**Selenium with TestNG**

**TestNG is a unit level framework for selenium webdriver**

**What is TestNG:**

TestNG is testing framework inspired from Junit and Nunit. Introducing new functionality that make it more powerful and easier to use.

**Why TestNG:**

1. Easy annotations
2. Test cases grouped easily
3. Data parameterization is possible
4. Easy test configuration
5. Parallel testing is possible
6. Produce multiple reports

**TestNG Annotations:**

|  |  |
| --- | --- |
| **@BeforeTest** | **@AfterTest** |
| @BeforeClass | @AfterClass |
| @BeforeMethod | @AfterMethod |
| @BeforeGroup | @AfterGroup |
| @BeforeSuite | @AfterSuite |
| @Test |  |

**Test Case writing:**

Step 1 – Write the business logic of the test

Step 2 – Insert TestNG annotation in the code

Step 3 – Add the information about your test in a testng.xml file

Step 4 – Run TestNG

**TestNG Reports:**

HTML Report

Emailable Report

Index.html Report

XML Report

**Sample Script:**

**public** **class** TestNg\_Class1 {

@BeforeClass

**public** **void** start(){

System.***out***.println("Start Sever and connections");

}

@BeforeMethod

**public** **void** login(){

System.***out***.println("Login to the application");

}

@Test(priority =1)

**public** **void** UpdateFlight() {

System.***out***.println("Update Flight");

}

@Test(priority =2)

**public** **void** DeleteFlight() {

System.***out***.println("Delete Flight");

}

@AfterMethod

**public** **void** logout(){

System.***out***.println("Logout from the application");

}

@AfterClass

**public** **void** stop(){

System.***out***.println("stop Sever and connections");

}

}

**Note :** In TestNG the @Test Annotation executes in alphabetical order.

**Assertions in TestNG:**

1. *assertEqual(String actual,String expected)*
2. *assertEqual(String actual,String expected, String message)*
3. *Assert.assertTrue(condition, message)*
4. *Assert.assertFalse(condition, message)*