What are TestNG Groups?

TestNG Groups allow us to categorize test methods and run specific groups of tests instead of executing all tests in a class or suite. This is useful for executing tests selectively, such as running only **smoke tests**, **regression tests**, or **integration tests**.

Why Use TestNG Groups?

- Run only relevant test cases.
- Avoid executing unnecessary tests.
- Easily organize and manage large test suites.
- Helps in CI/CD pipelines where specific tests need to run based on changes.

How to Use Groups in TestNG?

Groups are defined using the `groups` attribute in the `@Test` annotation.

Example: Defining Groups in Test Methods

```
import org.testng.annotations.Test;

public class TestGroupsExample {

    @Test(groups = {"smoke"})
    public void smokeTest() {
        System.out.println("Executing Smoke Test");
    }

    @Test(groups = {"regression"})
    public void regressionTest() {
        System.out.println("Executing Regression Test");
    }

    @Test(groups = {"sanity"})
    public void sanityTest() {
        System.out.println("Executing Sanity Test");
    }

    @Test(groups = {"smoke", "regression"})
    public void combinedTest() {
        System.out.println("Executing Test in Smoke & Regression Groups");
    }
}
```

Executing Groups in TestNG XML

TestNG XML allows selecting which test groups should be executed.

Example: Running Only Smoke Tests

```
<suite name="TestSuite">
    <test name="SmokeTestExecution">
        <groups>
            <run>
                <include name="smoke"/>
            </run>
        </groups>
        <classes>
            <class name="testpackage.TestGroupsExample"/>
        </classes>
    </test>
</suite>
```

Example: Running Multiple Groups

```
<suite name="TestSuite">
    <test name="Run Multiple Groups">
        <groups>
            <run>
                <include name="smoke"/>
                <include name="sanity"/>
            </run>
        </groups>
        <classes>
            <class name="testpackage.TestGroupsExample"/>
        </classes>
    </test>
</suite>
```

Excluding Test Groups

We can also exclude groups from execution.

Example: Running All Tests Except Regression

```
<suite name="TestSuite">
    <test name="Exclude Regression">
        <groups>
            <run>
                <exclude name="regression"/>
            </run>
        </groups>
        <classes>
            <class name="testpackage.TestGroupsExample"/>
        </classes>
    </test>
</suite>
```

Grouping Multiple Classes

We can define groups that apply to multiple classes.

```
<suite name="TestSuite">
    <test name="Multiple Classes Test">
        <groups>
            <run>
                <include name="smoke"/>
            </run>
        </groups>
        <classes>
            <class name="testpackage.ClassOne"/>
            <class name="testpackage.ClassTwo"/>
        </classes>
    </test>
</suite>
```

Using Groups with Dependencies

We can use the `dependsOnGroups` attribute to ensure some tests run only after a specific group has completed.

Example: Running Regression Tests After Smoke Tests

```
import org.testng.annotations.Test;

public class DependencyExample {

    @Test(groups = "smoke")
    public void smokeTest() {
        System.out.println("Executing Smoke Test");
    }

    @Test(groups = "regression", dependsOnGroups = "smoke")
    public void regressionTest() {
        System.out.println("Executing Regression Test After Smoke Test");
    }
}
```

Best Practices for Using TestNG Groups

- Use meaningful names for test groups (`smoke`, `regression`, `sanity`, etc.).
- Avoid overlapping groups unless necessary.
- Utilize `dependsOnGroups` to maintain execution order.
- Use XML to dynamically include/exclude groups.