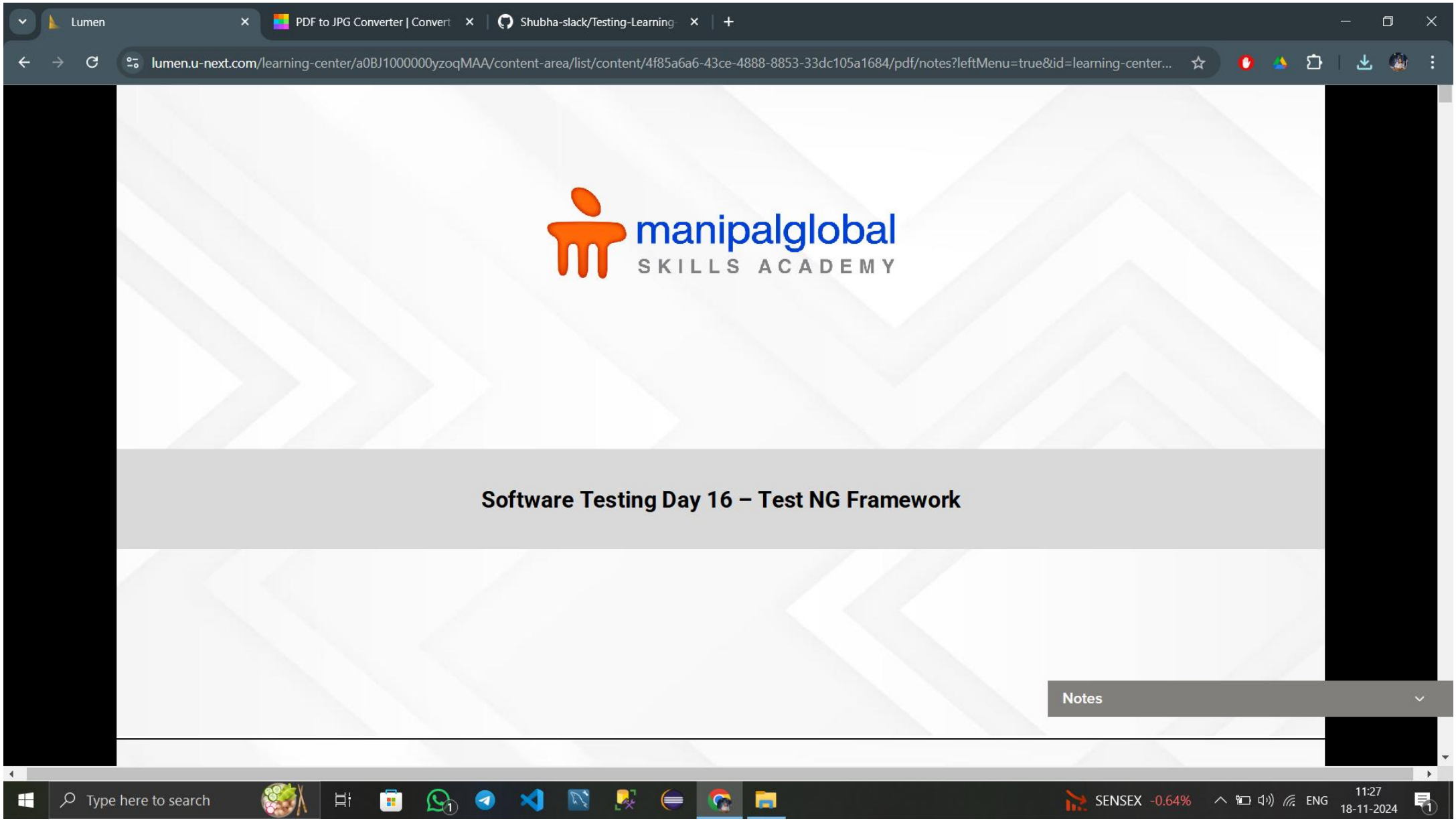
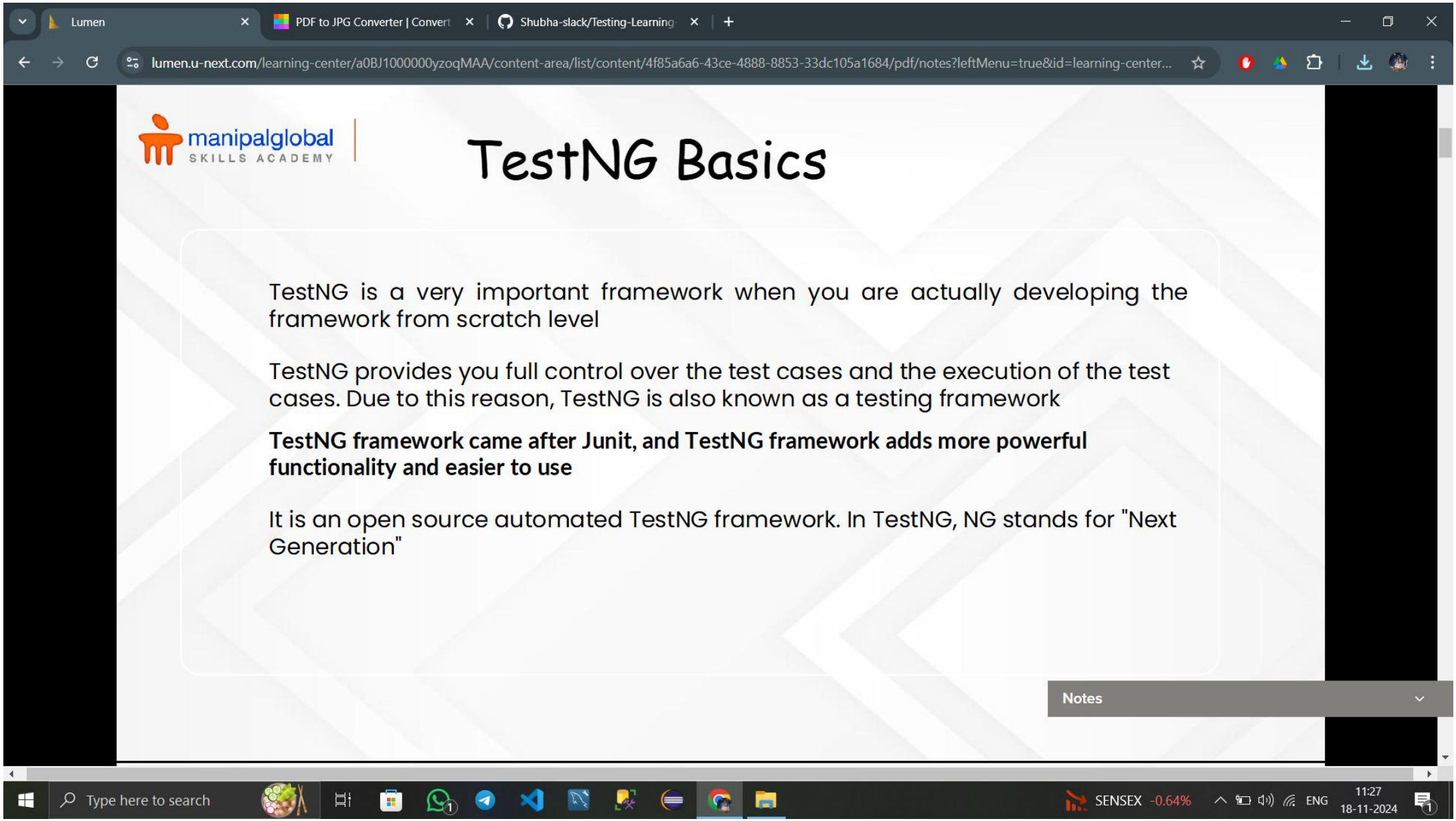

Lumen
PDF to JPG Converter | Convert
Shubha-slack/Testing-Learning
lumen.u-next.com/learning-center/a0BJ1000000yzoqMAA/content-area/list/content/4f85a6a6-43ce-4888-8853-33dc105a1684/pdf/notes?leftMenu=true&id=learning-center...
manipalglobal
SKILLS ACADEMY
Software Testing Day 16 – Test NG Framework
Notes
Type here to search
SENSEX -0.64%
ENG
11:27
18-11-2024

manipalglobal
SKILLS ACADEMY

Agenda

- Test NG Framework
- Features of TEST NG Framework
- Advantages of using TEST NG
- Disadvantages of using TEST NG
- Installation and set up
- Running of the automation scripts using testing
- Reporting capability

Notes


Lumen
PDF to JPG Converter | Convert
Shubha-slack/Testing-Learning
lumen.u-next.com/learning-center/a0BJ1000000yzoqMAA/content-area/list/content/4f85a6a6-43ce-4888-8853-33dc105a1684/pdf/notes?leftMenu=true&id=learning-center...
manipalglobal
SKILLS ACADEMY
TestNG Basics
TestNG is a very important framework when you are actually developing the framework from scratch level
TestNG provides you full control over the test cases and the execution of the test cases. Due to this reason, TestNG is also known as a testing framework
TestNG framework came after Junit, and TestNG framework adds more powerful functionality and easier to use
It is an open source automated TestNG framework. In TestNG, NG stands for "Next Generation"
Notes
Type here to search
SENSEX -0.64%
ENG
11:27
18-11-2024

manipalglobal SKILLS ACADEMY

Advantages Of TestNG

Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed, and the number of test cases skipped

Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which test case should be executed first

The same test case can be executed multiple times without loops just by using keyword called 'invocation count.

The TestNG framework can be easily integrated with tools like Maven, Jenkins, etc.

Parallel testing is possible

manipalglobal SKILLS ACADEMY

TestNG Annotations

@BeforeMethod: This will be executed before every @test annotated method

@AfterMethod: This will be executed after every @test annotated method

@BeforeClass: This will be executed before first @Test method execution. It will be executed one only time throughout the test case

@AfterClass: This will be executed after all test methods in the current class have been run

@BeforeTest: This will be executed before the first @Test annotated method. It can be executed multiple times before the test case

manipalglobal SKILLS ACADEMY

TestNG Annotations

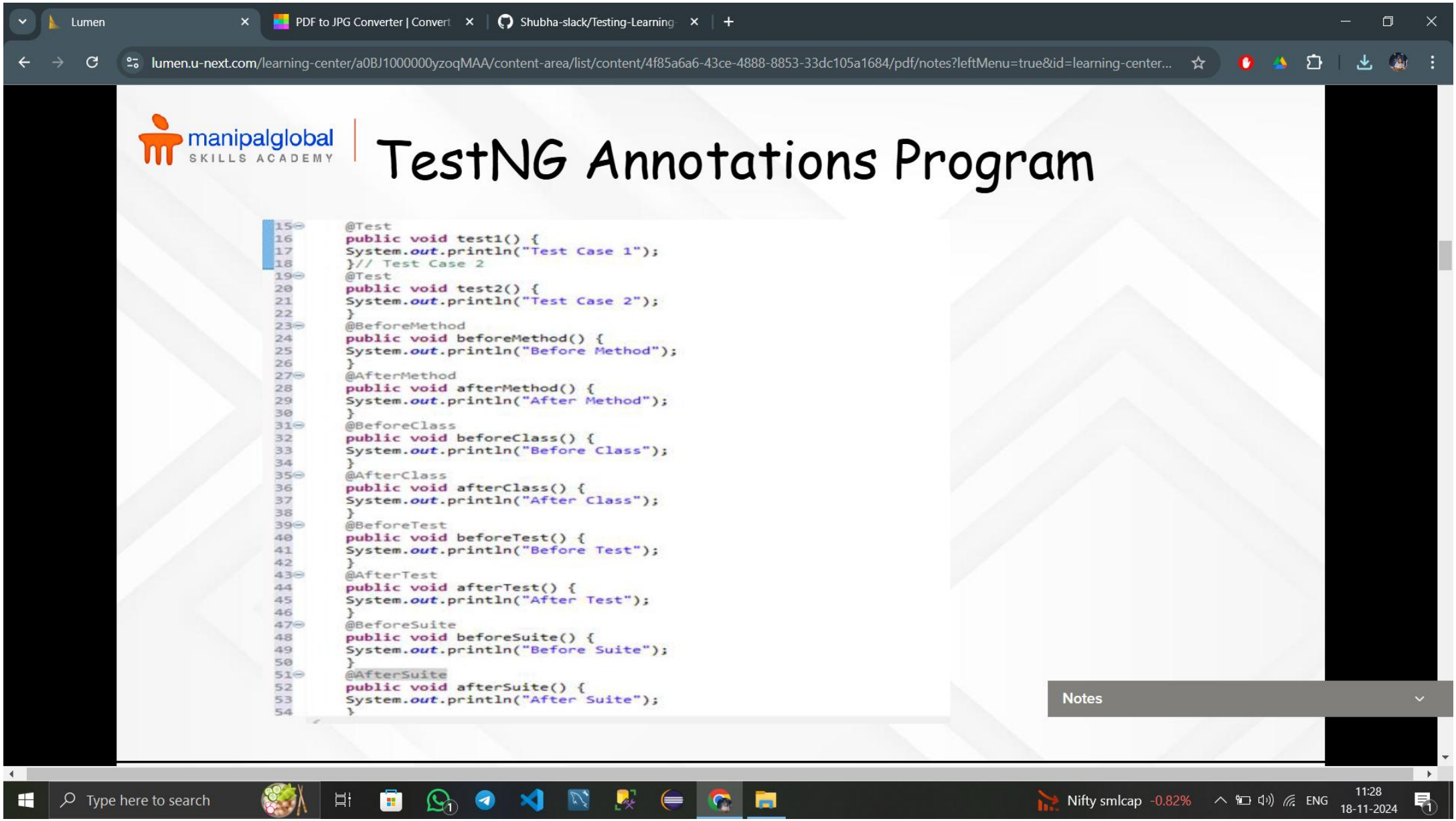
@AfterTest: A method with this annotation will be executed when all @Test annotated methods complete the execution of those classes inside the <test> tag in the TestNG.xml file

@BeforeSuite: It will run only once, before all tests in the suite are executed

@AfterSuite: A method with this annotation will run once after the execution of all tests in the suite is complete

@BeforeGroups: This method will run before the first test run of that specific group

@AfterGroups: This method will run after all test methods of that group complete their execution


Lumen
PDF to JPG Converter | Convert
Shubha-slack/Testing-Learning
lumen.u-next.com/learning-center/a0BJ1000000yzoqMAA/content-area/list/content/4f85a6a6-43ce-4888-8853-33dc105a1684/pdf/notes?leftMenu=true&id=learning-center...
manipalglobal
SKILLS ACADEMY
TestNG Annotations Program
@Test
public void test1() {
System.out.println("Test Case 1");
}// Test Case 2
@Test
public void test2() {
System.out.println("Test Case 2");
}
@BeforeMethod
public void beforeMethod() {
System.out.println("Before Method");
}
@AfterMethod
public void afterMethod() {
System.out.println("After Method");
}
@BeforeClass
public void beforeClass() {
System.out.println("Before Class");
}
@AfterClass
public void afterClass() {
System.out.println("After Class");
}
@BeforeTest
public void beforeTest() {
System.out.println("Before Test");
}
@AfterTest
public void afterTest() {
System.out.println("After Test");
}
@BeforeSuite
public void beforeSuite() {
System.out.println("Before Suite");
}
@AfterSuite
public void afterSuite() {
System.out.println("After Suite");
}
Notes
Type here to search
Nifty smlcap -0.82%
ENG
11:28
18-11-2024

manipalglobal SKILLS ACADEMY

TestNG Annotations output

```
[RemoteTestNG] detected TestNG version 7.2.0
Before Suite
Before Test
Before Class
Before Method
Test Case 1
After Method
Before Method
Test Case 2
After Method
After Class
After Test
PASSED: test1
PASSED: test2
===============================================
Default test
Tests run: 2, Failures: 0, Skips: 0
===============================================
After Suite
===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

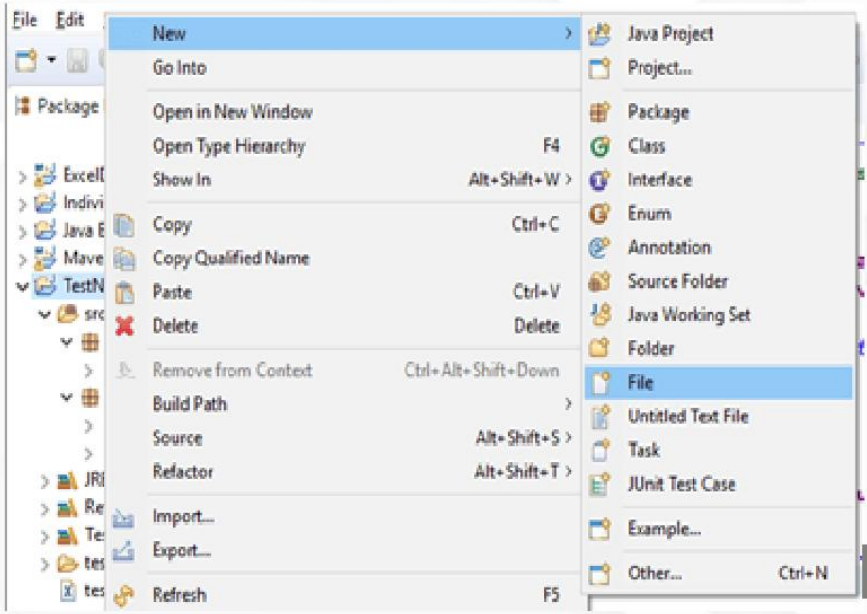
Lumen | PDF to JPG Converter | Convert | Shubha-slack/Testing-Learning

lumen.u-next.com/learning-center/a0BJ1000000yzoqMAA/content-area/list/content/4f85a6a6-43ce-4888-8853-33dc105a1684/pdf/notes?leftMenu=true&id=learning-center...

manipalglobal SKILLS ACADEMY

Creating TestNG xml file

Go to Project right Click New File



Notes

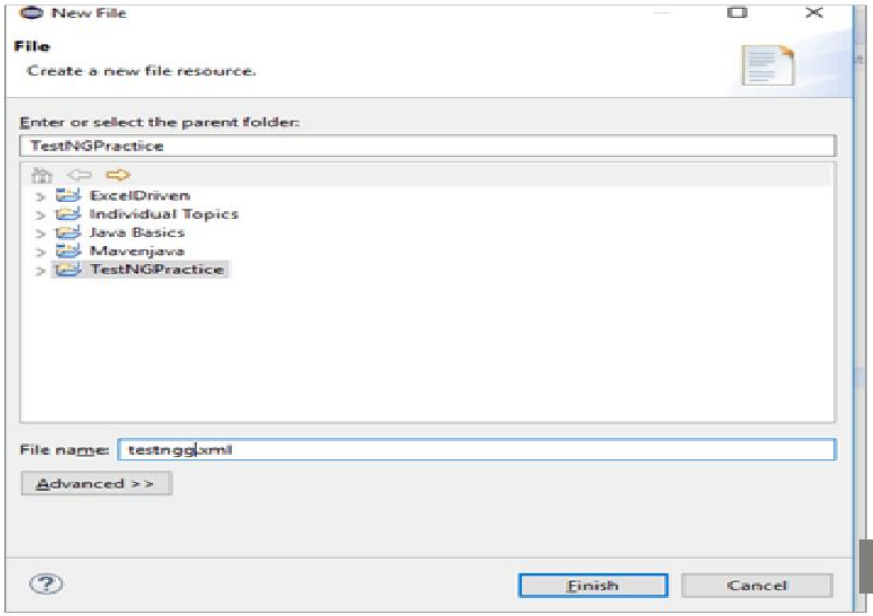
Type here to search

Nifty smlcap -0.82%

ENG 11:28 18-11-2024

Creating TestNG xml file

Add the file name as 'testng.xml' as shown in the below image and click on the Finish button

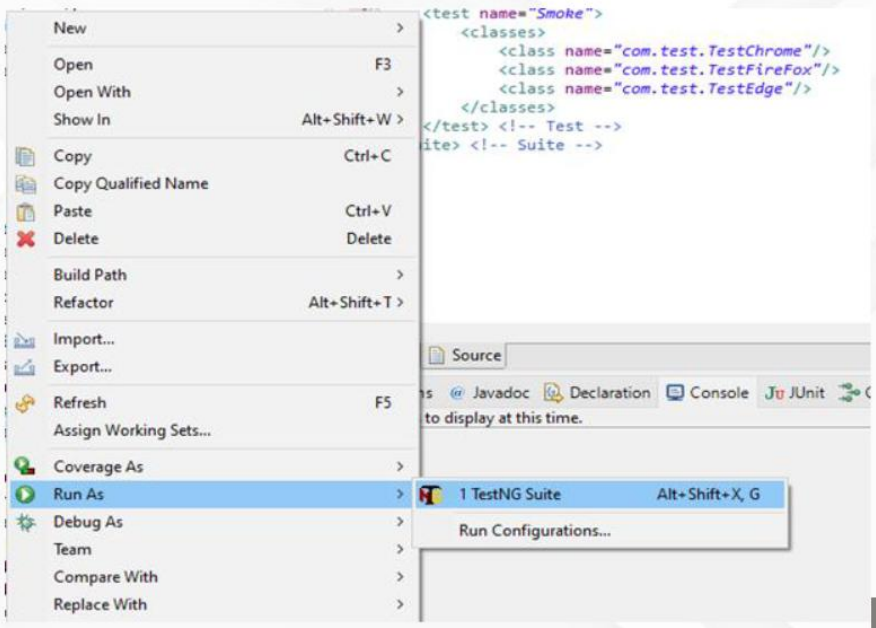


Notes

manipalglobal SKILLS ACADEMY

Executing TestNG xml file

Testng.xml right click Run As TestNG Suite

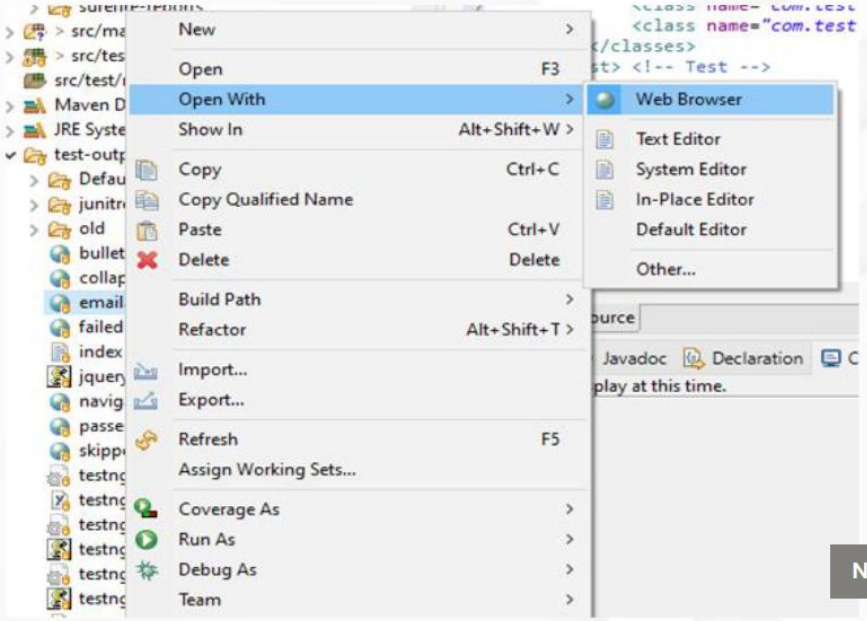


Notes

manipalglobal SKILLS ACADEMY

TestNG Reports

When we execute testng.xml file, and refresh the project. We will get test-output folder in that folder for reporting in TestNG



Notes

manipalglobal SKILLS ACADEMY

TestNG Reports

When we execute testng.xml file, and refresh the project. We will get test-output folder in that folder for reporting in TestNG

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	3	0	0	0	15,693		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
com.test.TestChrome	test	1608379389089	15591
com.test.TestFirefox	test	1608379389089	15591
com.test.TestEdge	test	1608379389089	15591

Notes

manipalglobal SKILLS ACADEMY

Depends On

In inherited dependent test methods in TestNG, we create dependency among the methods that belong to different classes, and one of the classes inherits the functionalities of another class

In the code snippet, LogIn method will execute only if the OpenBrowser test method is successfully completed

```java
package src.test;

import org.testng.annotations.Test;

public class SuperClass
{
    @Test
    public void OpenBrowser() {
        System.out.println("BrowserOpened");
    }
}
class InheritedDependencyTest extends SuperClass
{
    @Test(dependsOnMethods = { "OpenBrowser" })
    public void LogIn() {
        System.out.println("Logged In");
    }
}
```

Notes

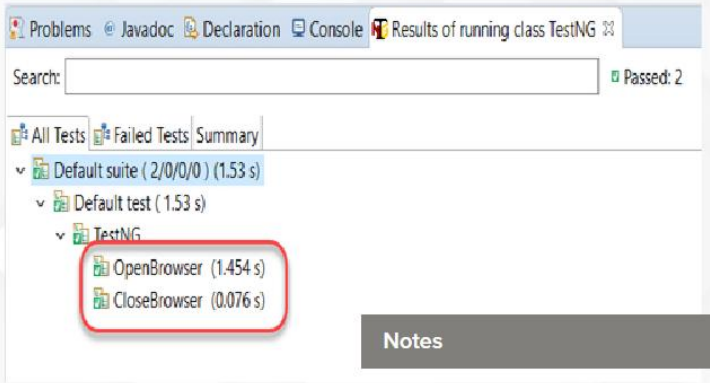
manipalglobal SKILLS ACADEMY

Priority in TestNG

Prioritization in TestNG is a way to provide a sequence to the methods so that they do not run out of order

Since alphabetically running test cases in TestNG have no logical sequence, providing priority to these test cases helps us managing our tests' execution

```
7  public class TestNG {
8      WebDriver driver = new ChromeDriver();
9
10     @Test (priority = 1)
11     public void CloseBrowser() {
12         driver.close();
13         System.out.println("Closing Google Chrome browser");
14     }
15
16     @Test (priority = 0)
17     public void OpenBrowser() {
18         System.out.println("Launching Google Chrome browser");
19         driver.get("https://www.demoqa.com");
20     }
21 }
22
```



Notes

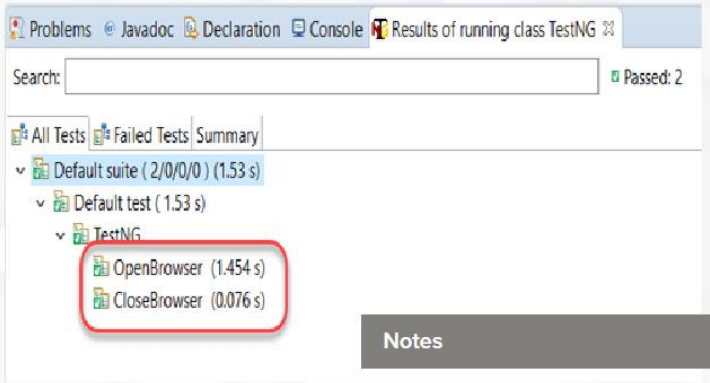
manipalglobal SKILLS ACADEMY

Priority in TestNG

Prioritization in TestNG is a way to provide a sequence to the methods so that they do not run out of order

Since alphabetically running test cases in TestNG have no logical sequence, providing priority to these test cases helps us managing our tests' execution

```
7  public class TestNG {
8      WebDriver driver = new ChromeDriver();
9
10     @Test (priority = 1)
11     public void CloseBrowser() {
12         driver.close();
13         System.out.println("Closing Google Chrome browser");
14     }
15
16     @Test (priority = 0)
17     public void OpenBrowser() {
18         System.out.println("Launching Google Chrome browser");
19         driver.get("https://www.demoqa.com");
20     }
21 }
22
```



Notes

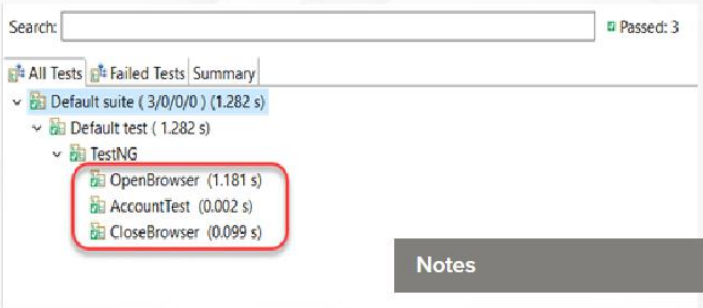
manipalglobal SKILLS ACADEMY

Priority in TestNG

Setting Priority 0, 1 and 1

```
7  public class TestNG {
8      WebDriver driver = new ChromeDriver();
9
10     @Test (priority = 1)
11     public void CloseBrowser() {
12         driver.close();
13         System.out.println("Closing Google Chrome browser");
14     }
15
16     @Test (priority = 0)
17     public void OpenBrowser() {
18         driver.get("https://www.demoqa.com");
19         System.out.println("Launching Google Chrome browser");
20     }
21
22     @Test (priority = 1)
23     public void AccountTest(){
24         System.out.println("Some tests for Customer Account");
25     }
26 }
27
```

If two or more methods have the same priorities in TestNG, then their running test sequence is alphabetic. Since "A" comes before "C", the method AccountTest ran first



Notes

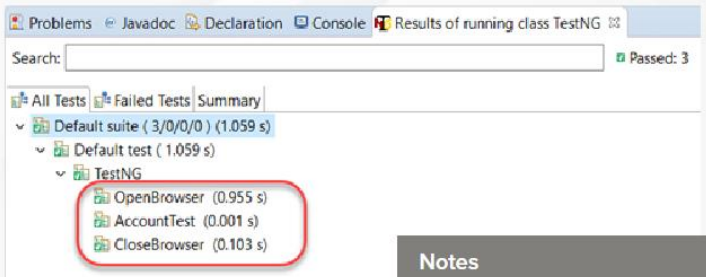
manipalglobal SKILLS ACADEMY

Priority in TestNG

Setting Priority 0, -1 and blank

```
7  public class TestNG {
8      WebDriver driver = new ChromeDriver();
9
10     @Test (priority = 0)
11     public void CloseBrowser() {
12         driver.close();
13         System.out.println("Closing Google Chrome browser");
14     }
15
16     @Test (priority = -1)
17     public void OpenBrowser() {
18         System.out.println("Launching Google Chrome browser");
19         driver.get("https://www.demoqa.com");
20     }
21
22     @Test
23     public void AccountTest(){
24         System.out.println("Some tests for Customer Account");
25     }
26 }
```

Observe that the AccountTest method ran before CloseBrowser even without having any priority because both sets to priority = 0, and hence, they run alphabetically



Notes

manipalglobal SKILLS ACADEMY

Groups in TestNG

We don't want to define test methods separately in different classes (depending upon functionality) and
At the same time want to ignore (not to execute) some test cases as if they does not exist in the code

So we can Group them. This is done by using "include" and "exclude" mechanism supported in testNG

Syntax : @Test (groups = { "SmokeTest", "RegressionTest" })

This xml snippet will only run the tests which are grouped as SmokeTest. Rest will be skipped

```
<groups>
    <run>
        <include name="SmokeTest" />
    </run>
</groups>
```

manipalglobal SKILLS ACADEMY

Groups in TestNG

In the below example, we took three classes having groups set as SmokeTest and RegressionTest which is quite a normal scenario

```java
package src.test;

import org.testng.annotations.Test;

public class Personal_loan {
    @Test(groups = { "SmokeTest" })
    public void WebLoginPersonalLoan() {
        System.out.println("Web Login Personal Loan");
    }

    @Test(groups = { "Regression" })
    public void MobileLoginPersonalLoan() {
        System.out.println("Mobile Login Personal Loan");
    }

    @Test(groups = { "Regression" })
    public void APILoginPersonalLoan() {
        System.out.println("API Login Personal Loan");
    }
}
```

```java
package src.test;

import org.testng.annotations.Test;

public class Home_loan {
    @Test(groups = { "Regression" })
    public void WebLoginHomeLoan() {
        System.out.println("Web Login Home Loan");
    }

    @Test(groups = { "SmokeTest" })
    public void MobileLoginHomeLoan() {
        System.out.println("Mobile Login Home Loan");
    }

    @Test(groups = { "Regression" })
    public void APILoginHomeLoan() {
        System.out.println("API Login Home Loan");
    }
}
```

```java
package src.test;

import org.testng.annotations.Test;

public class Car_loan {
    @Test(groups = { "Regression" })
    public void WebLoginCarLoan() {
        System.out.println("Web Login Car Loan");
    }

    @Test(groups = { "Regression" })
    public void MobileLoginCarLoan() {
        System.out.println("Mobile Login Car Loan");
    }

    @Test(groups = { "SmokeTest" })
    public void APILoginCarLoan() {
        Syste
    }
}
```

Notes

manipalglobal SKILLS ACADEMY

Groups in TestNG

As we include on SmokeTest in the groups tag in TestNG.xml file. Only tests for Smoke Test have executed which is shown in output

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd"> -->
3  <suite name="TestingApps">
4      <test name="Smoke">
5          <groups>
6              <run>
7                  <include name="SmokeTest" />
8              </run>
9          </groups>
10         <classes>
11             <class name="com.test.Personal_Loan" />
12             <class name="com.test.Home_loan" />
13             <class name="com.test.Car_Loan" />
14         </classes>
15     </test> <!-- Test -->
16 </suite> <!-- Suite -->
```

Markers | Properti... | Servers | Data So... | Snippets | Console | Results ...

```
<terminated> TestNGtutorial_testng.xml [TestNG] C:\Program Files\Java\jre-9.0.4\bin\javaw.exe (16-May-2019,
[TestNG] Running:
  D:\TestNGtutorial\testng.xml

Web Login Personal Loan
Mobile Login Home Loan
API Login Home Loan

===============================================
test_suite
Total tests run: 3, Failures: 0, Skips: 0
===============================================
```

Notes

manipalglobal SKILLS ACADEMY

DataProviders in TestNG

The DataProviders in TestNG are another way to pass the parameters in the test function, the other one being TestNG parameters

DataProviders pass different values to the TestNG Test Case in a single execution and in the form of TestNG Annotations. It is a part of the inbuilt TestNG data-driven testing for which TestNG is quite popular

@DataProvider annotation helps us write data-driven test cases.

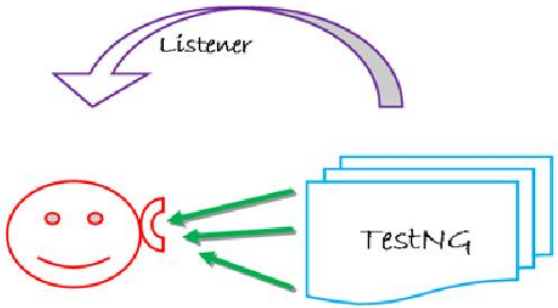
Syntax of DataProvider is as follows

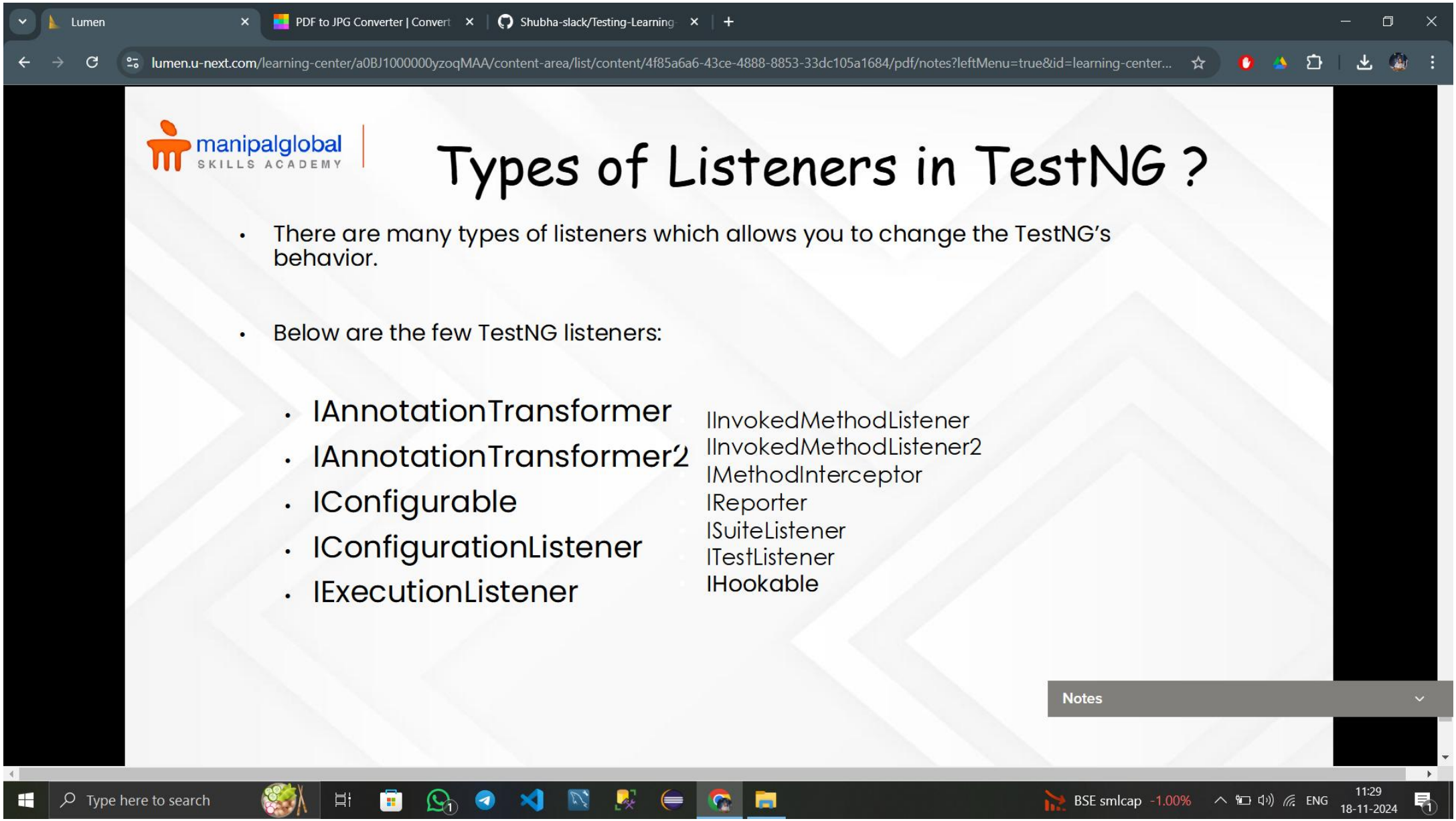
```
@DataProvider (name = "name_of_dataprovider")
public Object[][] dpMethod() {
    return new Object [][] { values}
}
```

What is Listeners in TestNG?

manipalglobal SKILLING SYSTEM

- Listener is defined as interface that modifies the default TestNG's behavior.
- As the name suggests Listeners "listen" to the event defined in the selenium script and behave accordingly.
- It is used in selenium by implementing Listeners Interface.
- It allows customizing TestNG reports or logs. There are many types of Te




Lumen
PDF to JPG Converter | Convert
Shubha-slack/Testing-Learning
lumen.u-next.com/learning-center/a0BJ1000000yzoqMAA/content-area/list/content/4f85a6a6-43ce-4888-8853-33dc105a1684/pdf/notes?leftMenu=true&id=learning-center...
manipalglobal
SKILLS ACADEMY
Types of Listeners in TestNG ?
There are many types of listeners which allows you to change the TestNG's behavior.
Below are the few TestNG listeners:
IAnnotationTransformer
IAnnotationTransformer2
IConfigurable
IConfigurationListener
IExecutionListener
IInvokedMethodListener
IInvokedMethodListener2
IMethodInterceptor
IReporter
ISuiteListener
ITestListener
IHookable
Notes
Type here to search
BSE smlcap -1.00%
ENG
11:29
18-11-2024

manipalglobal SKILL ACADEMY

ITestListener Methods

- OnStart- OnStart method is called when any Test starts.
- onTestSuccess- onTestSuccess method is called on the success of any Test.
- onTestFailure- onTestFailure method is called on the failure of any Test.
- onTestSkipped- onTestSkipped method is called on skipped of any Test.
- onTestFailedButWithinSuccessPercentage- method is called each time Test fails but is within success percentage.
- onFinish- onFinish method is called after all Tests are executed.