

Chapter 3:

TestNG Annotations

What Are TestNG Annotations

What Are TestNG Annotations? A TestNG Annotation is data that have a special meaning for a Java method. It provides information about how the annotation will control the execution order. I see TestNG Annotations as Pre-Conditions, Conditions, and Post-Conditions. Why? Because our Automation Test Scripts are very similar to Manual Test Cases. Sometimes with Manual Test Cases, there are Pre-Conditions that must be set up before we start our test. Our test is the Condition and after our test is the Post-Condition.

It's the same with automation, we have Pre-Conditions that must be set up before we test then we have our test. Our test is the Condition. Next is the Post-Condition that is performed after we complete our testing. Notice the @ symbol. In some places, you may hear @ symbol or @ sign. Both names are short for at a rate which is an accounting term. This symbol is placed in front of each TestNG Annotation. All of the Pre-Conditions begin with @Before while the Condition is @Test. @Test is a key annotation because it performs our test. Last, we have the Post-Conditions that all start with @After.

How To Add TestNG Annotations

How To Add TestNG Annotations? Let's go to Eclipse. There is more than 1 way to add an annotation. I believe most people start from scratch and write our annotation. For example, write @BeforeMethod then write our method. public void setup () and import the annotation. A short cut for importing and organizing our imports is CTRL + SHIFT + O. If you want to see a list of Eclipse shortcuts select CTRL + SHIFT + L and we see a lot of shortcuts including Organize Imports.

Another way to add TestNG Annotations is by selecting the package, right click, New then Other. Type TestNG, Select TestNG class then click Next. Do you see the Annotations? Let's select all of the annotations except for DataProvider. We are going to cover DataProvider in a subsequent chapter called Data Driven Testing. Data Driven Testing is when we run 1 Test Script with many sets of data. Change the Class Name to Configuration_Annotations and Click Finish.

Configuration Annotations

A Configuration Annotation is an annotation that begins with Before or After. They are called Configuration Annotations because the Before Annotations help us set up variables and configurations before starting test execution. The After Annotations help us clean up everything after executing our test. If we take a look at the TestNG Annotations package, we will see all of the annotations. The Configuration Annotations start at AfterClass and stop at BeforeTest.

In our Editor, the annotations are displayed in order from lowest to highest. Let's start with the highest level and add some print statements. BeforeSuite/AfterSuite run before a suite start and after all Test Methods. So, let's write BeforeSuite, Chrome - Set Up System Property. AfterSuite, Chrome – Clean Up All Cookies.

BeforeTest/AfterTest run before a test start and after all Test Methods. Now that we have set up Chrome, BeforeTest will Open Chrome and AfterTest will Close Chrome. BeforeClass/AfterClass run before a test class start and after all Test Methods. After opening Chrome, BeforeClass will Open Test Application and AfterClass will Close Test Application.

Next, we have BeforeMethod/AfterMethod which run before a Test Method and after a Test Method. Now that we finally opened the Test Application, let's Sign In and Sign Out. Our test will Search For Customer so change the name to searchCustomer. We can add as many test annotations as we want but let's add 1 more to Search For Product and change the name to searchProduct.

The annotations can be placed in any order on the editor because TestNG identifies the methods by looking up the annotation. For example, we can place BeforeSuite anywhere on this editor and it will always execute first. Next, we will see the order of how these annotations execute.

TestNG Annotations Execution Flow

The execution flow depends on our annotations. Therefore, the methods execute according to the rank of each annotation. Let's go to Eclipse and change the annotation order.

We will change the order on the Editor to match the order in how they will show up on the Console. BeforeSuite, BeforeTest, BeforeClass, BeforeMethod, first Test Method Search Customer, second Test Method Search Product, AfterMethod, AfterClass, AfterTest, and AfterSuite. Let's Run.

We see BeforeSuite / Chrome – Set Up System Property, then we see BeforeTest / Open Chrome, next is BeforeClass / Open Test Application, and we have BeforeMethod / which is Sign In. Here's the interesting part, the BeforeMethod always run before the Test Method. In this case, the first Test Method is Search For Customer which has an @Test Annotation and the AfterMethod always run after the Test Method. The AfterMethod is Sign Out.

We finished searching for a customer. However, Chrome is still set up, Chrome is still open, and the application remains open but we are not signed into the application. Now we start over with the BeforeMethod, which is a Pre-Condition to Sign In before Searching For A Product then the Post-Condition, AfterMethod Sign Out of the application. This shows us how TestNG executes the Pre-Condition, the Condition, and the Post-Condition based on our annotations.

Next, we have the AfterClass / Close Test Application, AfterTest / Close Chrome, and AfterSuite – Clean Up All Cookies. In the Console, we see both searchCustomer and searchProduct PASSED. The TestNG results tab also shows both tests PASSED.