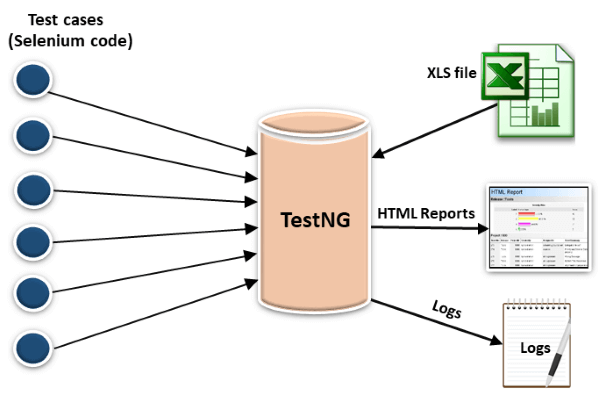
What is TestNG

* TestNG is a very important framework when you are actually developing the framework from scratch level.
* TestNG provides you full control over the test cases and the execution of the test cases. Due to this reason, TestNG is also known as a testing framework.
* Cedric Beust is the developer of a TestNG framework.
* If you want to run a test case A before that as a pre-request you need to run multiple test cases before you begin a test case A. You can set and map with the help of TestNG so that pre-request test cases run first and then only it will trigger a test case A. In such way, you can control the test cases.
* TestNG framework came after Junit, and TestNG framework adds more powerful functionality and easier to use.
* It is an open source automated TestNG framework. In TestNG, NG stands for "**Next Generation**".
* TestNG framework eliminates the limitations of the older framework by providing more powerful and flexible test cases with help of easy annotations, grouping, sequencing and parametrizing.

Advantages of TestNG over Junit



* In TestNG, annotations are easier to understand than Junit.
* It produces the HTML reports for implementation.
* It also generates the Logs.
* In TestNG, there is no constraint available such as @beforeclass and @afterclass which is present in Junit.
* TestNG enables you to group the test cases easily which is not possible in JUnit.
* TestNG supports three additional levels such as @Before/After suite, @Before/AfterTest, and Before/AfterGroup.
* TestNG does not extend any class. TestNG framework allows you to define the test cases where each test case is independent of other test cases.
* It allows you to run the test cases of a particular group. Let's consider a scenario where we have created two groups such as 'Smoke' and 'Regression'. If you want to execute the test cases in a 'Regression' group, then this can only be possible in the TestNG framework.
* Parallel execution of test cases, i.e., running multiple test cases is only possible in the TestNG framework.

Prerequisite

Before learning TestNG, you should have a basic understanding of testing and java.

Audience

Our TestNG Tutorial is designed to help beginners and professionals.

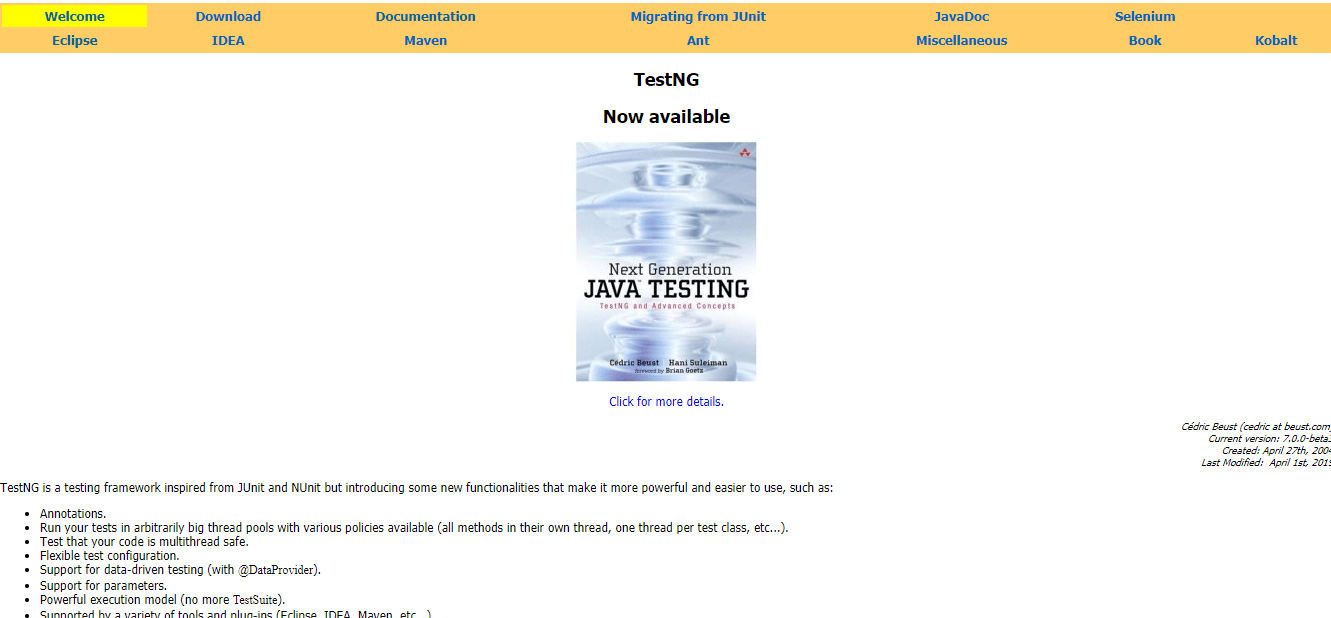
Problems

We assure that you will not find any problem in this TestNG Tutorial. But if there is any mistake, please post the problem in a contact form.

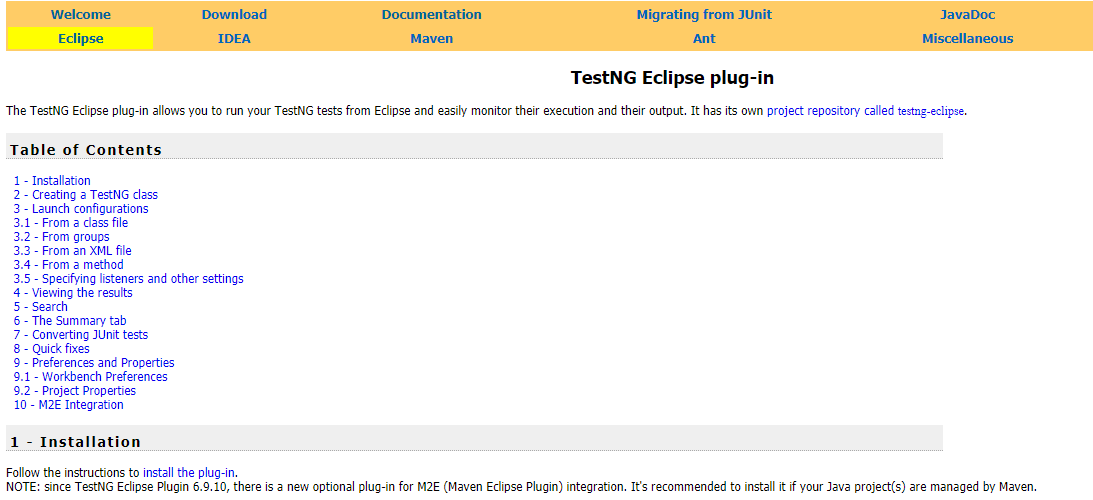
# **TestNG Installation and Configuration in Eclipse**

**Step 1:** Go to the official website of the TestNG. Click on the link given below: <https://testng.org/doc/>

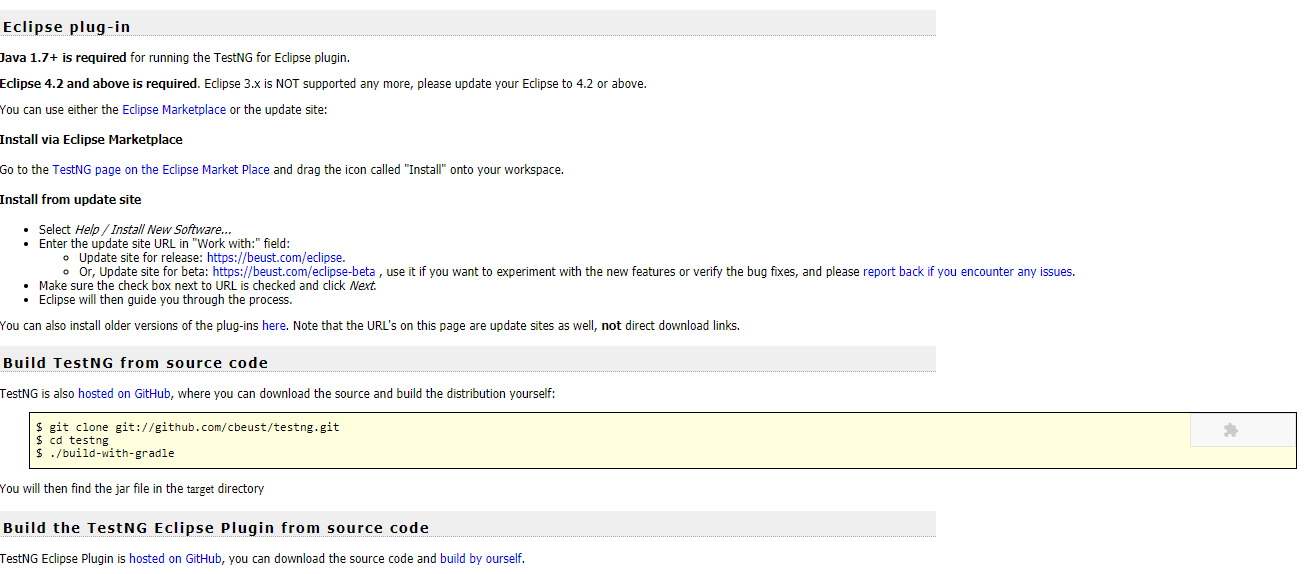
On clicking the above link, the screen appears as shown below:



**Step 2:** Click on the Eclipse appearing on the menu bar. After clicking on the Eclipse, the screen appears as shown below:

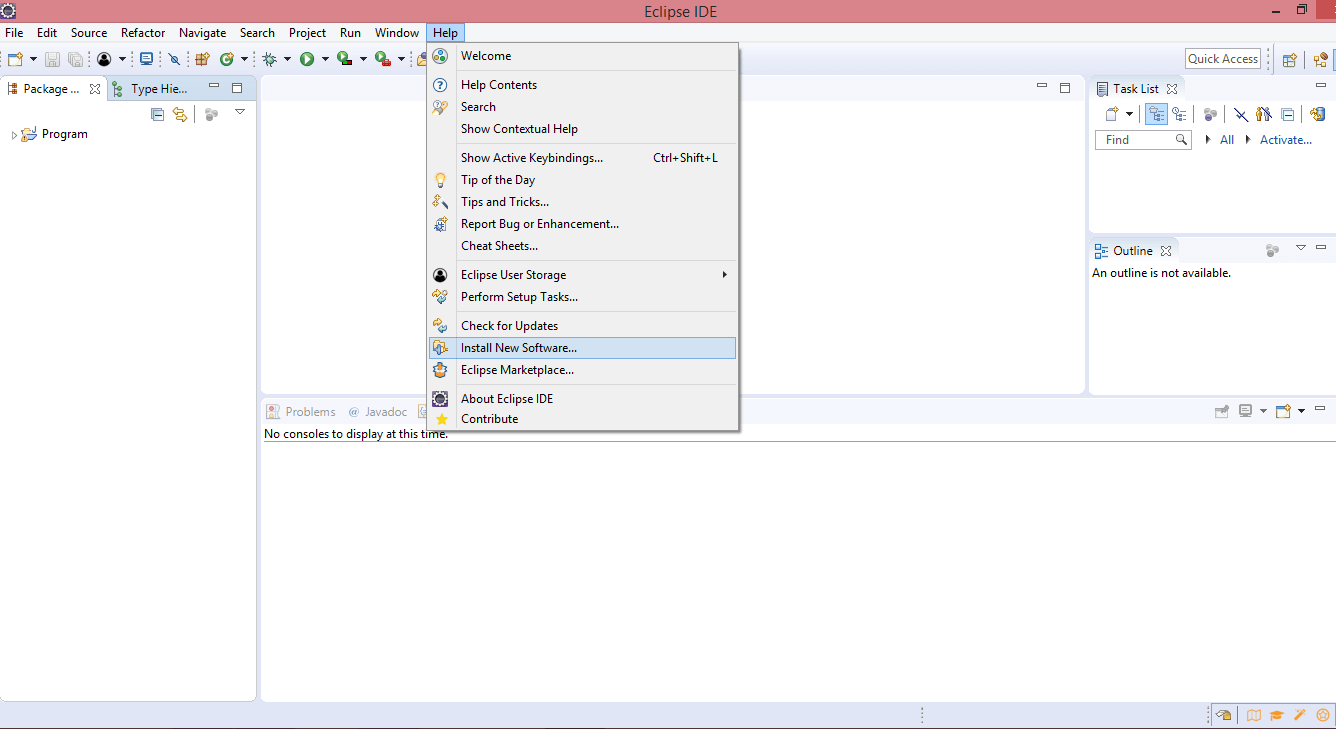


**Step 3:** Click on the Installation appearing on the top of the Table of Contents, and then click on the "**install the plug-in**". On clicking on the link "**install the plug in**", the screen appears as shown below:

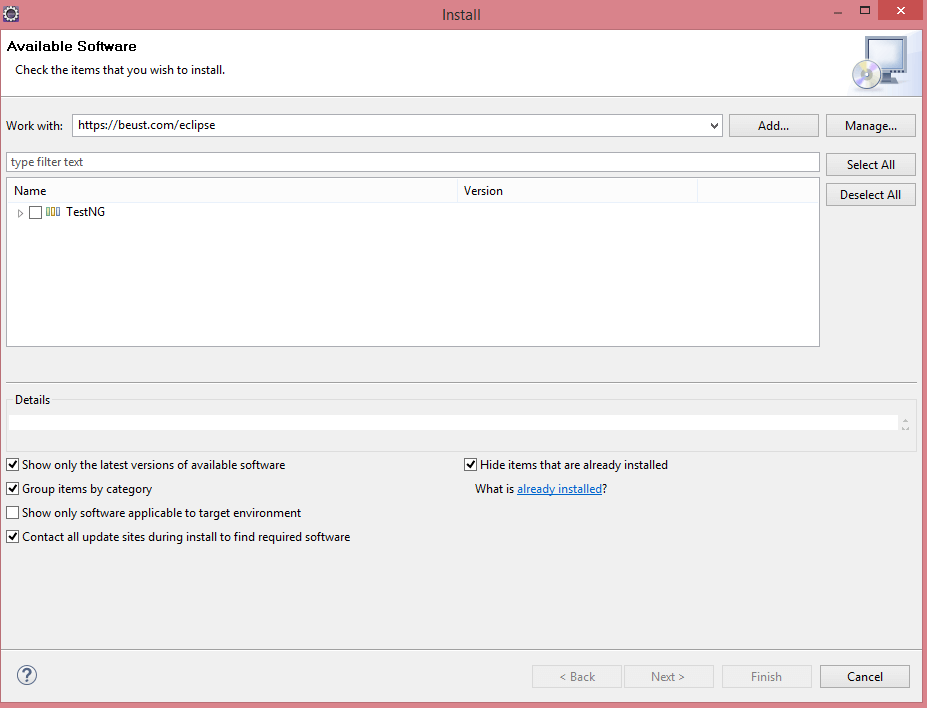


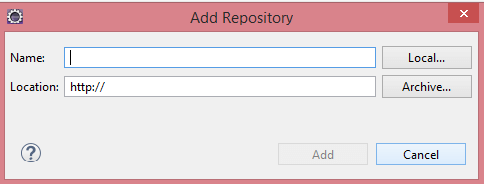
In the above screen, the URL is given <https://beust.com/eclipse> under the category "**Install from update site**". To install the TestNG plug-in in Eclipse, we need to add this given URL in Eclipse.

**Step 4:** Open the Eclipse. Click on the **Help** appearing on the menu bar and then click on the **Install New Software**.

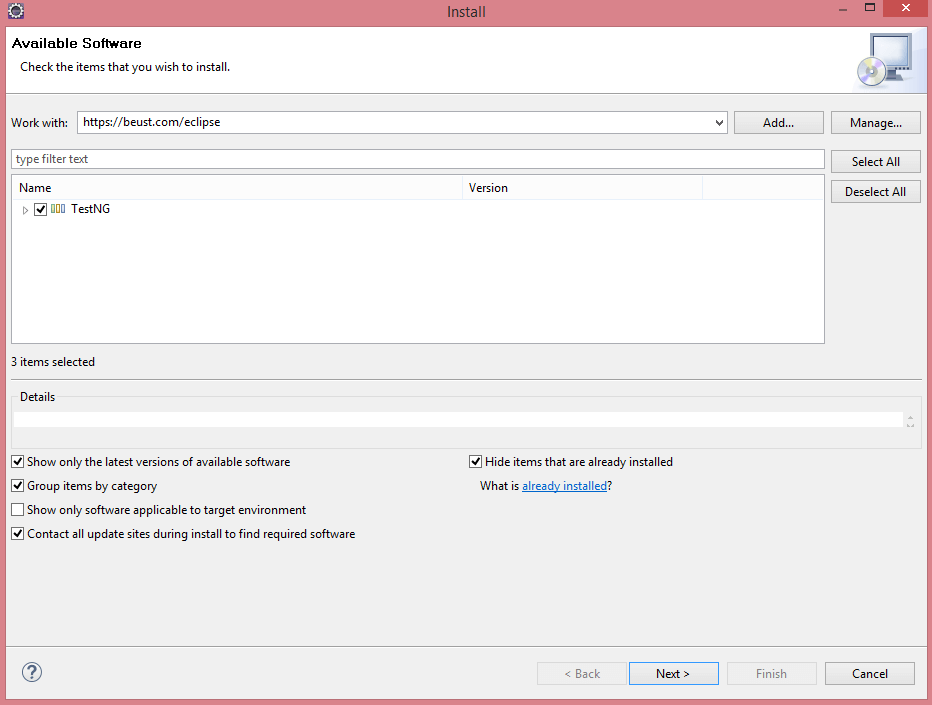


**Step 5:** Copy the URL <https://beust.com/eclipse>. Once you paste the URL, then press the Enter. However, in your case, you will see **Pending** for few seconds, thereafter you will see that TestNG plug-in has been loaded.

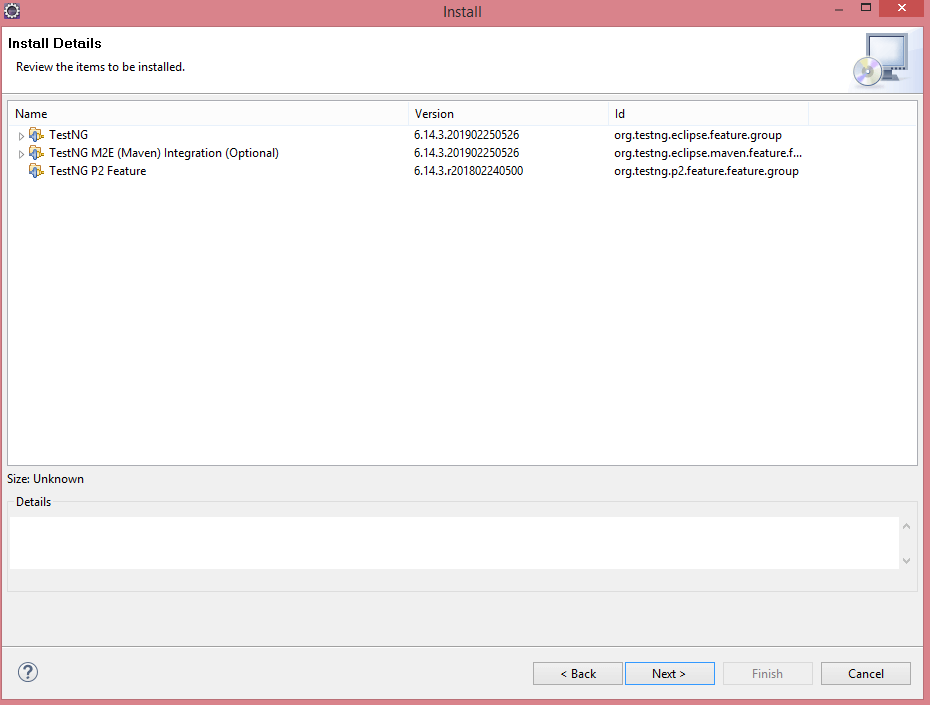




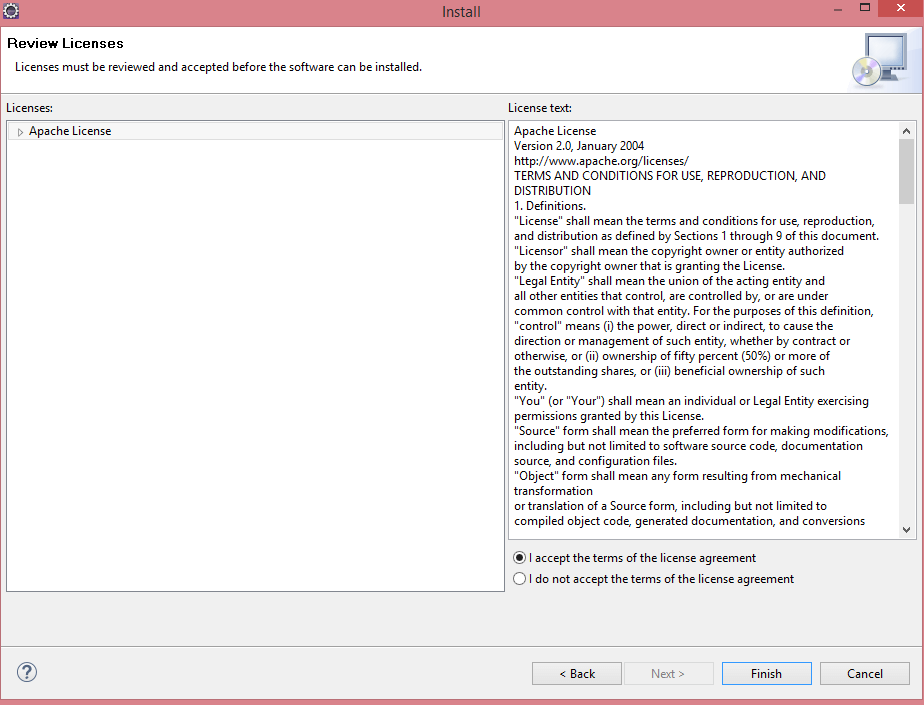
**Step 6:** Click on the TestNG checkbox.



**Step 7:** In the below screen, three dependencies of TestNG are shown. Now click on the Next.



**Step 8:** Accept the license and then click on the Finish.



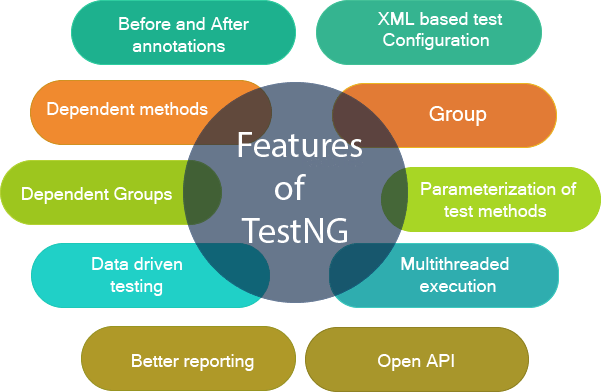
#### **Note: There are two ways of adding TestNG in your Eclipse. You can add TestNG either by installing the TestNG plug-in or adding the TestNG jar file in the Build Path. TestNG plug-in is more preferrable than testng.jar file as in case of a plug-in, you do not have to put in every project. However, in case of a testng.jar file, you need to add the jar file in a build path of every project.**

### **Download the TestNG jar file**

You can download the TestNG jar file from the link given below:

<https://mvnrepository.com/artifact/org.testng/testng/6.7>

# **Features of TestNG**



### **Multiple Before and After annotation options**

Before and after annotations are used to execute a certain set of code before and after executing the test methods. These annotations are used to set the variables or configuration before the start of the execution of test methods and clean up all the variables after the execution ends. Some of the Before and After annotations are @BeforeSuite, @BeforeTest, @BeforeGroups, @BeforeClass, etc.

### **XML-based test configuration**

Test suites in a Testng are mainly configured by using the XML-based file. Testng.xml file is used to organize and run the test suites. The testng.xml file is used to create the test suites by using classes, test methods, packages as well as by using the test groups. It is also used to pass the parameters to test classes or methods.

### **Dependent methods**

Dependency is a feature of Testng that allows a test method to depend on the single or group of test methods. Dependency works on the principle "depend-on-method" which must be either in the same class or in the inherited base class. This is the most important feature in TestNG that tells the TestNG to run the dependent test method after the execution of a given test method. You can also configure whether you want dependent test method should be executed or not even after the execution of the given test method fails.

### **Groups/group of groups**

TestNG groups allow you to group the test methods. By using TestNG groups, you can declare the methods in a group as well as you can declare the groups within a group. The Testng group can be used to include a certain set of groups and can exclude another set of groups.

### **Dependent groups**

Similar to the Dependent methods, test methods in a group can depend on the test methods of another group.

### **Parameterization of test methods**

One of the most important feature of TestNG is Parameterization. This feature allows you to pass the arguments as parameters and this achieved by using testng@Parameters annotation. We can pass the parameters to test methods in two ways, i.e., testng.xml file and DataProviders.

### **Data-driven testing**

TestNG allows users to perform data-driven testing. This testing allows users to execute the same test multiple times with multiple sets of data. To achieve the data-driven testing, DataProvider feature is used. DataProvider is a data feeder method that executes the test method with multiple sets of data.

### **Multithreaded execution**

Multithreaded execution is the parallel execution of tests. Multithreading means the execution of multiple parts of software at the same time. Based on the configuration in the XML file, multiple threads are started, and test methods are executed in them. Multithreaded execution saves a lot of execution time.

### **Better reporting**

Testng provides XML and HTML reports by default for test execution. You can even add your own custom reports when required.

### **Open API**

TestNG contains the open API means API is publicly available to the developers. This feature allows you to create your custom extensions in your framework when required.

# **What is TestNG Annotation?**

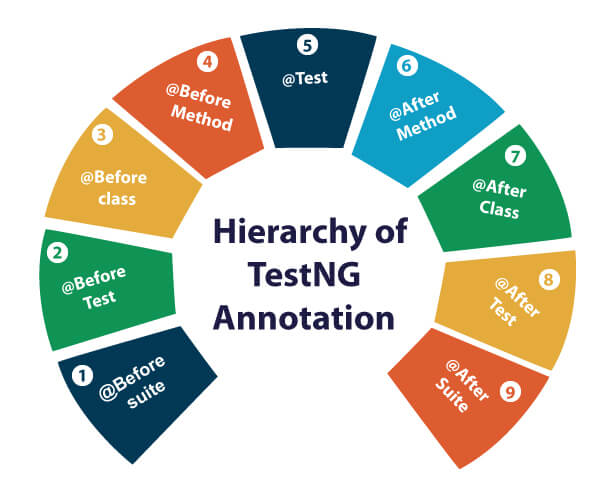
TestNG Annotation is a piece of code which is inserted inside a program or business logic used to control the flow of execution of test methods.

## List of TestNG Annotations



|  |  |
| --- | --- |
| **TestNG Annotation** | **Description** |
| [@BeforeSuite](https://www.javatpoint.com/testng-beforesuite-annotation) | The @BeforeSuite annotated method will run before the execution of all the test methods in the suite. |
| [@AfterSuite](https://www.javatpoint.com/testng-aftersuite-annotation) | The @AfterSuite annotated method will run after the execution of all the test methods in the suite. |
| [@BeforeTest](https://www.javatpoint.com/testng-beforetest-annotation) | The @BeforeTest annotated method will be executed before the execution of all the test methods of available classes belonging to that folder. |
| [@AfterTest](https://www.javatpoint.com/testng-aftertest-annotation) | The @AfterTest annotated method will be executed after the execution of all the test methods of available classes belonging to that folder. |
| [@BeforeClass](https://www.javatpoint.com/testng-beforeclass-annotation) | The @BeforeClass annotated method will be executed before the first method of the current class is invoked. |
| [@AfterClass](https://www.javatpoint.com/testng-afterclass-annotation) | The @AfterClass annotated method will be invoked after the execution of all the test methods of the current class. |
| [@BeforeMethod](https://www.javatpoint.com/testng-beforemethod-annotation) | The @BeforeMethod annotated method will be executed before each test method will run. |
| [@AfterMethod](https://www.javatpoint.com/testng-aftermethod-annotation) | The @AfterMethod annotated method will run after the execution of each test method. |
| [@BeforeGroups](https://www.javatpoint.com/testng-beforegroups-annotation) | The @BeforeGroups annotated method run only once for a group before the execution of all test cases belonging to that group. |
| [@AfterGroups](https://www.javatpoint.com/testng-aftergroups-annotation) | The @AfterGroups annotated method run only once for a group after the execution of all test cases belonging to that group. |

## Hierarchy of the TestNG Annotations:



* @BeforeSuite
* @BeforeTest
* @BeforeClass
* @BeforeMethod
* @Test
* @AfterMethod
* @AfterClass
* @AfterTest
* @AfterSuite

## Benefits of using TestNG Annotations:

* TestNG Annotations made the life of testers very easy. Based on your requirements, you can access the test methods, i.e., it has no predefined pattern or format.
* You can pass the additional parameters to TestNG annotations.
* In the case of TestNG annotations, you do not need to extend any test classes.
* TestNG Annotations are strongly typed, i.e., errors are detected at the compile time.