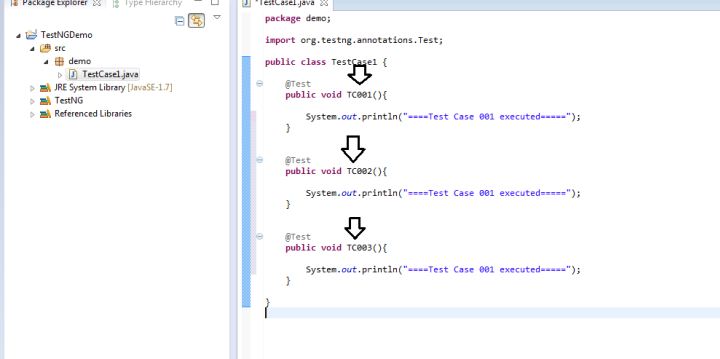
1. We can create test suite in JUnit and TestNG both frameworks.
2. Timeout Test Is possible very easily in both the frameworks.
3. We can ignore specific test case execution of software web application from suite in both the frameworks.
4. It is possible to create expected exception test for software web application in both the frameworks.
5. Annotations - Few annotations are similar in both frameworks suite like @Test, @BeforeClass, @AfterClass. JUnit's Annotations @Before and @After are similar to TestNG's @BeforeMethod and @AfterMethod annotations.

**Difference Between JUnit and TestNG**

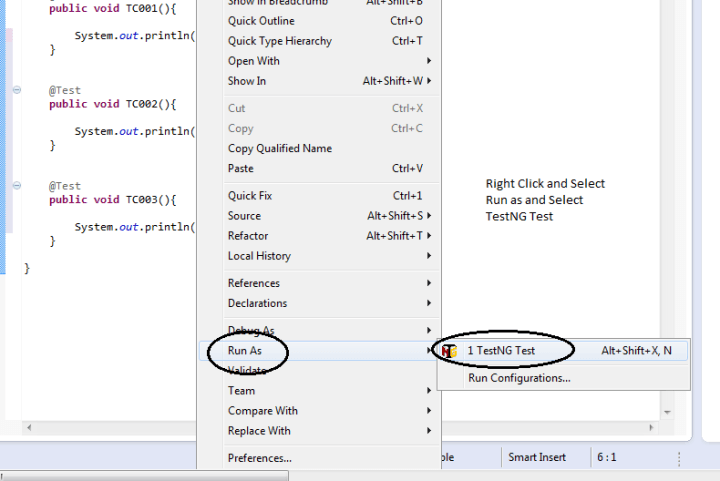
1. In TestNG, Parameterized test configuration is very easy while It is very hard to configure Parameterized test in JUnit.
2. TestNG support group test but it is not supported in JUnit.
3. TestNG has a feature to configure dependency test. Dependency test configuration for software web application is not possible in JUnit.
4. TestNG support @BeforeTest, @AfterTest, @BeforeSuite, @AfterSuite, @BeforeGroups, @AfterGroups which are not supported in JUnit.
5. Test prioritization, Parallel testing is possible in TestNG. It is not supported by JUnit.

**17) how to generate reports using testing**

Step 1- Create a Simple java class and write some test cases (in my case I have written 3 test case)

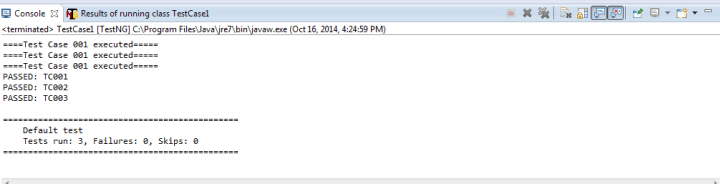
[](https://i0.wp.com/learn-automation.com/wp-content/uploads/2015/03/118.png)

**Step 2- To run script simply right click and select TestNG Test.**

[](https://i1.wp.com/learn-automation.com/wp-content/uploads/2015/03/119.png)

**Step 3- Now test case will start execution and you will get console notification about the status and wait until the program is getting finished.**

**Report type 1- Console output.**

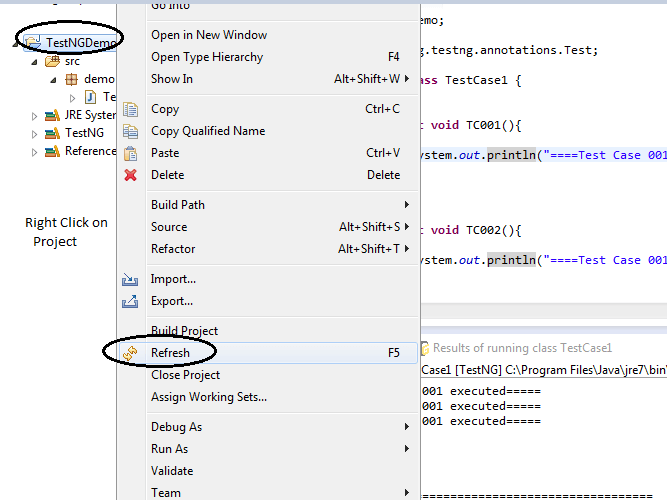
[](https://i1.wp.com/learn-automation.com/wp-content/uploads/2015/03/124.png)

Since console output is not useful because you can’t share and send to others so we required some HTML reports as well

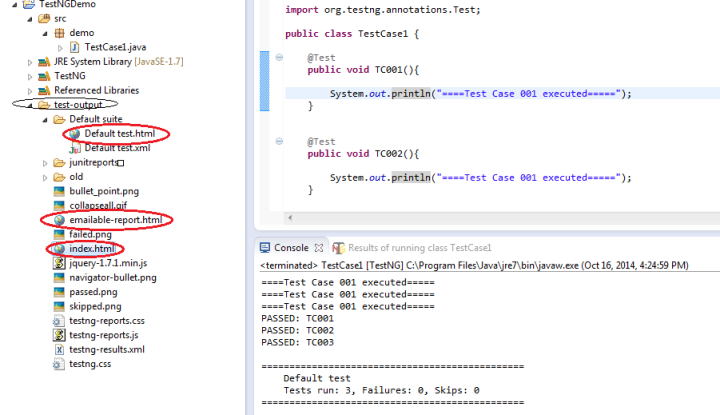
TestNG guys already gave a solution for this so for even single test case you will get HTML reports as well.

**Report type 2- HTML report**

We have to refresh your project and reports folder will come automatically.

[](https://i1.wp.com/learn-automation.com/wp-content/uploads/2015/03/125.png)

**After refreshing, you will get below folder ready**

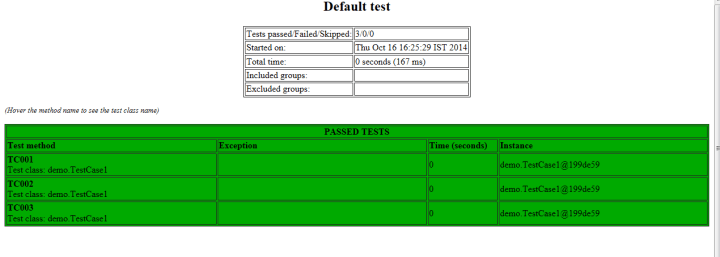
[](https://i0.wp.com/learn-automation.com/wp-content/uploads/2015/03/126.png)

Let’s open each report and check the content

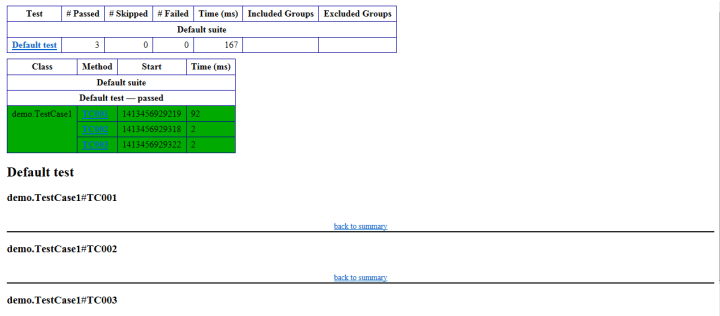
Note- Open Each report in Web Browser select each report then right click and open in browser

## Generate report in selenium webdriver

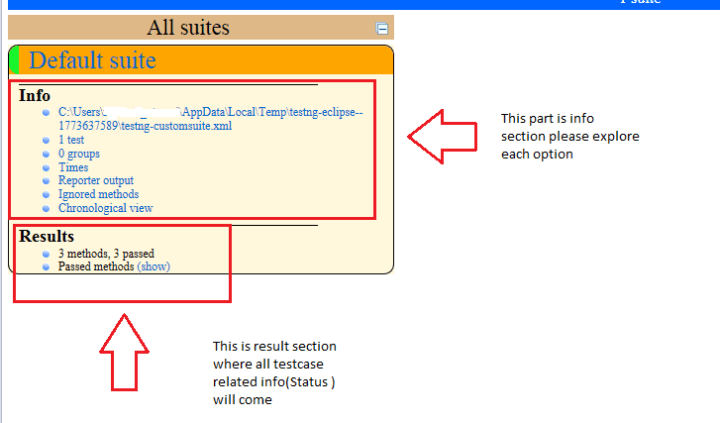
**First open Default test.html**

[](https://i0.wp.com/learn-automation.com/wp-content/uploads/2015/03/127.png)

**Second open emailable-report.html**

[](https://i0.wp.com/learn-automation.com/wp-content/uploads/2015/03/128.png)

**Third, open Index.html**

[](https://i0.wp.com/learn-automation.com/wp-content/uploads/2015/03/129.png)