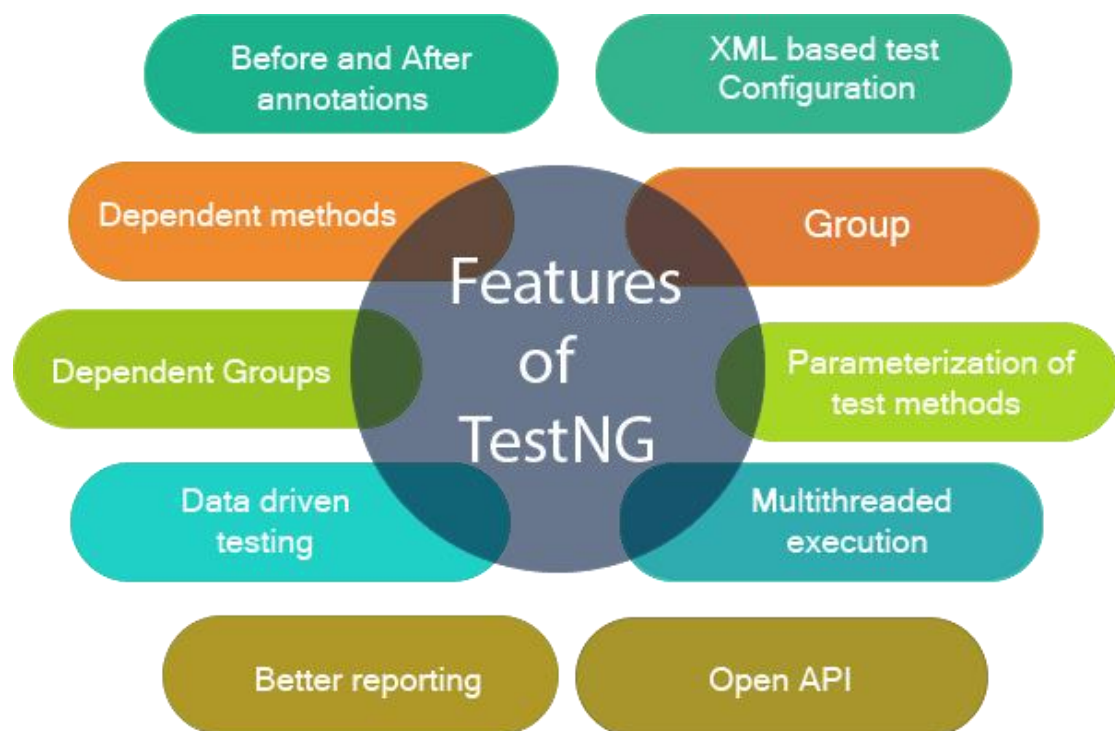


Test NG (Next Generation)

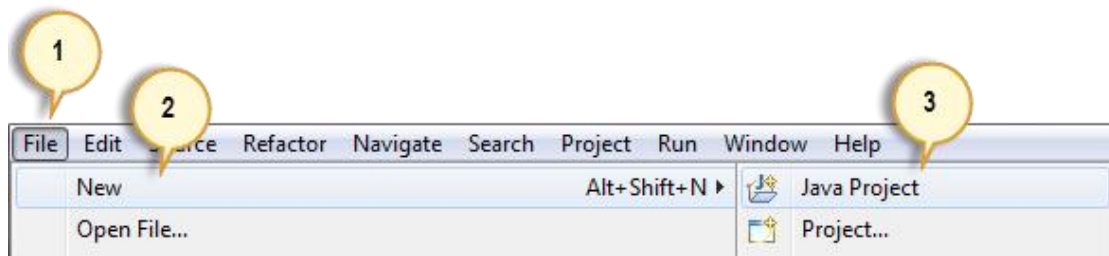
1. TestNG tutorial provides basic and advanced concepts of TestNG framework.
2. TestNG is one of the most widely used open source testing framework used in automation testing suite.
3. Our TestNG tutorial includes all topics of a testing framework such as Features, installation, running test cases, annotations, dependent test, parallel tests, before and after annotations, etc.
4. TestNG framework came after Junit, and TestNG framework adds more powerful functionality and easier to use.
5. It is an open source automated TestNG framework. In TestNG, NG stands for **"Next Generation"**.

Features of TestNG

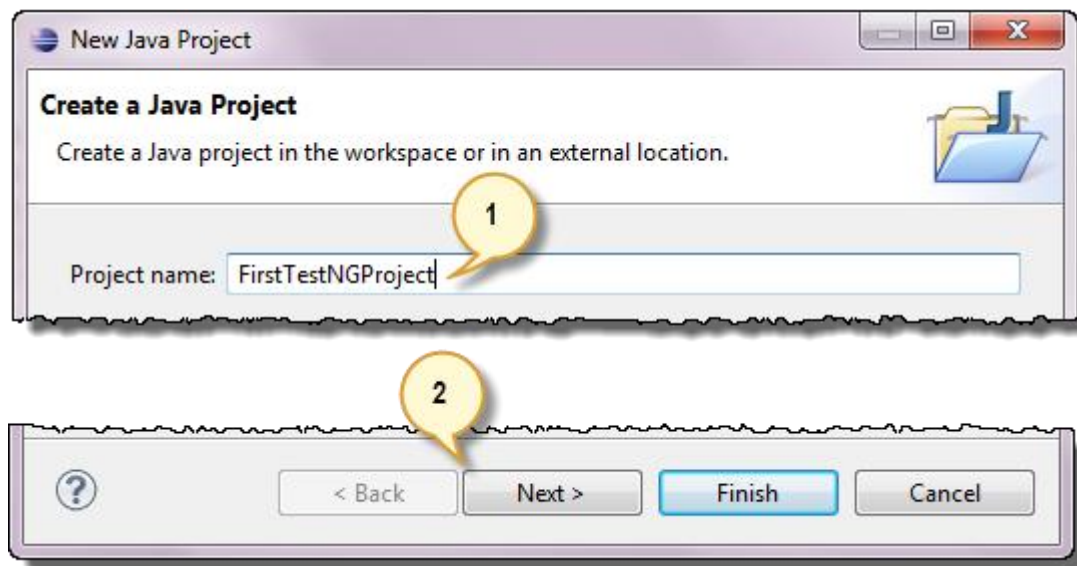


Let's create a new project in TestNG.

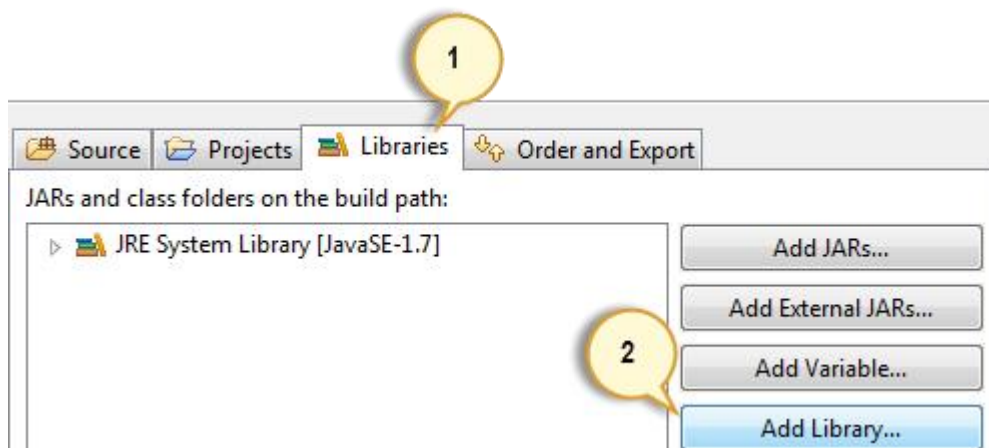
Step 1: Create a TestNG project. It is created in the same way as we create the java project. Click on the **File > New > Java project**.



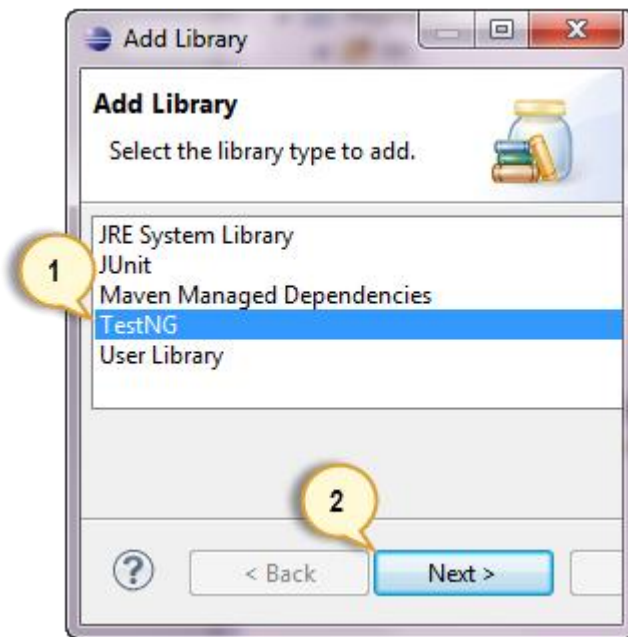
Step 2: Type “FirstTestNGProject” as the Project Name then click Next.



Step 3: We will now start to import the TestNG Libraries onto our project. Click on the “Libraries” tab, and then “Add Library...”

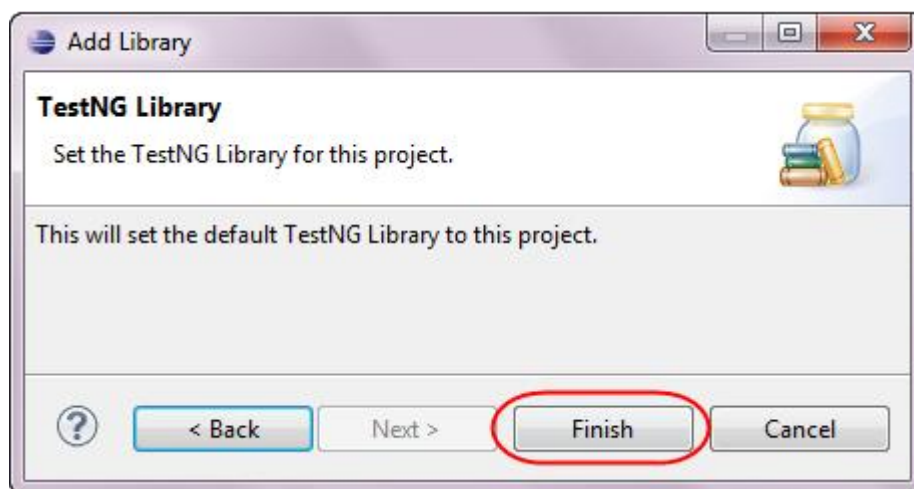


Step 4: On the Add Library dialog, choose “TestNG” and click

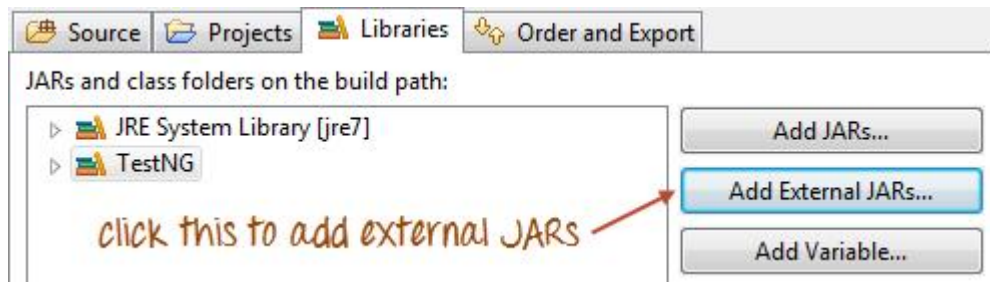


Next.

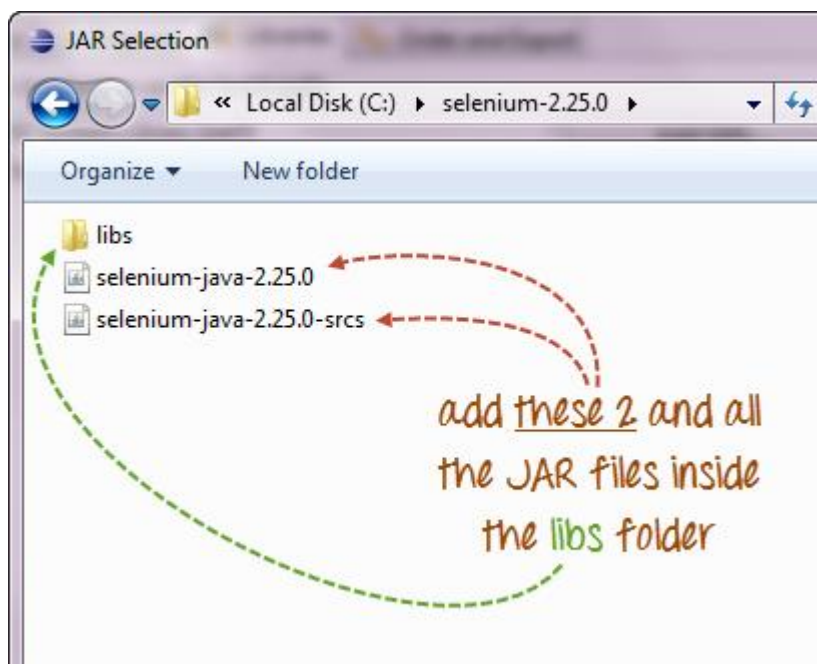
Step 5: Click Finish.



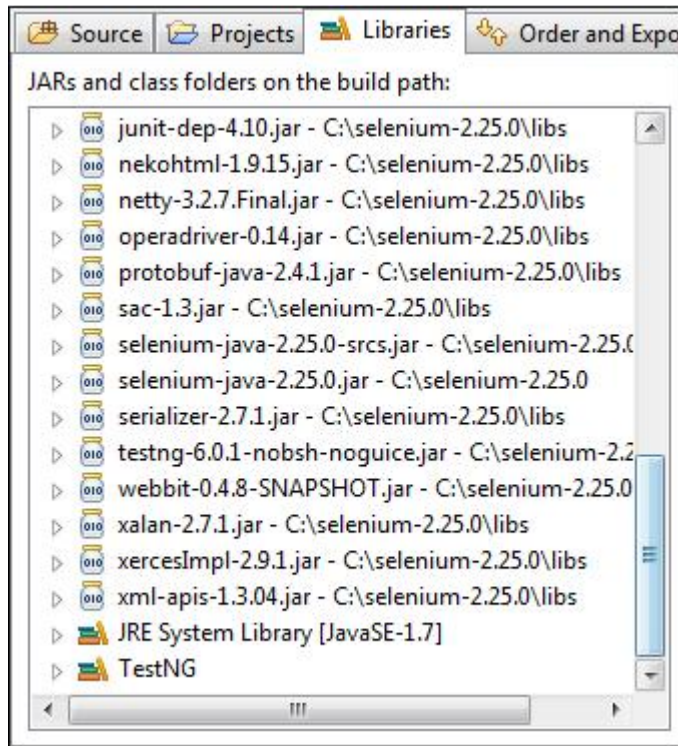
Step 6: We will now add the JAR files that contain the Selenium API. These files are found in the Java client driver that we downloaded from <https://www.selenium.dev/downloads/> when we were installing Selenium and Eclipse in the previous chapters.



Then, navigate to where you have placed the Selenium JAR files.



After adding the external JARs, your screen should look like this.



Step 7: Click Finish and verify that our FirstTestNGProject is visible on Eclipse's Package Explorer window.



Now that we are done setting up our project in this TestNG tutorial, let us create a new TestNG file.

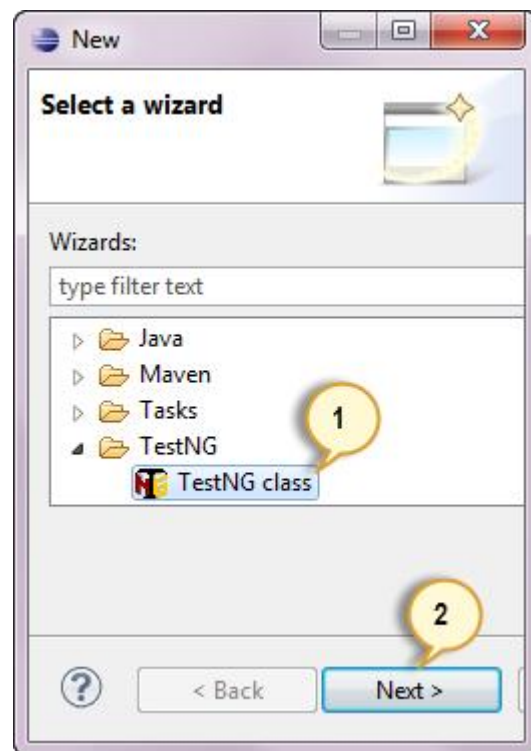
Step 1: Click on 'src' and Choose other.

Right-click on the "src" package folder then choose New > Other...



Step 2: Select TestNG class.

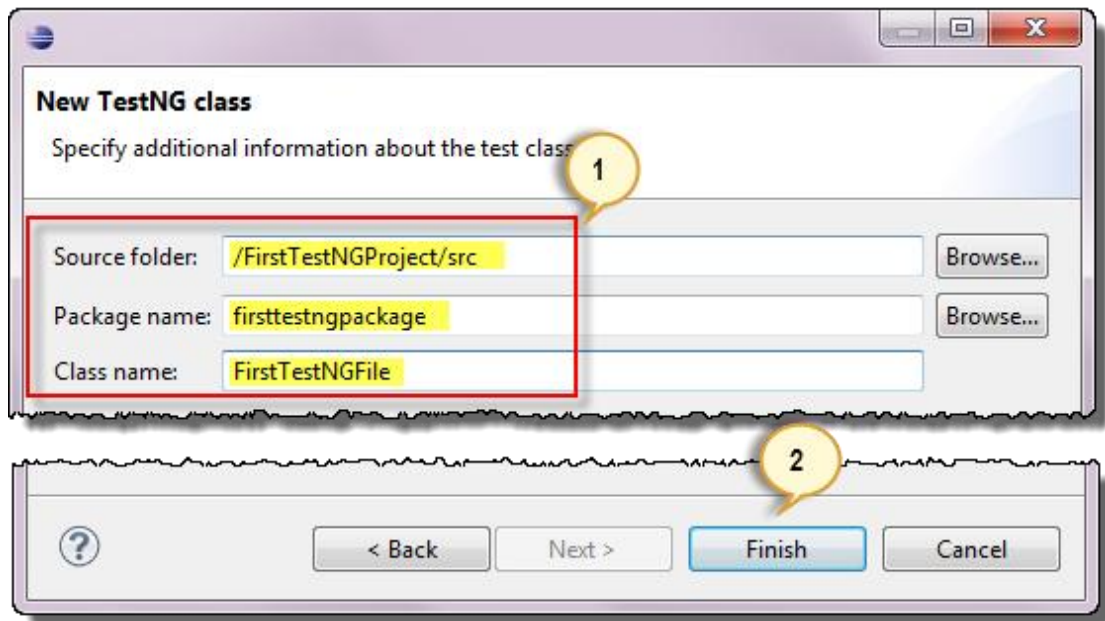
Click on the TestNG folder and select the “TestNG class” option. Click Next.



Step 3: Type the values.

Type the values indicated below on the appropriate input boxes and click Finish.

Notice that we have named our Java file as “FirstTestNGFile”.



Step 4: Template Created.

Eclipse should automatically create the template for our TestNG file shown below.

```
FirstTestNGFile.java
FirstTestNGProject > src > firsttestngpackage
1 package firsttestngpackage;
2
3 import org.testng.annotations.Test;
4
5 public class FirstTestNGFile {
6     @Test
7     public void f() {
8     }
9 }
10
```

Example code

```
package firsttestngpackage;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.*;
public class firsttestngfile {
    public String baseUrl = "http://demo.guru99.com/test/newtours/";
```

```

String driverPath = "C:\\geckodriver.exe";
public WebDriver driver ;

@BeforeTest
public void launchBrowser() {
    System.out.println("launching firefox browser");
    System.setProperty("webdriver.gecko.driver", driverPath);
    driver = new FirefoxDriver();
    driver.get(baseUrl);
}

@Test
public void verifyHomepageTitle() {
    String expectedTitle = "Welcome: Mercury Tours";
    String actualTitle = driver.getTitle();
    Assert.assertEquals(actualTitle, expectedTitle);
}

@AfterTest
public void terminateBrowser(){
    driver.close();
}
}

```

TestNG Annotations

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@AfterSuite: The annotated method will be run after all tests in this suite have run.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

@AfterMethod: The annotated method will be run after each test method.

@Test: The annotated method is a part of a test case