

Grouping of Tests

In this topic, we will see how to group tests.

WE'LL COVER THE FOLLOWING ^

- What is the grouping of tests?
- How to group tests?

What is the grouping of tests?

TestNG allows to group related tests using tags. We can also have groups depending on other groups. TestNG allows having configuration methods that can run before starting or ending of tests belonging to a group. We can invoke tests belonging to a certain group or regular expressions too, while excluded from another group.

How to group tests?

Each test can belong to one or more groups. This grouping can be done to test methods or test classes. When giving at both places, the one given at the test method takes precedence. This becomes ***Partial test grouping***, as all the test methods in the class belong to a group, and this method belongs to another.

```
@Test(groups = { "functional" })
public class SampleTestClass {

    @Test(groups = { "sanity", "functional" })
    public void testA() {
    }

    @Test(groups = { "sanity" })
    public void testB() {
    }
}
```

```
public void testC() {  
}  
  
@Test(dependsOnGroups = { "sanity" } )  
public void testD() {  
}  
}
```

Here, in the above example, `testA` belongs to both `sanity` and `functional` groups. `testB` belongs to the `sanity` group only. `testC` belongs to the `functional` group only because there is no group explicitly annotated over this test method and therefore inherits from the class level `@Test`.

`testD` will not start until all the test methods belonging to the group `sanity` are completed, and `org.testng.TestNGException` will be thrown if no tests belonging to `sanity` were run before `testD`.

Now that you are familiar with grouping the tests, in the next lesson, you will learn about `@DataProvider` and `@Factory` annotations.