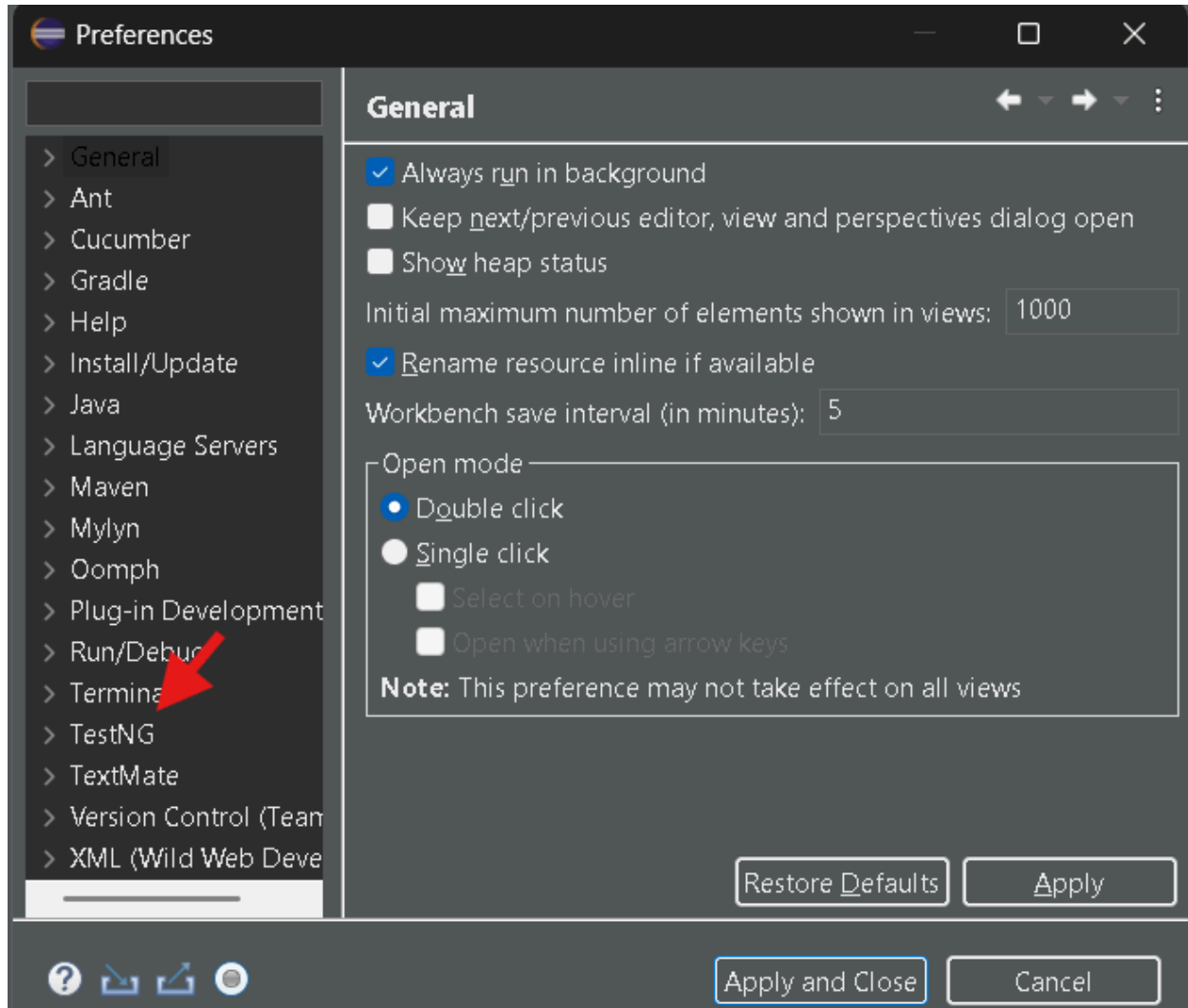
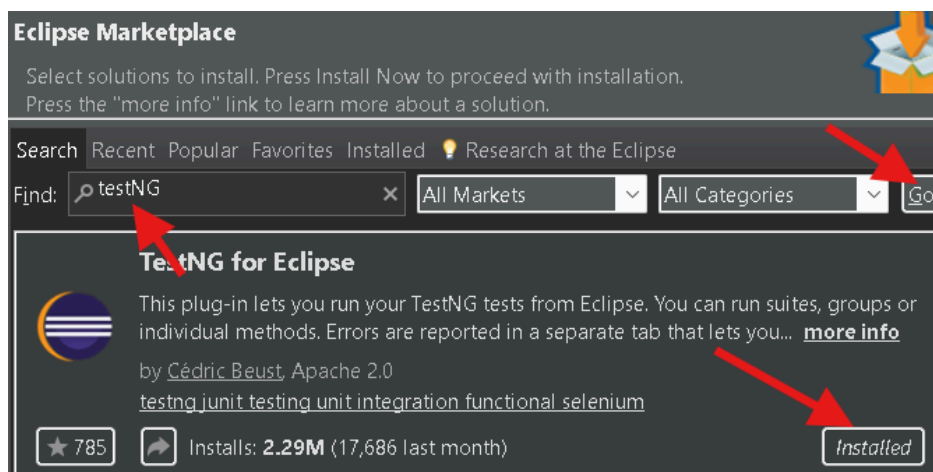
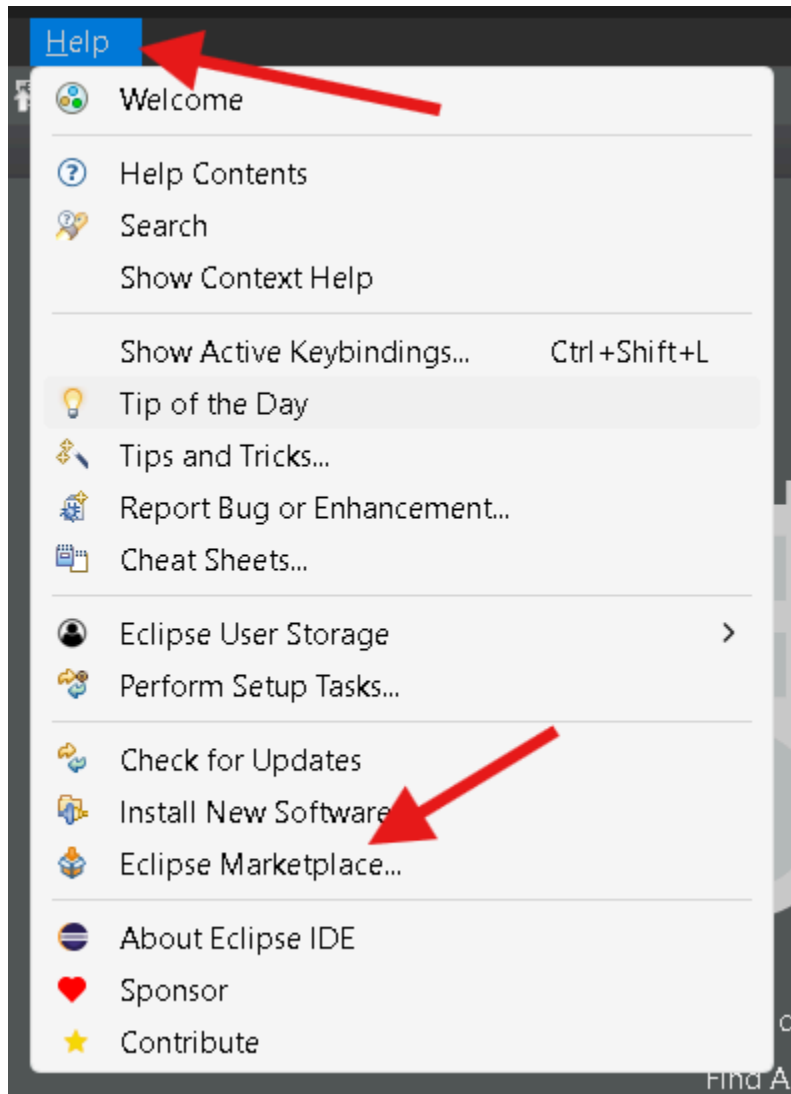


## To Check TestNG is installed before hand:

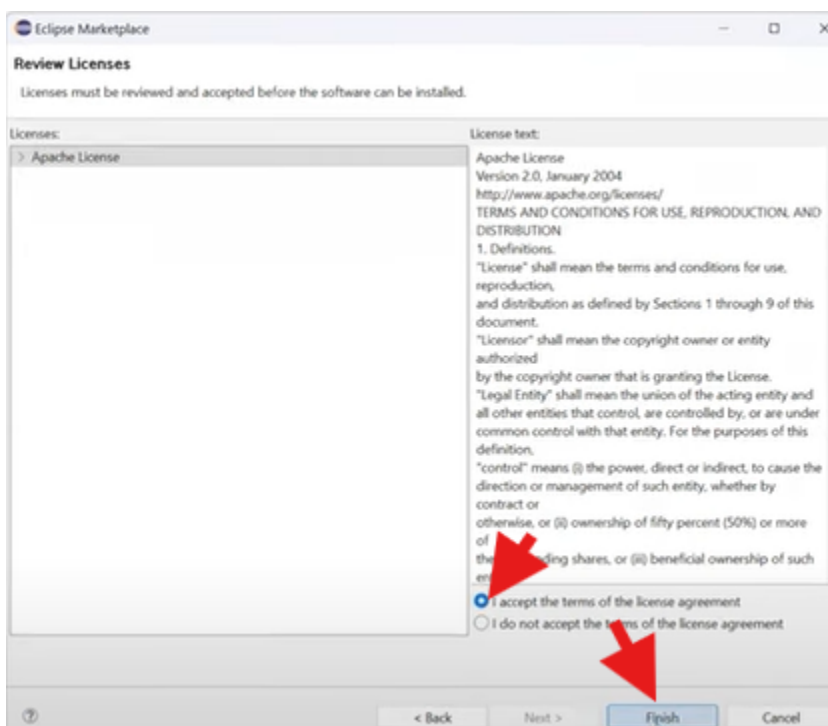
Window -> Preferences -> TestNG must be visible(If installed).



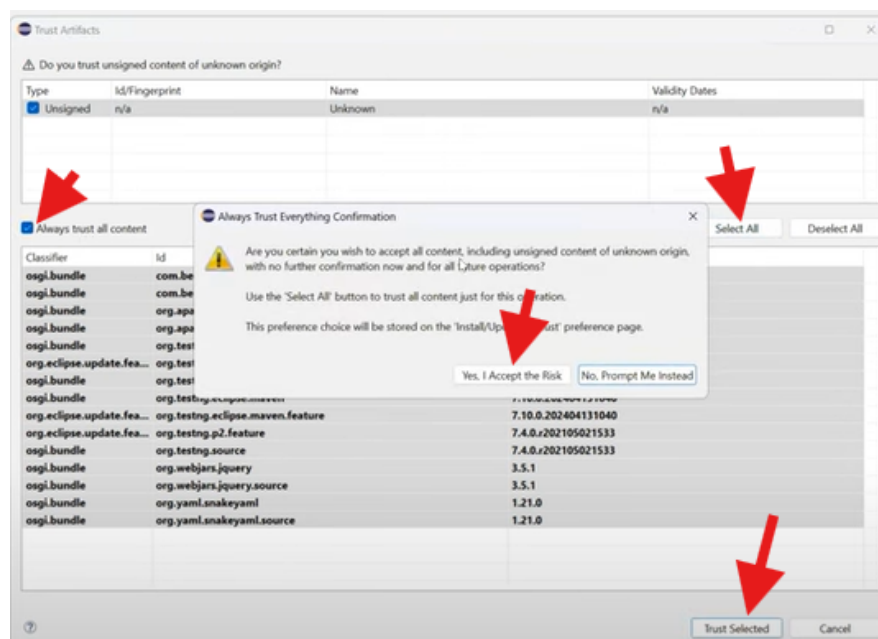
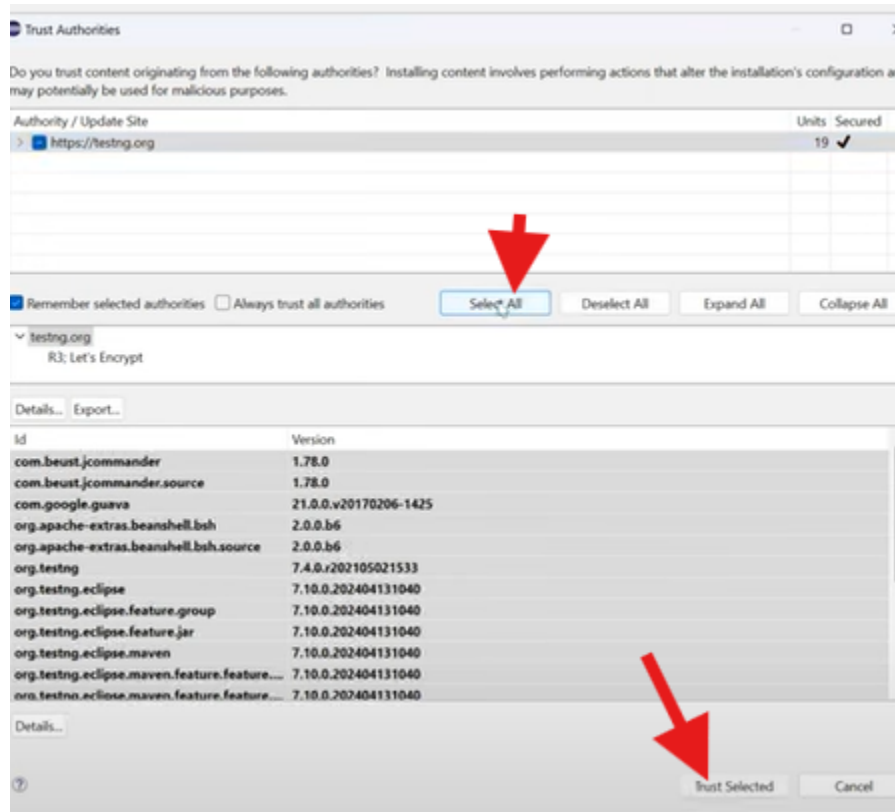
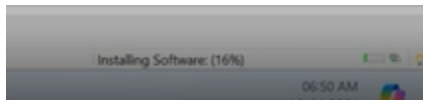
## To install TestNG:

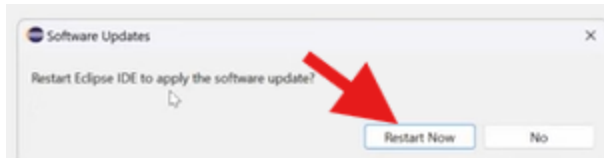


If not installed:

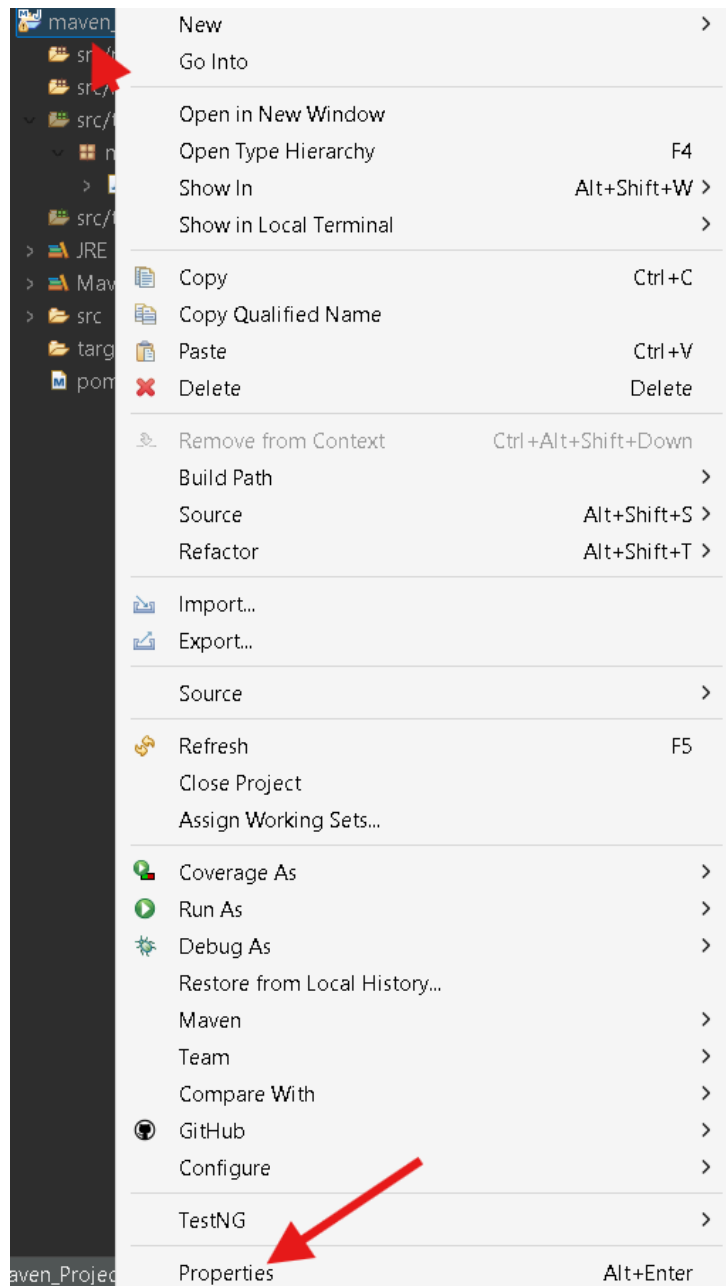


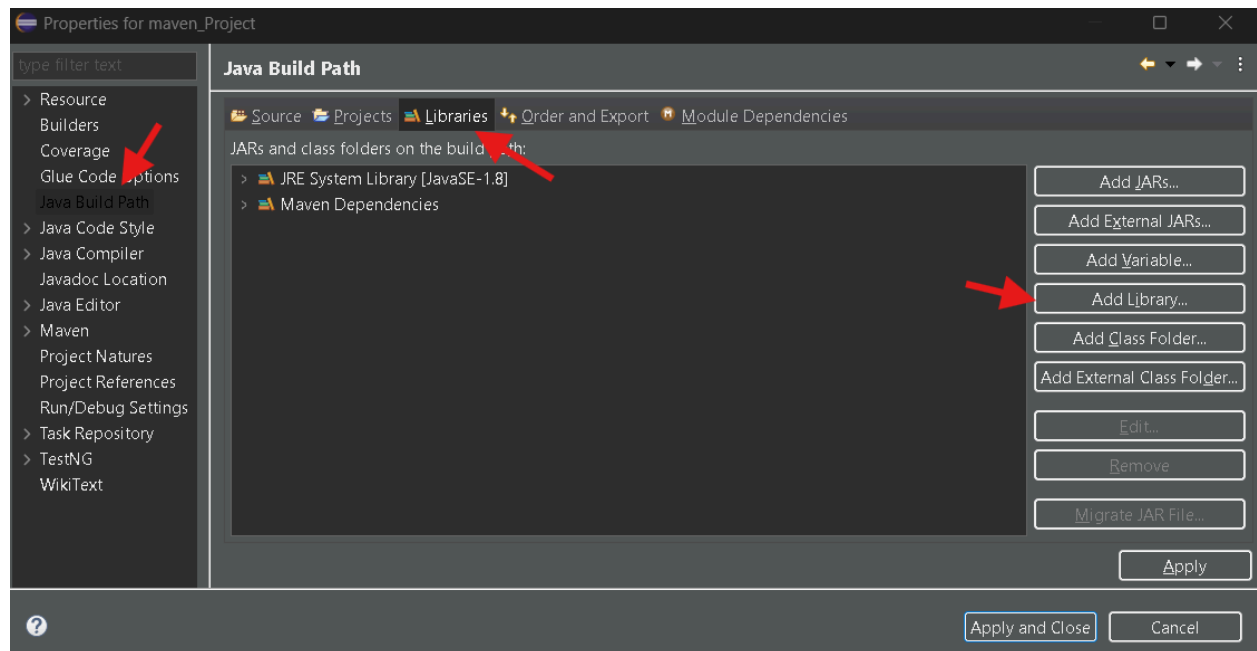
Wait until fully installed:

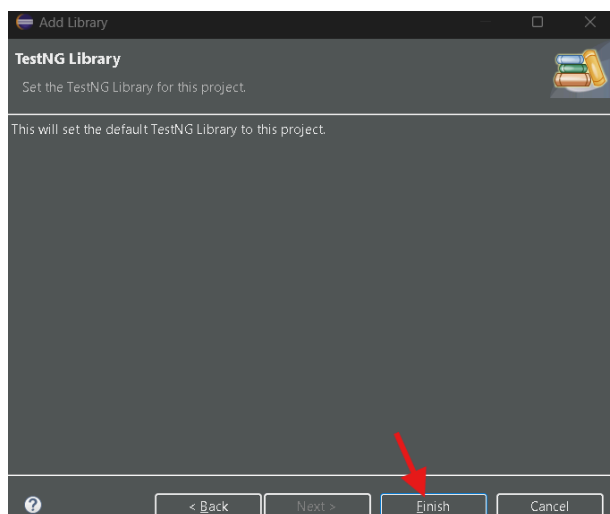
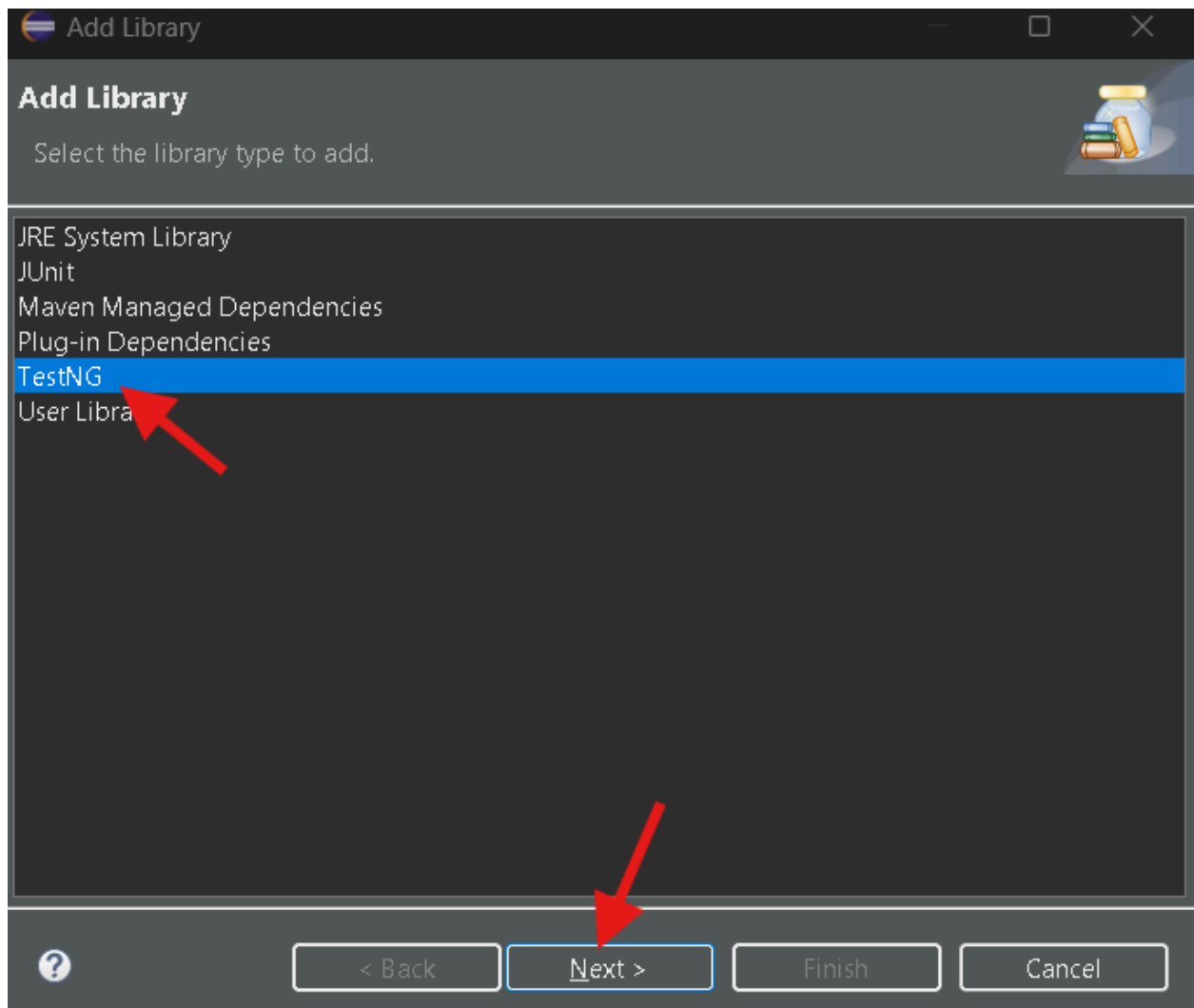


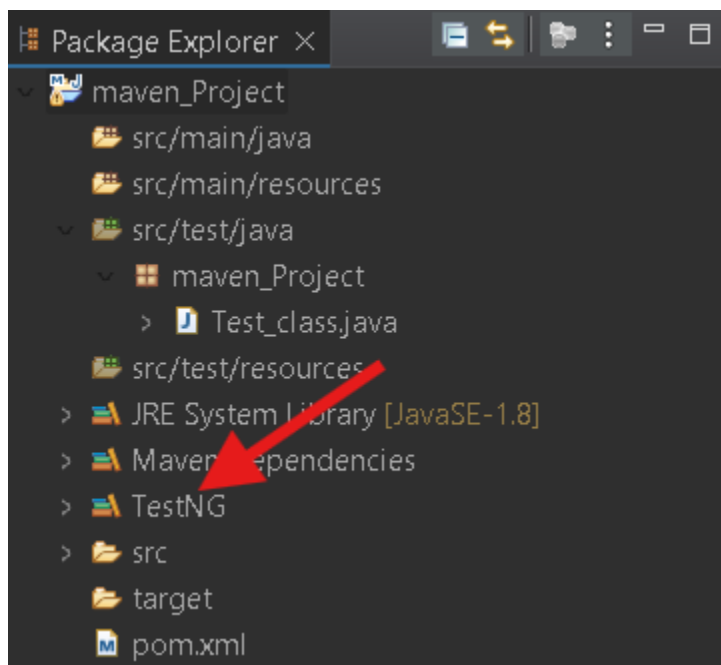
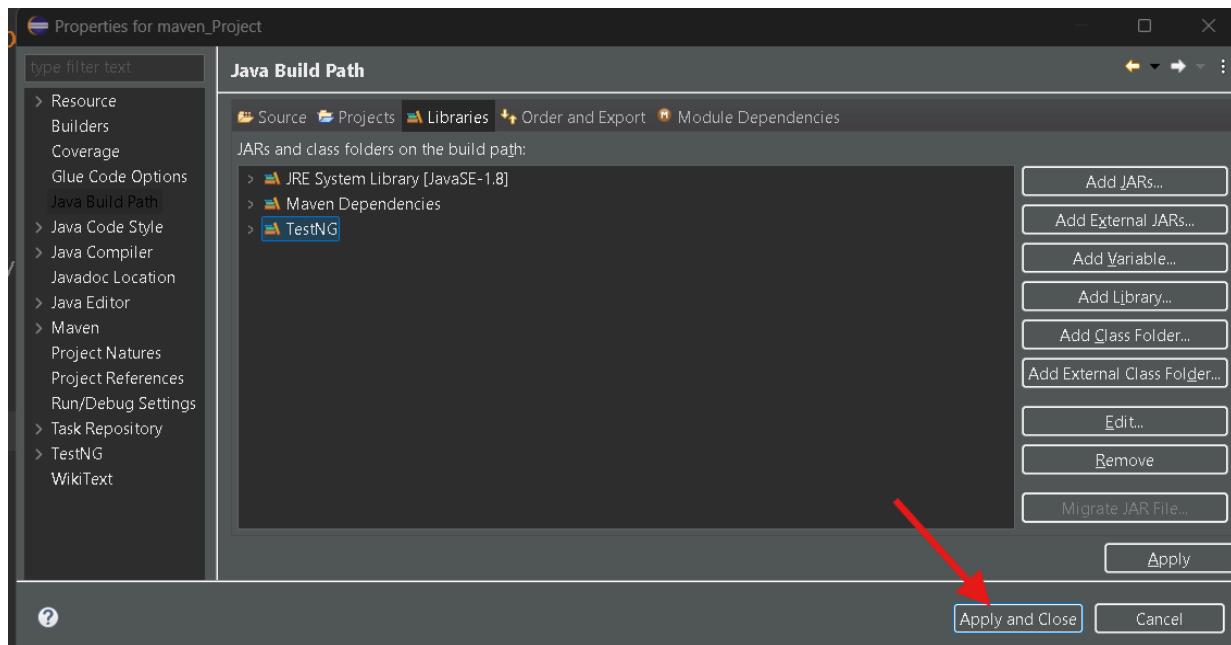


To add the TestNG to the Project:









Then adding the dependencies using Maven Repository:

Search: TestNG -> Search



Search for groups, artifacts, categories

Home » [org.testng](#) » [testng](#)



## TestNG

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionality easier to use. It supports test configured by annotations, data-driven testing, parametric tests, etc.

License	Apache 2.0
Categories	Testing Frameworks & Tools
Tags	testing testng quality
HomePage	<a href="https://testng.org">https://testng.org</a>
Ranking	#53 in MvnRepository (See Top Artifacts) #6 in Testing Frameworks & Tools
Used By	12,438 artifacts

Central (89)	Redhat GA (1)	Redhat EA (2)	FenixEdu (1)	Gradle Releases (1)	IBM OLRepo (1)
EmergyaPub (1)	ICM (4)	JCenter (1)	Spring Lib M (17)	Spring Lib Release (1)	Spring Plugins (1)
Version ▼			Vulnerabilities		Repository
7.11.x	7.11.0				Central

Add annotations `@Test` to run Test for methods.

```
Package Explorer ×
maven_Project
├── src/main/java
├── src/main/resources
├── src/test/java
│   ├── App_package
│   │   └── app_Class.java
│   ├── maven_Project
│   │   └── Test_class.java
│   └── src/test/resources
├── JRE System Library [JavaSE-1.8]
├── Maven Dependencies
├── TestNG
├── src
├── target
└── pom.xml

1 package App_package;
2
3 import org.testng.annotations.Test;
4
5 /*
6  Opens a app.
7  Does Login.
8  Does Logout.
9  */
10
11 Run ALL
12 public class app_Class {
13     @Test
14     void openapp() {
15         System.out.println("Opening Application...");
16     }
17 }
```

## Code:

```
package App_package;
import org.testng.annotations.Test;
/*
Opens a app.
Does Login.
Does Logout.
*/
public class app_Class {
    @Test
    void openapp() {
        System.out.println("Opening Application...");
    }

    @Test
    void login() {
        System.out.println("Logging into Application...");
    }

    @Test
    void logout() {
        System.out.println("Logging out Application...");
    }
}
```

Right click on the code -> Run As -> TestNG Test

[RemoteTestNG] detected TestNG version 7.11.0

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

SLF4J(W): See <https://www.slf4j.org/codes.html#noProviders> for further details.

Logging into Application...

Logging out Application...

Opening Application...

PASSED: App\_package.app\_Class.login

PASSED: App\_package.app\_Class.logout

PASSED: App\_package.app\_Class.openapp

=====

Default test

Tests run: 3, Failures: 0, Skips: 0

=====

=====

Default suite

Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

=====

Test annotations lets us to run the application in TestNG Test.

Above code output is not procedural/in alphabetical order, like opening the app comes first, then login, then logout.

## To Run the tests in procedural steps:

```
package App_package;
import org.testng.annotations.Test;
public class app_Class {
    @Test(priority = 1)
    void openapp() {
        System.out.println("Opening Application...");
    }
    @Test(priority = 2)
    void login() {
        System.out.println("Logging into Application...");
    }
    @Test(priority = 3)
    void logout() {
        System.out.println("Logging out Application...");
    }
}
```

## Output:

[RemoteTestNG] detected TestNG version 7.11.0

SLF4J(W): No SLF4J providers were found.

SLF4J(W): Defaulting to no-operation (NOP) logger implementation

SLF4J(W): See <https://www.slf4j.org/codes.html#noProviders> for further details.

Opening Application...

Logging into Application...

Logging out Application...

PASSED: App\_package.app\_Class.openapp

PASSED: App\_package.app\_Class.logout

PASSED: App\_package.app\_Class.login

=====

Default test

Tests run: 3, Failures: 0, Skips: 0

=====

=====

Default suite

Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

=====

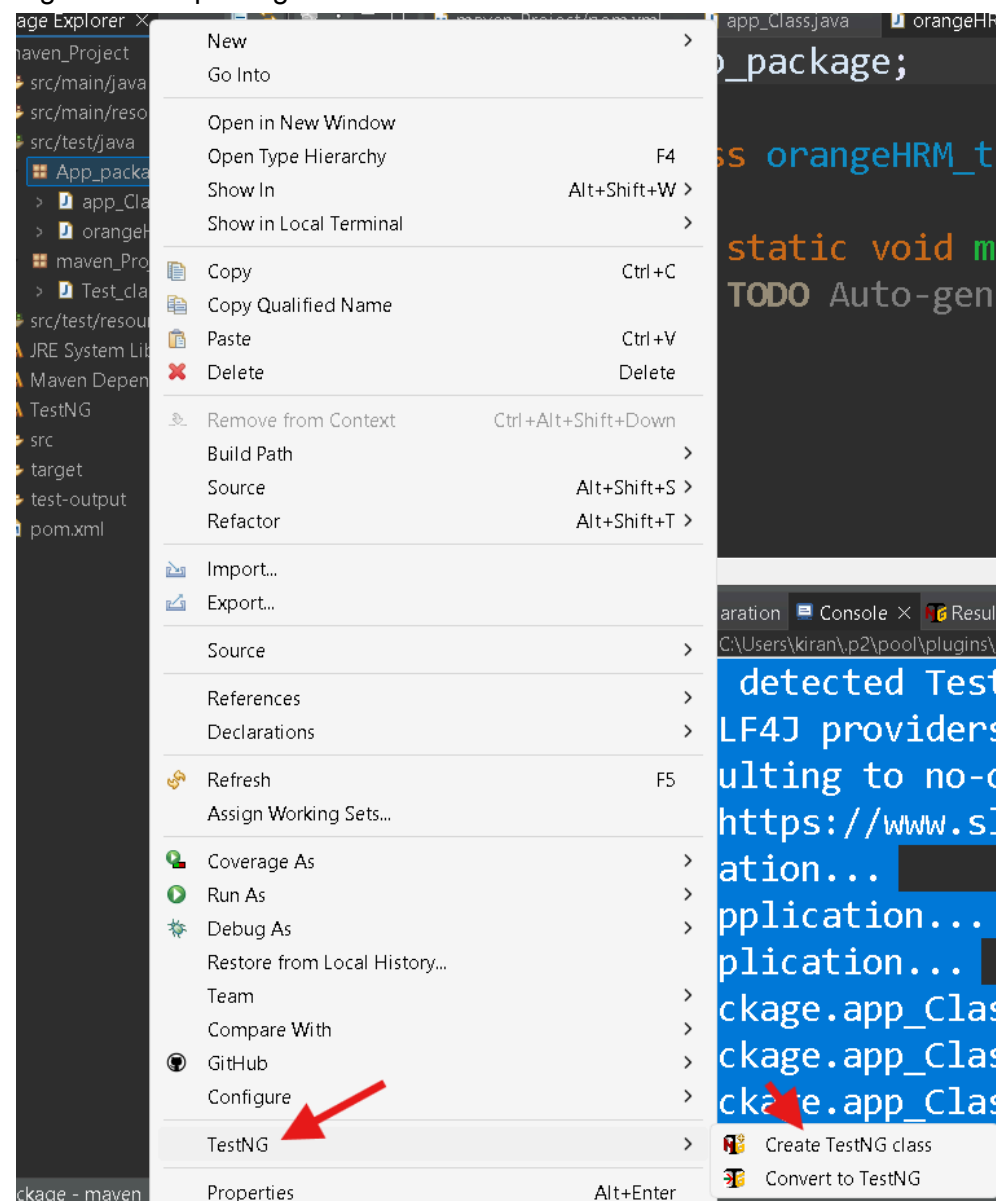
By default value of priority is ZERO.

If priority is same it runs in alphabetical order.

If @Test annotations are not given then, it does not run the test after running the test annotations.

## Create TestNG Class:

Right click on package -> TestNG -> Create TestNG Class



### New TestNG class

Specify additional information about the test class.

Source folder:

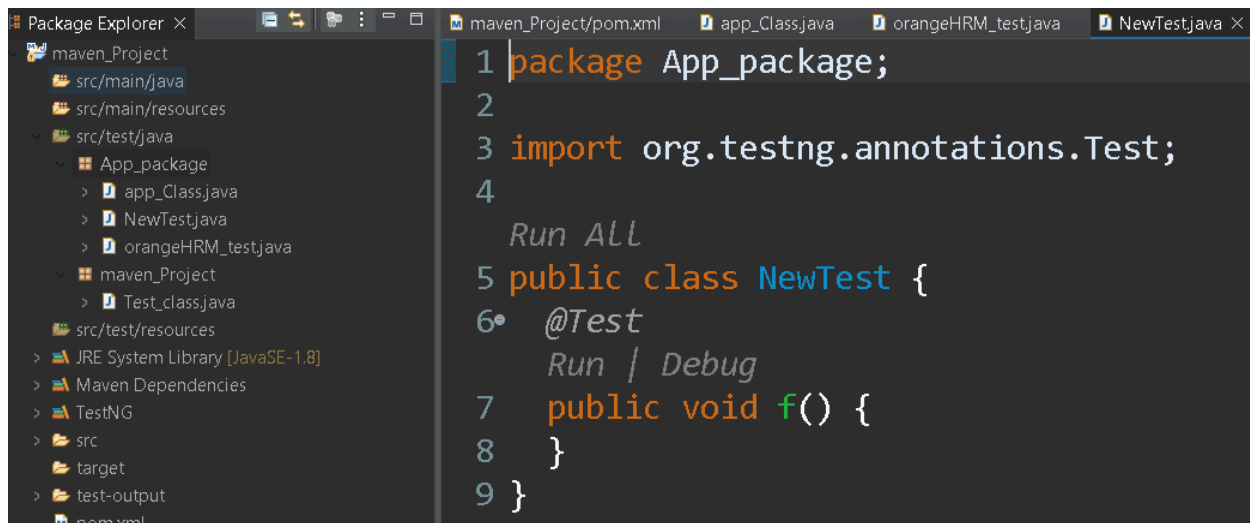
Package name:

Class name:

Annotations

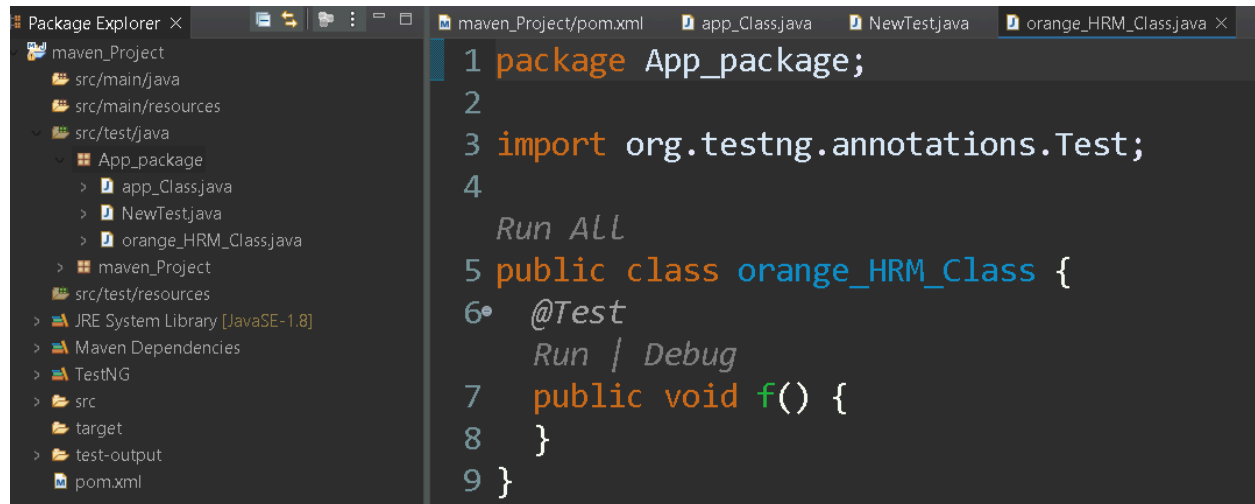
<input type="checkbox"/> @BeforeMethod	<input type="checkbox"/> @AfterMethod	<input type="checkbox"/> @DataProvider
<input type="checkbox"/> @BeforeClass	<input type="checkbox"/> @AfterClass	
<input type="checkbox"/> @BeforeTest	<input type="checkbox"/> @AfterTest	
<input type="checkbox"/> @BeforeSuite	<input type="checkbox"/> @AfterSuite	

XML suite file:



```
1 package App_package;
2
3 import org.testng.annotations.Test;
4
5 Run ALL
6 public class NewTest {
7     Run | Debug
8     @Test
9     public void f() {
```

Similarly, do for the class orange\_HRM\_Class.



## Code:

```
package App_package;
```

```
import java.time.Duration;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

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

```
public class orange_HRM_Class {  
    WebDriver driver;
```

```
    public static void p(Object obj) {  
        System.out.println(obj);  
    }
```

```
@Test(priority = 1)
```

```
public void open_app() {  
    driver = new ChromeDriver();  
    driver.manage().window().maximize();  
//    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));  
    driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");  
}
```

```
@Test(priority = 2)
```

```

public void test_logo() {
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(5));
    WebElement logo =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@alt='company-branding'
]")));
    boolean is_displayed = logo.isDisplayed();
    if (is_displayed) {
        p("Yes, Logo displayed!");
    } else p("No, Logo not displayed!");
}

@Test(priority = 3)
public void test_login() {
    String username = "Admin";
    String password = "admin123";
    //form/descendant::button
    String path = "(//form/descendant::";
    driver.findElement(By.xpath(path + "input)[2]")).sendKeys(username);
    driver.findElement(By.xpath(path + "input)[3]")).sendKeys(password);
    driver.findElement(By.xpath(path + "button"))).click();
}

@Test(priority = 4)
public void close() {
    driver.quit();
}
}

```

## Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Yes, Logo displayed!
PASSED: App_package.orange_HRM_Class.test_login
PASSED: App_package.orange_HRM_Class.test_logo
PASSED: App_package.orange_HRM_Class.open_app
PASSED: App_package.orange_HRM_Class.close

```

```

=====
Default test
Tests run: 4, Failures: 0, Skips: 0
=====

```

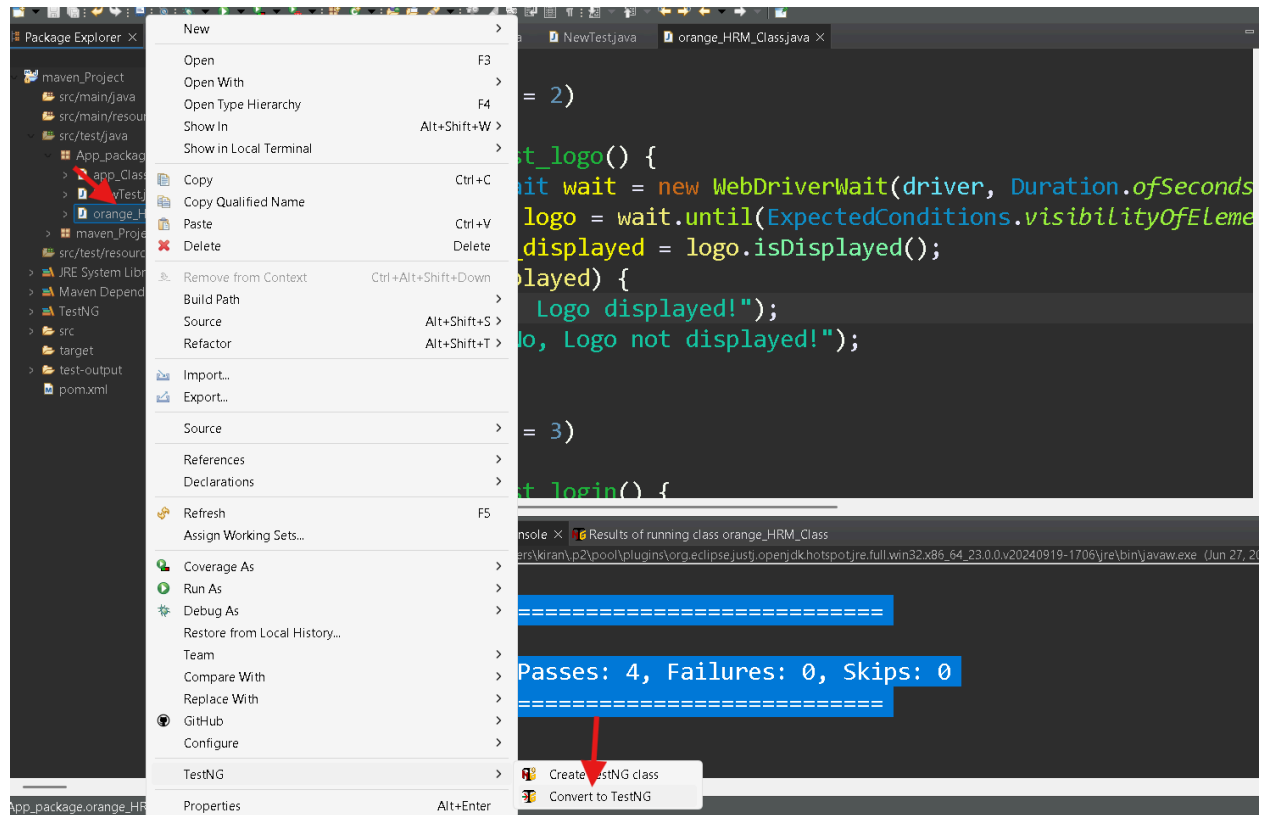
=====

Default suite

Total tests run: 4, Passes: 4, Failures: 0, Skips: 0

=====

To run test cases using XML file:





Refactoring

# Generate testng.xml

☒ Generate testng.xml

Location: /maven\_Project/src/test/java/App\_package 

Browse...

Suite name: edited\_Suite

Test name: edited\_Test

Class selection: 

Classes

 Parallel mode: 

none

 Thread count:

Preview

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="edited_Suite">
  <test thread-count="5" name="edited_Test">
    <classes>
      <class name="App_package.orange_HRM_Class"/>
    </classes>
  </test> <!-- edited_Test -->
</suite> <!-- edited_Suite -->
```

Code generation

suite() methods: 

Remove

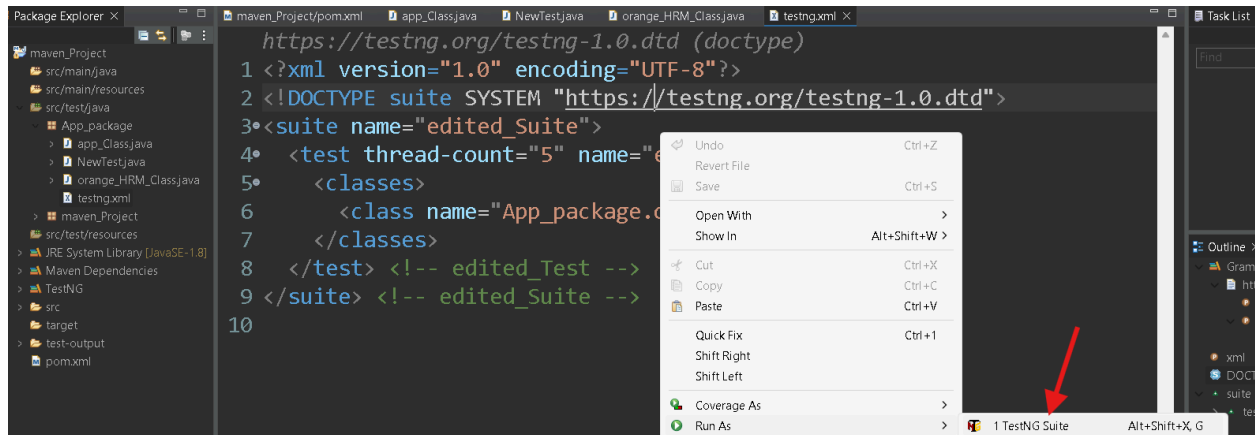
?

< Back

Next >

Finish

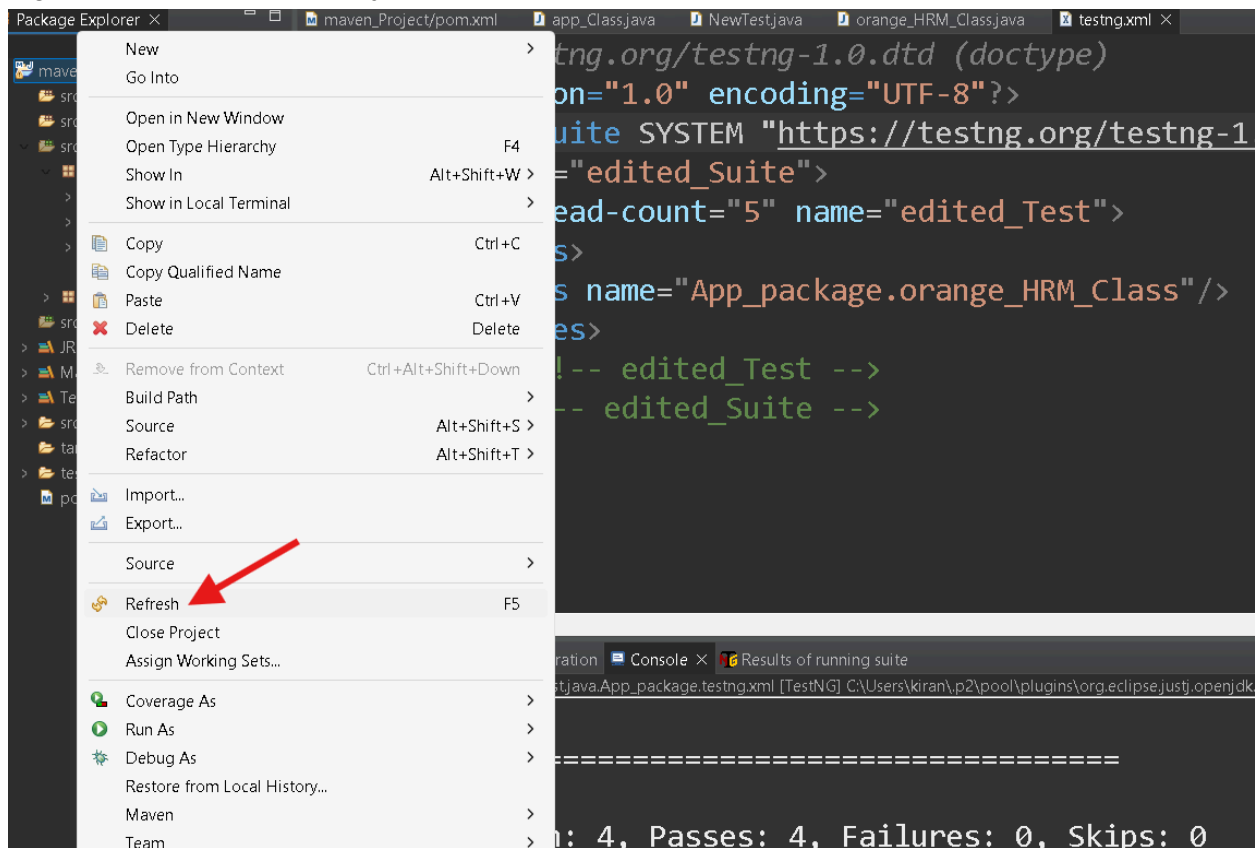
Cancel



By xml we can run single, multiple or group of tests.

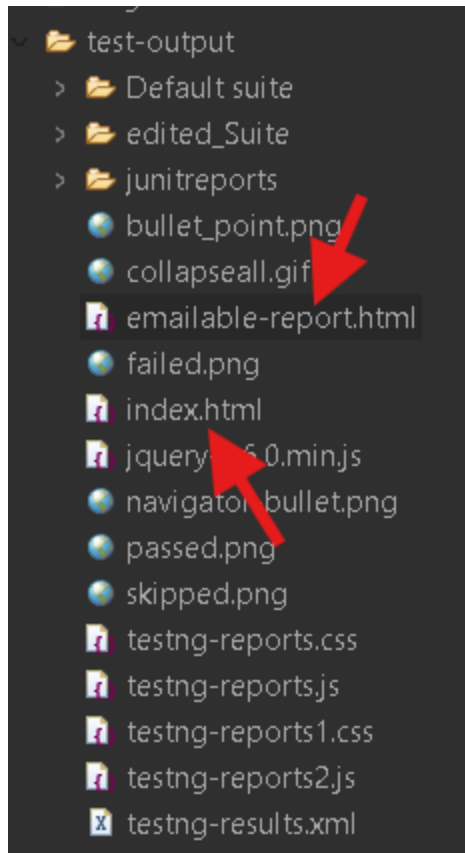
To see the report:

Right click on the maven project and refresh:

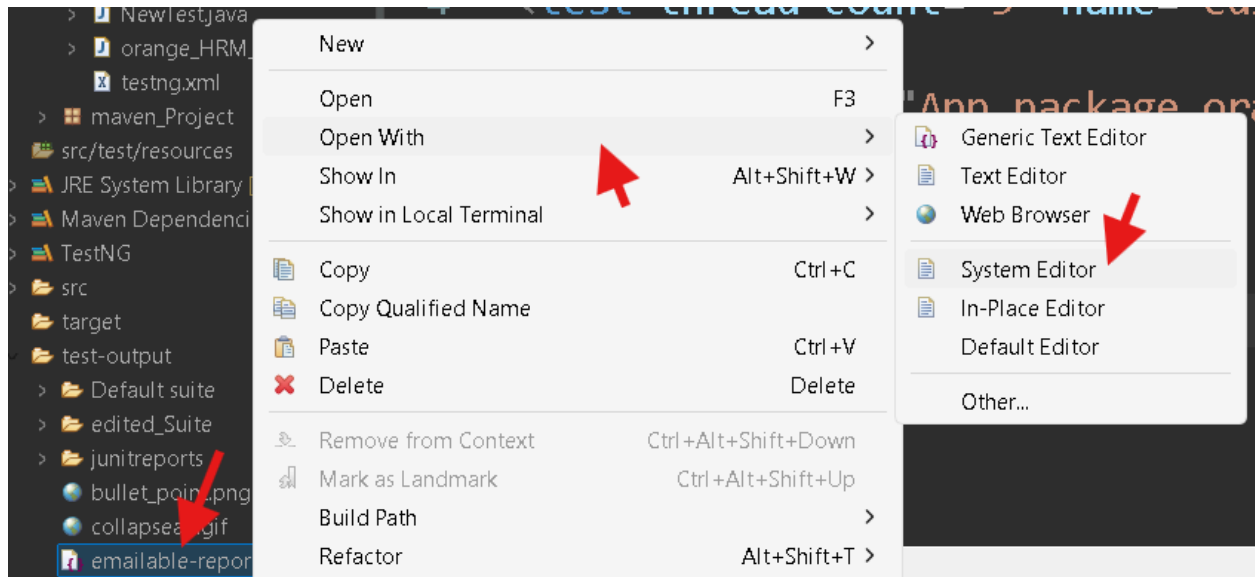


Then at test-output folder:

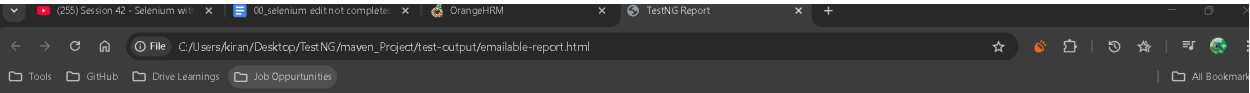
Two files are important those are emailable-report.html and index.html.



To see the report:



Then the report shown as:



Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
edited_Suite							
edited_Test	4	0	0	0	7,156		

Class	Method	Start	Time (ms)
edited_Suite			
edited_Test — passed			
App_package.orange_HRM_Class	close	1751002215792	121
	open_app	1751002208778	5821
	test_login	1751002215671	120
	test_logo	1751002214602	1068

edited\_Test

App\_package.orange\_HRM\_Class#close

[back to summary](#)

App\_package.orange\_HRM\_Class#open\_app

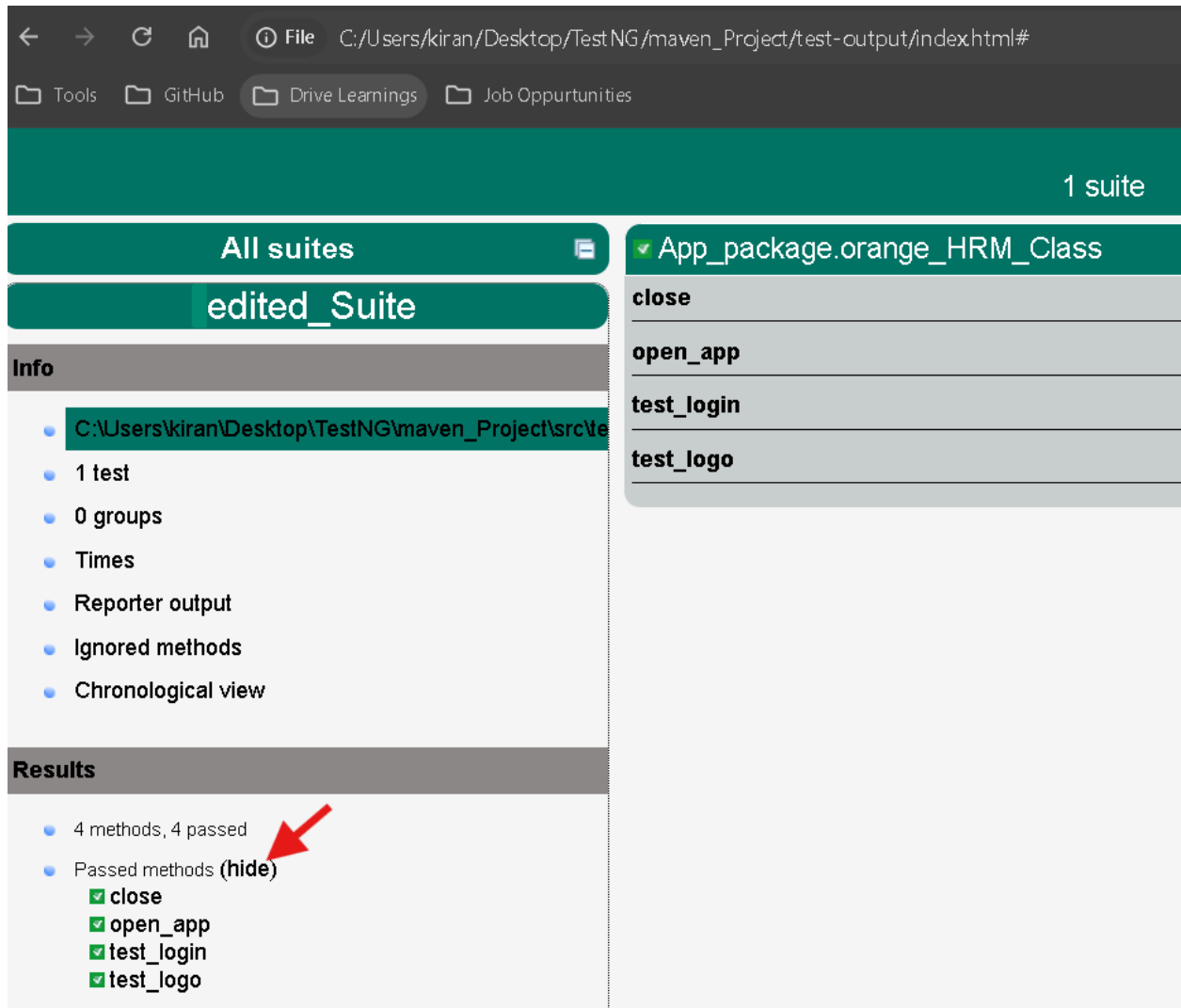
[back to summary](#)

App\_package.orange\_HRM\_Class#test\_login

[back to summary](#)

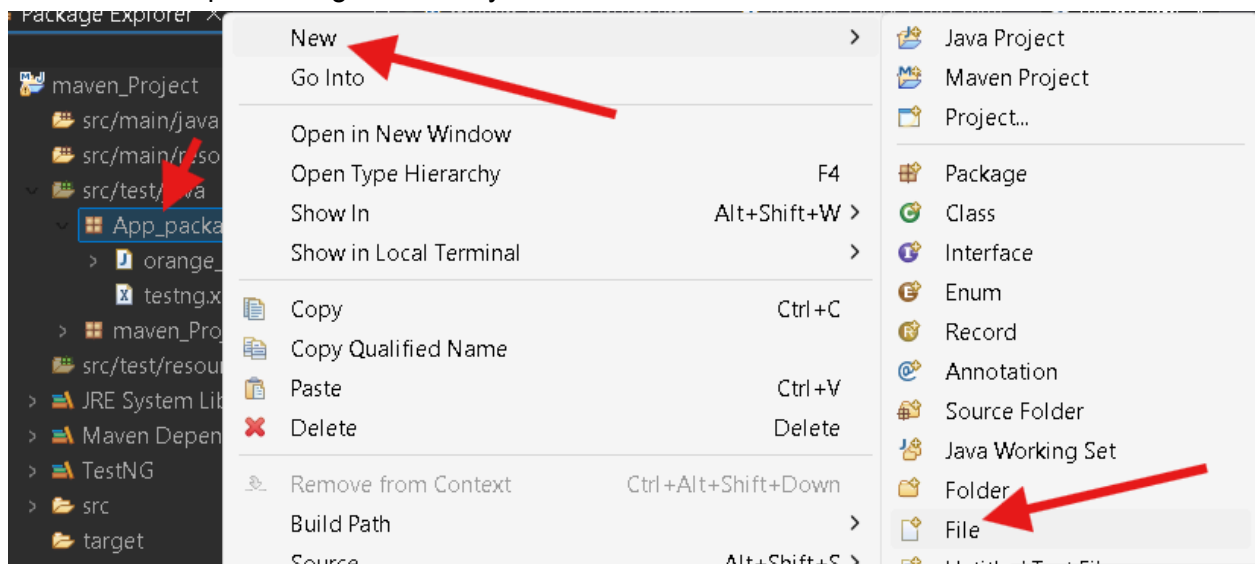
App\_package.orange\_HRM\_Class#test\_logo

For index.html:

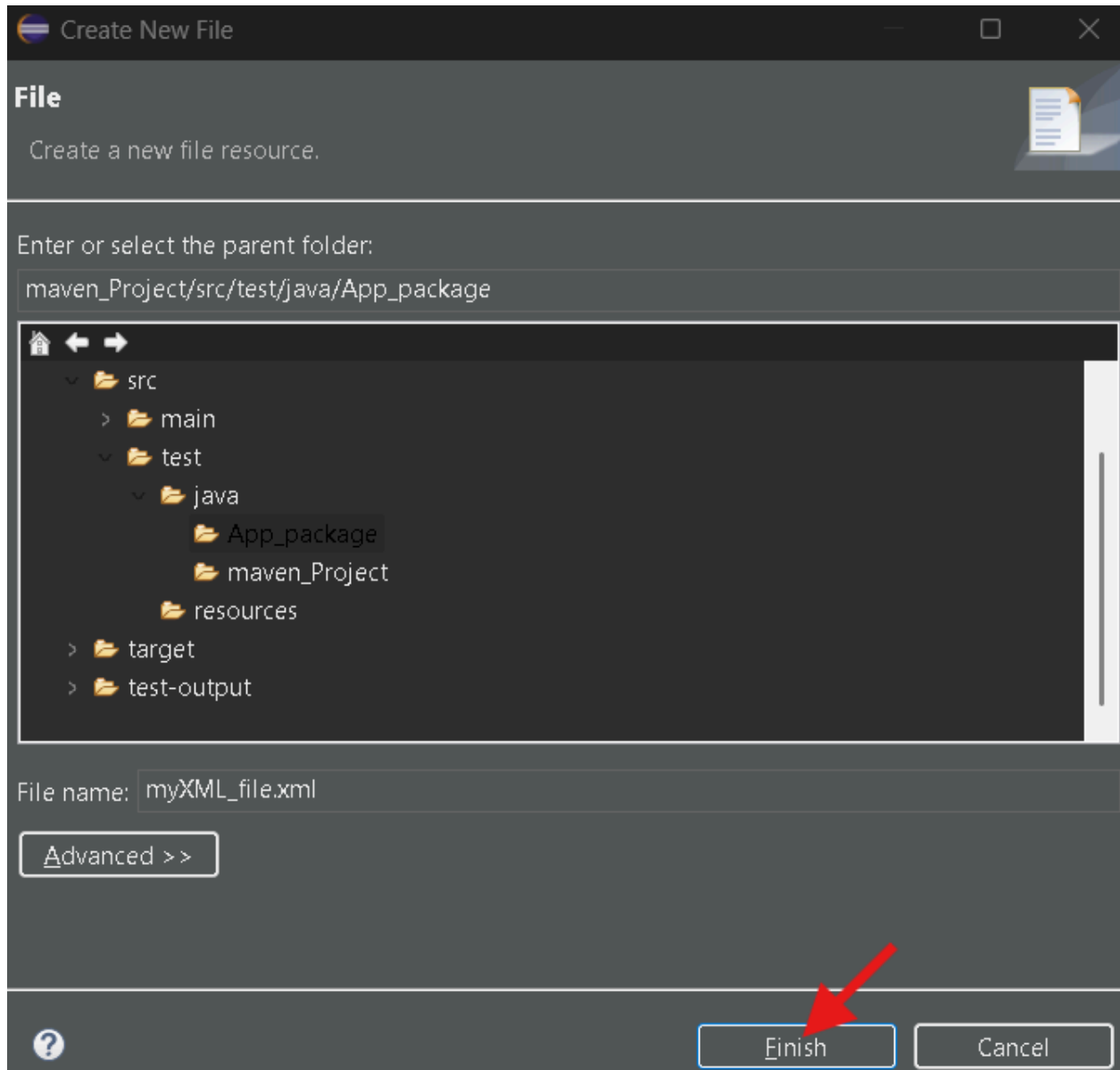


The screenshot shows a Selenium IDE test suite report in a web browser. The address bar indicates the file path: `C:/Users/kiran/Desktop/Test NG/maven_Project/test-output/index.html#`. The browser's tab bar shows 'Tools', 'GitHub', 'Drive Learnings', and 'Job Oppurtunities'. The report header shows '1 suite' and 'App\_package.orange\_HRM\_Class'. The 'All suites' tab is active, and the 'edited\_Suite' tab is also visible. The 'Info' section lists the test suite details: 'C:\Users\kiran\Desktop\TestNG\maven\_Project\src\le', '1 test', '0 groups', 'Times', 'Reporter output', 'Ignored methods', and 'Chronological view'. The 'Results' section shows '4 methods, 4 passed' and a list of passed methods: 'close', 'open\_app', 'test\_login', and 'test\_logo'. A red arrow points to the 'Passed methods (hide)' link. The right sidebar shows the test suite details for 'App\_package.orange\_HRM\_Class', including the methods 'close', 'open\_app', 'test\_login', and 'test\_logo'.

These above reports are generated by xml execution. Now let's create our own xml file.



The screenshot shows the Eclipse IDE Package Explorer on the left, displaying the project structure: 'maven\_Project' with subfolders 'src/main/java', 'src/main/resources', 'src/test/java', and 'src/test/resources'. The 'App\_package' folder is selected. A right-click context menu is open over the 'App\_package' folder. The menu options are: 'New' (highlighted with a red arrow), 'Go Into', 'Open in New Window', 'Open Type Hierarchy' (F4), 'Show In' (Alt+Shift+W), 'Show in Local Terminal', 'Copy' (Ctrl+C), 'Copy Qualified Name', 'Paste' (Ctrl+V), 'Delete' (Delete), 'Remove from Context' (Ctrl+Alt+Shift+Down), 'Build Path', and 'Source' (Alt+Shift+S). The 'New' submenu is open, showing options: 'Java Project', 'Maven Project', 'Project...', 'Package', 'Class', 'Interface', 'Enum', 'Record', 'Annotation', 'Source Folder', 'Java Working Set', 'Folder' (highlighted with a red arrow), and 'File'.



### Code of first\_TestNG\_Class.java:

```
package App_package;

import org.testng.annotations.Test;

public class first_TestNG_Class {
    @Test(priority = 1)
    public void open_website() {
        System.out.println("Web Application opened...");
    }
}
```

```

@Test(priority = 2)
public void login() {
    System.out.println("Logged in to application...");
}

@Test(priority = 3)
public void close_website() {
    System.out.println("Web Application closed...");
}
}

```

## testng.xml file having:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="edited_Suite">
    <test thread-count="5" name="edited_Test">
        <classes>
            <class name="App_package.orange_HRM_Class"/>
            <class name="App_package.first_TestNG_Class"/>
        </classes>
    </test> <!-- edited_Test -->
</suite> <!-- edited_Suite -->

```

## Output:

```

[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Yes, Logo displayed!
Web Application opened...
Logged in to application...
Web Application closed...

```

```

=====
edited_Suite
Total tests run: 7, Passes: 7, Failures: 0, Skips: 0
=====

```

## XML file:

Its the test suites.  
 Test suites -> group of test cases

Test cases -> contains test steps  
Xml file for test suites

Xml file contains classes  
Classes contains methods