TEST AUTOMATION UNIVERSITY
Rex Jones II

Chapter 4:
TestNG XML File

View xml File To See TestNG Suite, Test, Class, & Methods

1 Test

View xml file. The purpose of an xml file is to store data and to carry data for all of our testing. We see a tag for suite, a tag for test, a tag for class, and a tag for methods. A Suite can have 1 or more tests. A Test can have 1 or more classes and a Class can have 1 or more methods. The xml file gives us a picture on why the annotations execute in a particular order.

Let's Run our Test Suite. We see the same output. It's the same output because this execution has 1 test tag. We see BeforeSuite, BeforeTest, BeforeClass, BeforeMethod executes before both Test Methods – Search For Customer and Search For Product, then the AfterMethod executes after both Test Methods.

Next is AfterClass, AfterTest, and AfterSuite. Do you see how the execution order is consistent with the xml file? Let's look at an xml file that has 2 test tags.

2 Tests

This xml file does not include the Test Methods together. We see the 1st Test Name is Search For A Customer and the 2nd Test Name is Search For A Product. Let's run. The output is different. This time our execution does not immediately Sign back into the application after Signing out of the application. Now, the execution processes AfterClass – Close Test Application and AfterTest – Close Chrome after signing out. Why? Because in the xml file, the Test Methods are not in included the same class. Therefore, execution will not sign back in to start our next test. Notice, Chrome is still set up after executing BeforeSuite. That means the AfterSuite annotation is the only annotation that have not been executed. That completes searching for a customer.

Next is Search For A Product which starts with BeforeTest – Open Chrome, BeforeClass – Open Test Application, then BeforeMethod – Sign In and Search For A Product. Finally, AfterMethod – Signs Out, AfterClass – Close Test Application, AfterTest – Close Chrome, now AfterSuite – Clean Up All Cookies from Chrome.

In reality, we probably would not use 4 Before Configuration Annotations and 4 After Configuration Annotations. However, I wanted you to see how the xml file is connected to the annotations and how the xml file allows us to execute the same methods in 1 test or more than 1 test.

Which Annotation(s) Do We Choose For Testing

Which annotations do we choose for testing? The annotations we choose for testing depends on our Test Requirements. Do we need our test to set up a Pre-Condition before every Test Method and clean up with a Post-Condition after every Test Method or do we need 1 Pre-Condition before all of the Test

Methods and 1 Post-Condition after all of the Test Methods? Let's walkthrough our Test Application then I will show you the difference using 2 pairs of Configuration Annotations and 3 Test Methods.

The Test Requirement is to sign in with Username Admin and Password admin123 then click the Login button. After signing in, we click the Admin tab then search for a user. Finally, we sign out.

Our code uses a BeforeMethod annotation and AfterMethod annotation. First, we setup the test, then sign into the application, after signing into the application, we search for a user and sign out. After signing out we teardown our test which is close the browser.

Let's run and see what happens. We see 2 failures in the Console. The results tab also shows 2 Failures. Go back to the Console and look at our first Test signIn. Step 1 shows we opened Chrome and the application, Step 2 shows we signed into the application, but Steps 3 and 4 are missing before Step 5 close Chrome and the application. It passed but did not perform the steps we expected.

Let's look at the next test userSearch. Step 1 Open Chrome & the application. Steps 2, 3, and 4 are missing but we see Step 5 which Close Chrome and the application. Why did the Test Script skip steps 2 through 4? It skipped those steps because we cannot open Chrome and the application then search for a user. We cannot search for a user because we have not signed into the application. The console shows Failed: userSearch. Unable to locate element. It could not locate the Admin tab to begin searching for a user.

Same with the userSignOut. Our Test Script could not sign out of the application because it never signed into the application. Steps 2, 3, and 4 are missing. If we scroll down, the console shows FAILED: userSignOut. With this Test Requirement, we must use BeforeClass/AfterClass or BeforeTest/AfterTest. So, our code does not open the application before every Test Method and close the application after every Test Method.

Now watch what happens when we use the same code but change the Configuration Annotations to BeforeClass and AfterClass. Let's Run. All 3 Test Scripts PASSED. The results tab shows all 3 PASSED. The Console shows Step 1, we opened Chrome and the application, Step 2, we signed into the application, Step 3, we searched for a user, Step 4 we signed out of the application, and Step 5, we closed Chrome and the application.

TestNG provides a lot of annotations for our Test Requirements. This is a list of the annotations. Next, we will see how the Test Methods execute using a priority attribute.