TestNG Introduction

TestNG is a testing framework inspired from ***JUnit*** and ***NUnit*** but introducing some new functionality that make it more powerful and easier to use.It is an open source automated testing framework; where ***NG*** *of* ***TestNG*** *means* ***N****ext* ***G****eneration*. TestNG is similar to JUnit but it is much more powerful than JUnit but still it's inspired by JUnit. It is designed to be better than JUnit, especially when testing integrated classes. Pay special thanks to *Cedric Beust who is the creator of TestNG*.
TestNG eliminates most of the limitations of the older framework and gives the developer the ability to write more flexible and powerful tests with help of easy annotations, grouping, sequencing & parametrizing.

Benefits of TestNG

There are number of benefits but from Selenium perspective, major advantages of TestNG are :

1. It gives the ability to produce ***HTML Reports*** of execution
2. ***Annotations*** made testers life easy
3. Test cases can be ***Grouped & Prioritized*** more easily
4. ***Parallel*** testing is possible
5. Generates ***Logs***
6. Data ***Parameterization*** is possible

Test Case Writing

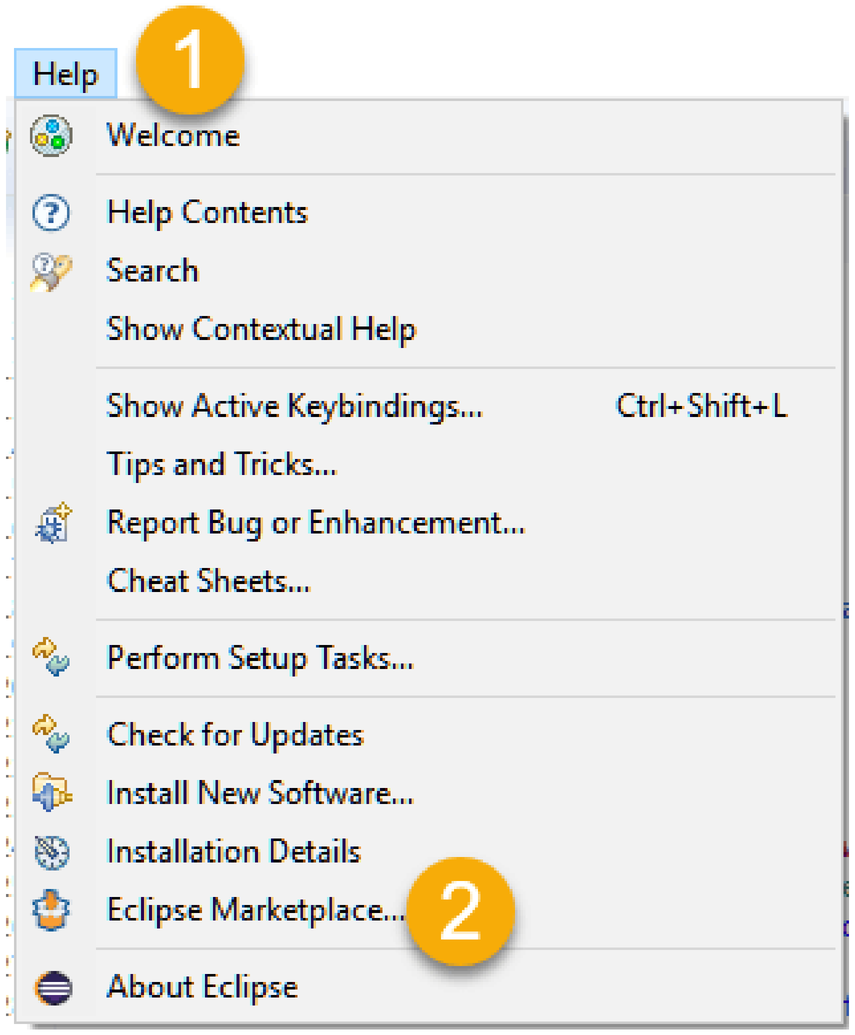
Writing a test in TestNG is quite simple and basically involves following steps:

Step 1 – Write the business logic of the test
Step 2 – Insert TestNG annotations in the code
Step 3 – Add the information about your test (e.g. the class names, methods names, groups names etc…) in a testng.xml file
Step 4 – Run TestNG

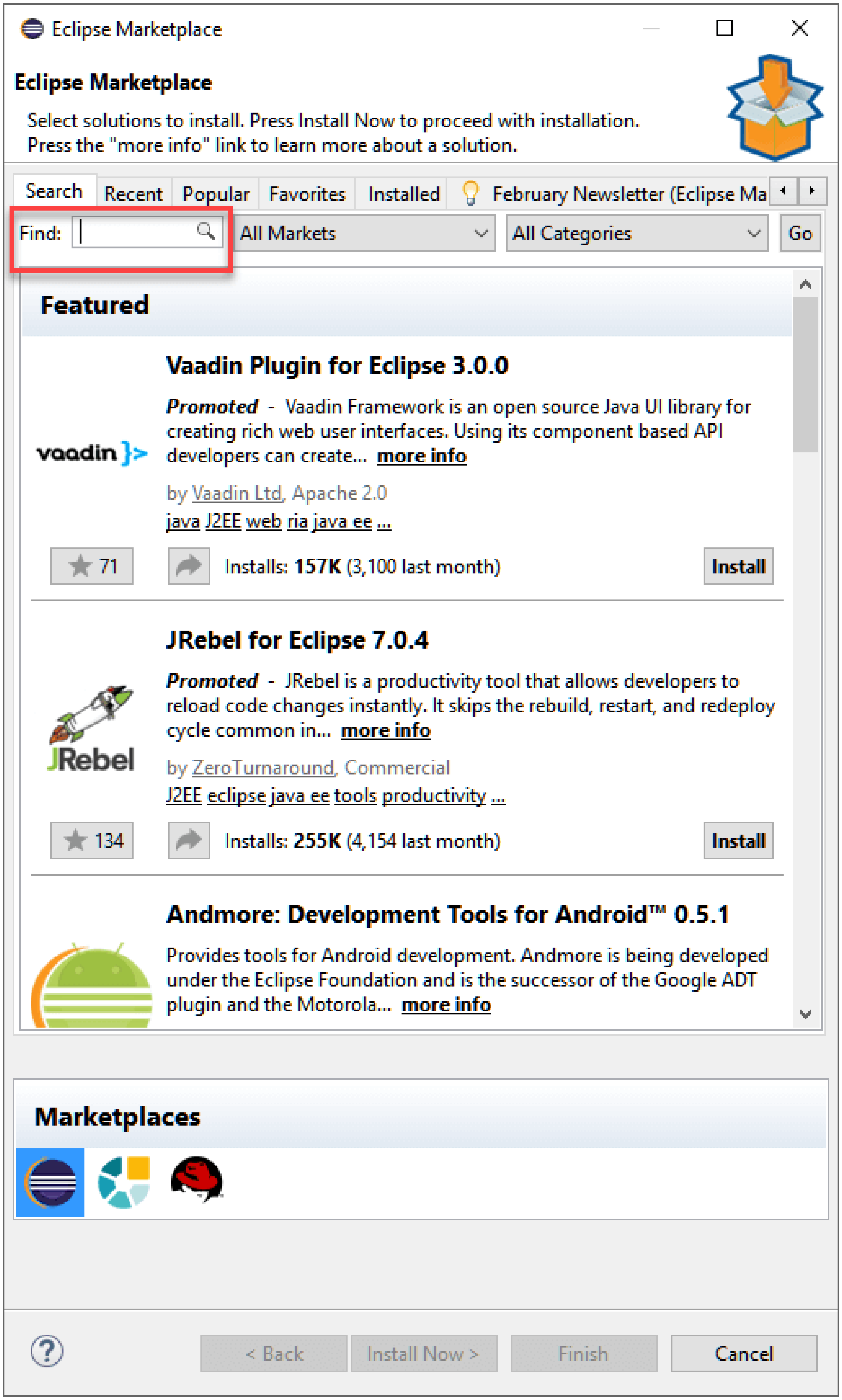
Installing TestNG in Eclipse

Step 1:

- Launch Eclipse.
- On the menu bar, click Help.
- Choose the "Eclipse Marketplace..." option.



Step 2: In the Eclipse Marketplace dialog box, type TestNG in the search box and press the search button(magnifying glass) or press enter key



Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Featured

Vaadin Plugin for Eclipse 3.0.0

Promoted - Vaadin Framework is an open source Java UI library for creating rich web user interfaces. Using its component based API developers can create... more info

by Vaadin Ltd, Apache 2.0
java J2EE web ria java ee ...

★ 71 Installs: **157K** (3,100 last month) Install

JRebel for Eclipse 7.0.4

Promoted - JRebel is a productivity tool that allows developers to reload code changes instantly. It skips the rebuild, restart, and redeploy cycle common in... more info

by ZeroTurnaround, Commercial
J2EE eclipse java ee tools productivity ...

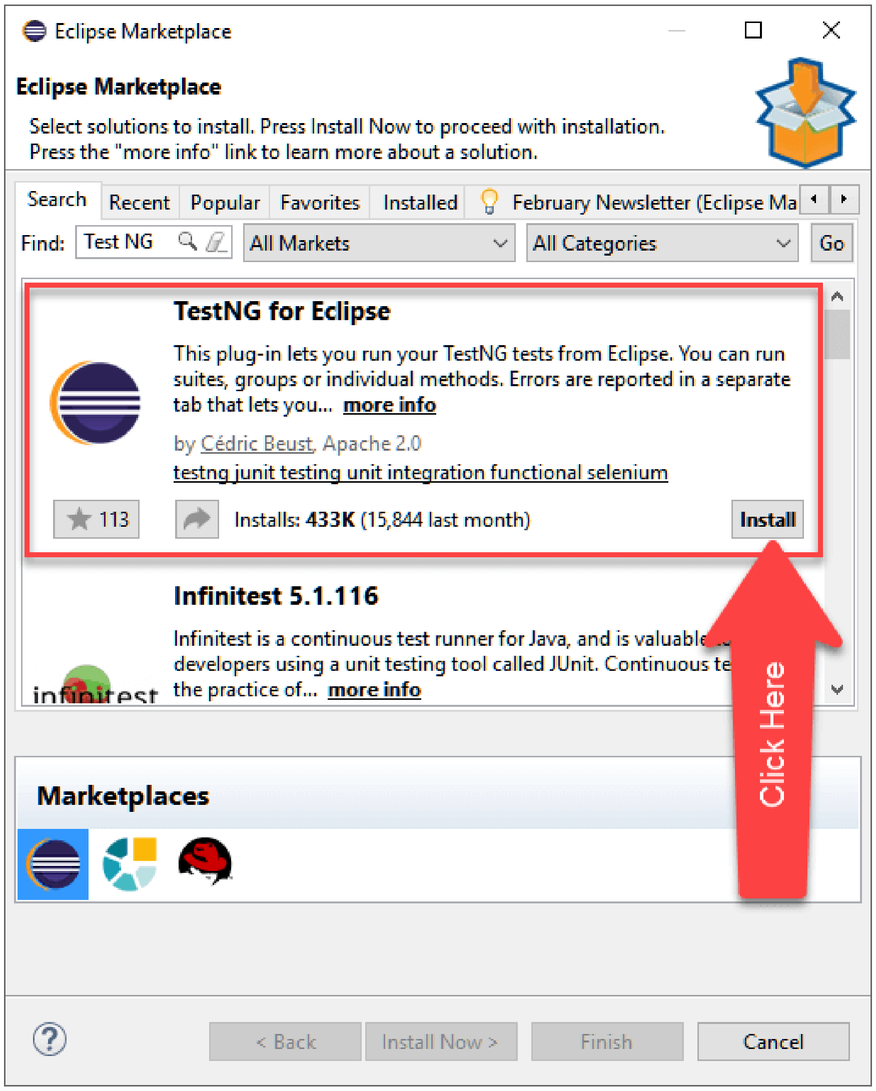
★ 134 Installs: **255K** (4,154 last month) Install

Andmore: Development Tools for Android™ 0.5.1

Provides tools for Android development. Andmore is being developed under the Eclipse Foundation and is the successor of the Google ADT plugin and the Motorola... more info

Marketplaces

Step 3: Click Install


Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: Test NG — All Markets — All Categories

TestNG for Eclipse

This plug-in lets you run your TestNG tests from Eclipse. You can run suites, groups or individual methods. Errors are reported in a separate tab that lets you... more info

by Cédric Beust, Apache 2.0

testng junit testing unit integration functional selenium

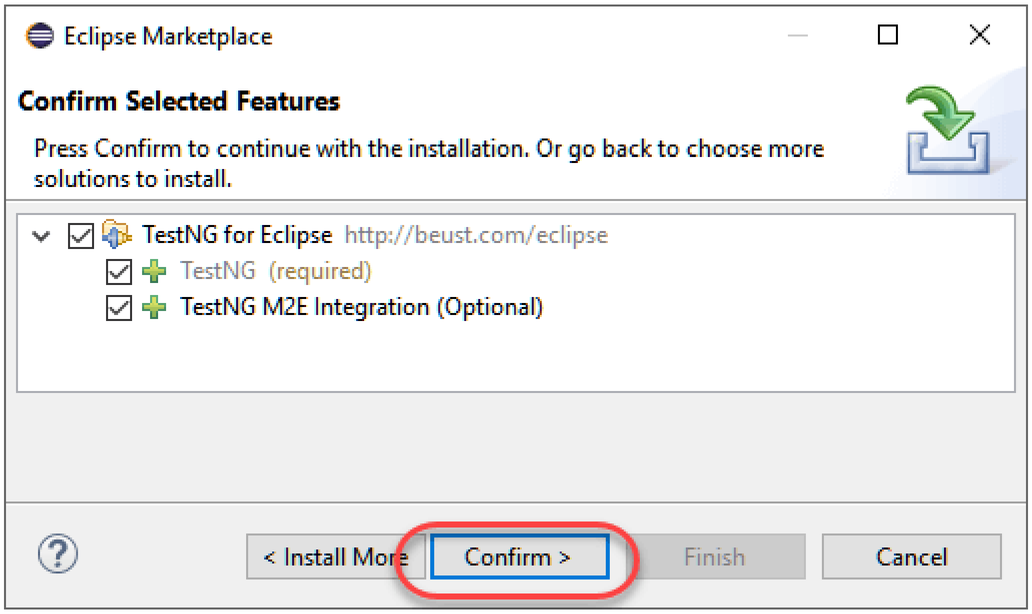
113 — Installs: **433K** (15,844 last month) — Install

Infinitest 5.1.116

Infinitest is a continuous test runner for Java, and is valuable to developers using a unit testing tool called JUnit. Continuous te... the practice of... more info

Marketplaces

Step 4: A new window for feature selection will open, Do not change anything and Click on confirm button


Eclipse Marketplace

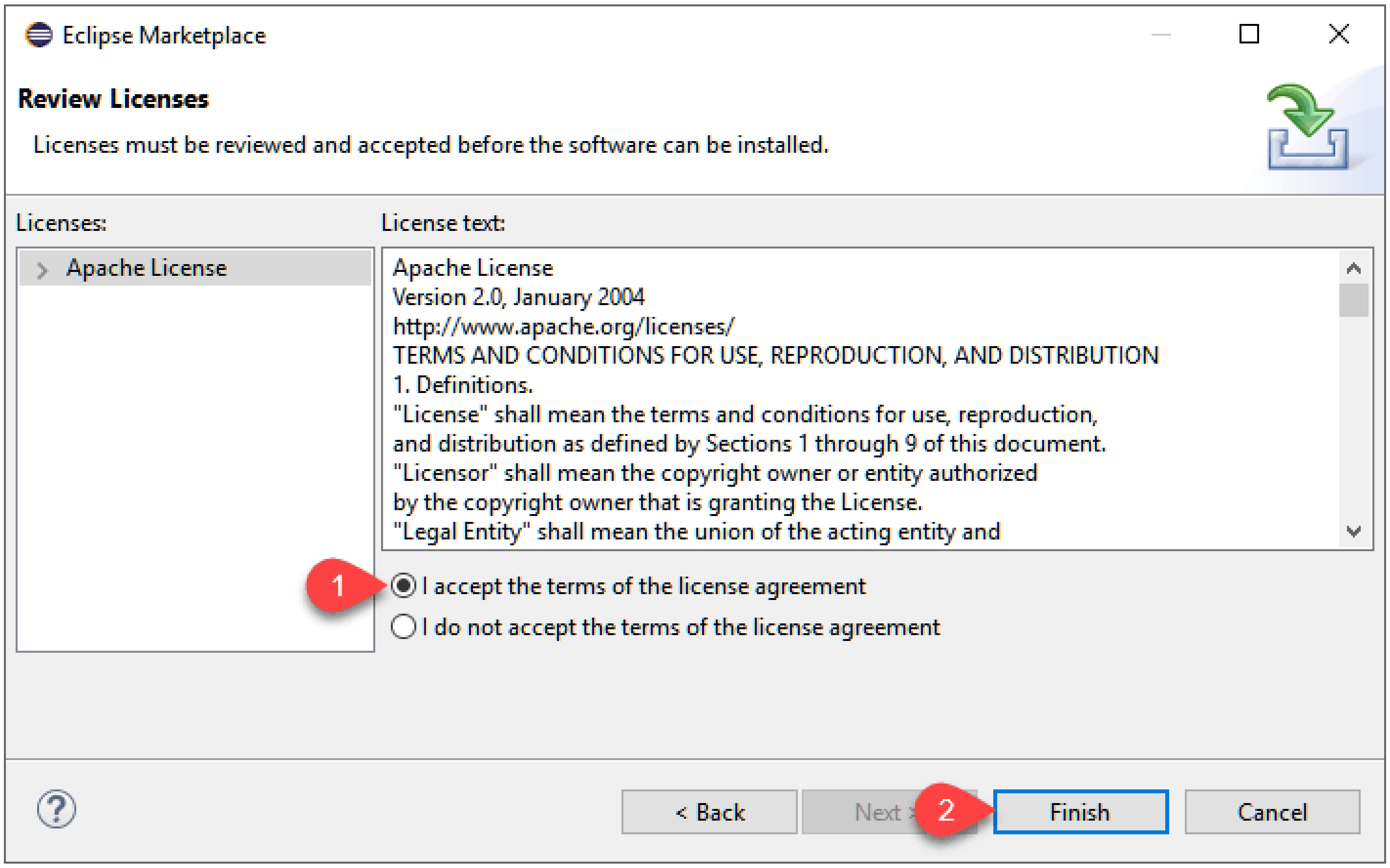
Confirm Selected Features

Press Confirm to continue with the installation. Or go back to choose more solutions to install.

- [✓] TestNG for Eclipse http://beust.com/eclipse
  - [✓] TestNG (required)
  - [✓] TestNG M2E Integration (Optional)

Step 5:

- Click Next again on the succeeding dialog box until you reach the License Agreement dialog.
- Click "I accept the terms of the license agreement" then click Finish.


Eclipse Marketplace

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

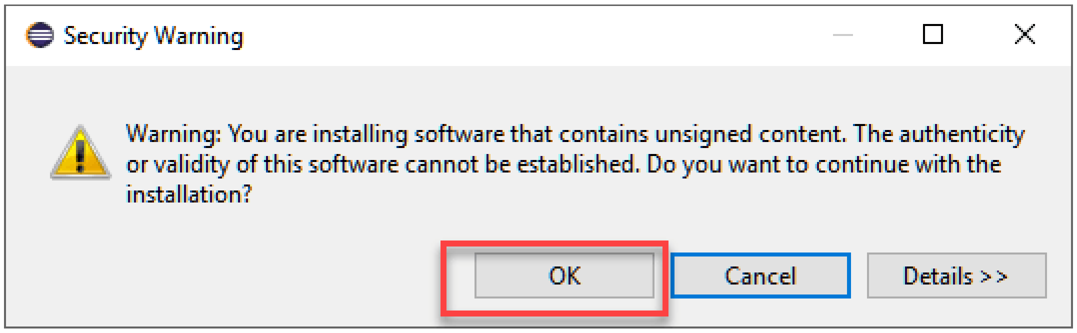
Apache License

License text:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized
by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and

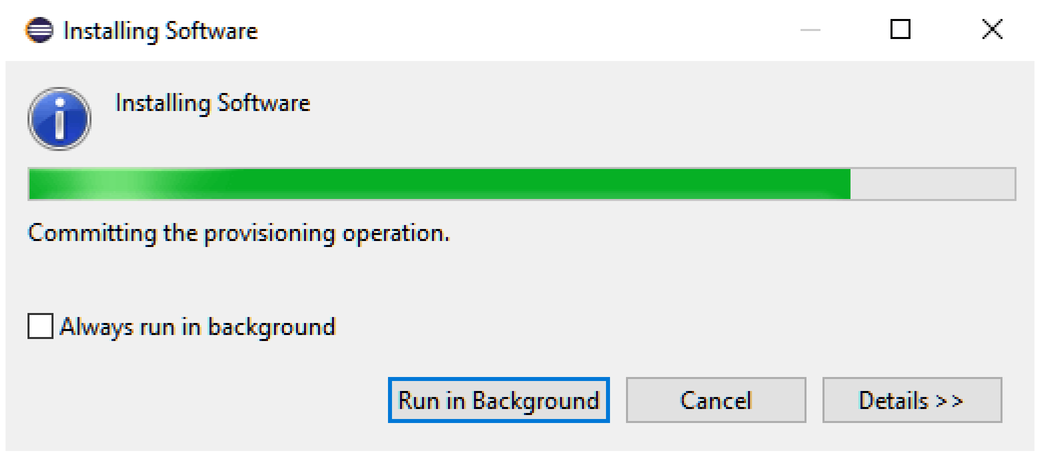
[✓ I accept the terms of the license agreement]
I do not accept the terms of the license agreement

Step 6: If you encounter a Security warning, just click OK

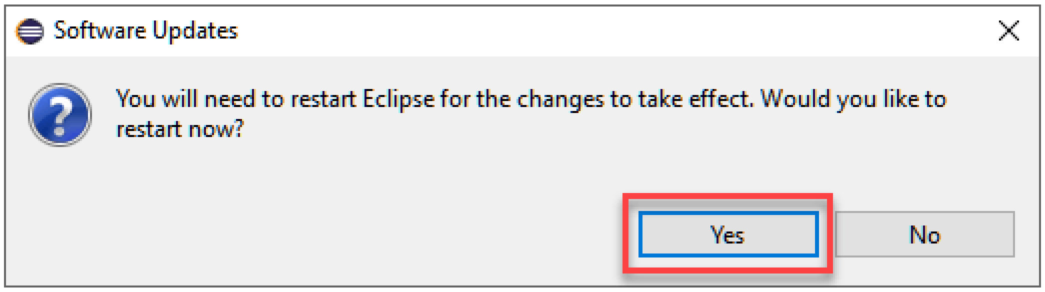

Security Warning

Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

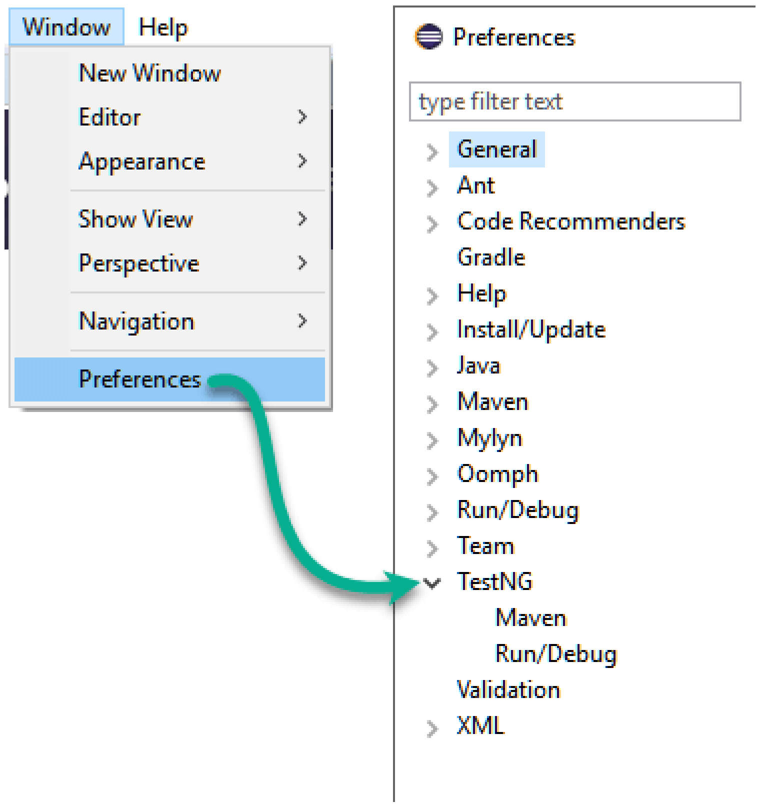
Wait for the installation to finish



Step 7: When Eclipse prompts you for a restart, just click Yes.


Software Updates
You will need to restart Eclipse for the changes to take effect. Would you like to restart now?
Yes
No

Step 8: After the restart, verify if TestNG was indeed successfully installed. Click Window > Preferences and see if TestNG is included on the Preferences list.

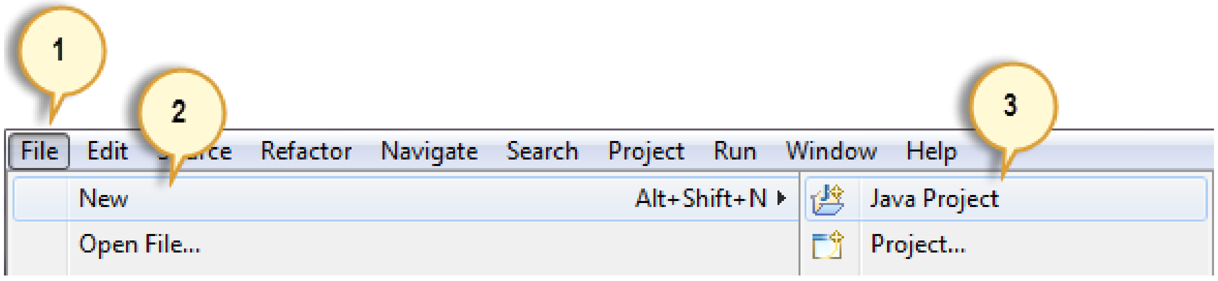


First test case using annotations

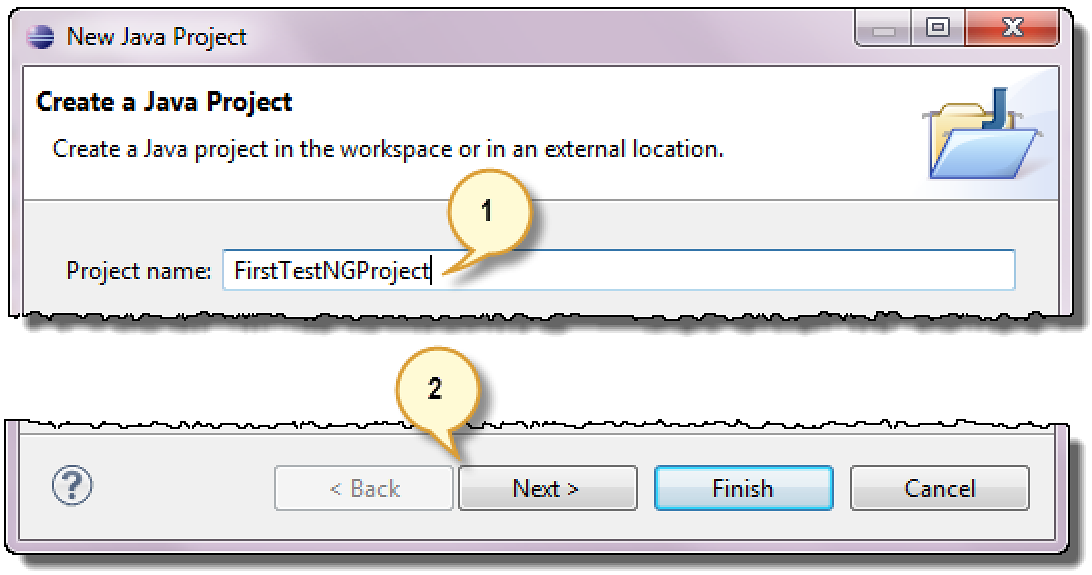
Before we create a test case, we should first setup a new TestNG Project in Eclipse and name it as "FirstTestNGProject".

Setting up a new TestNG Project

Step 1: Click File > New > Java Project

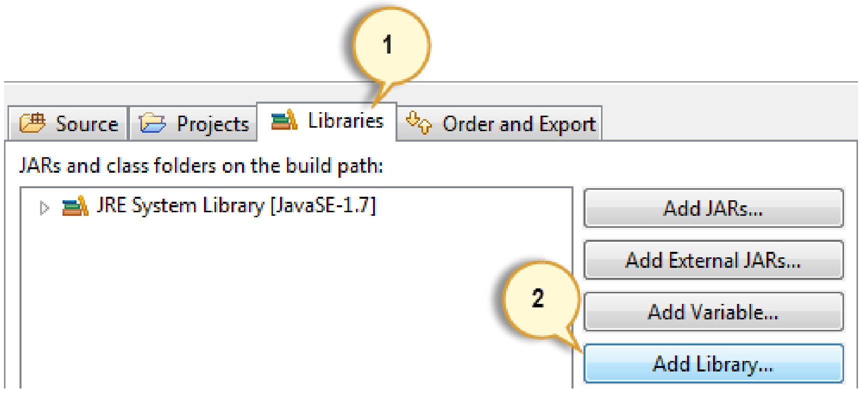

1
2
3
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Java Project
Project...

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

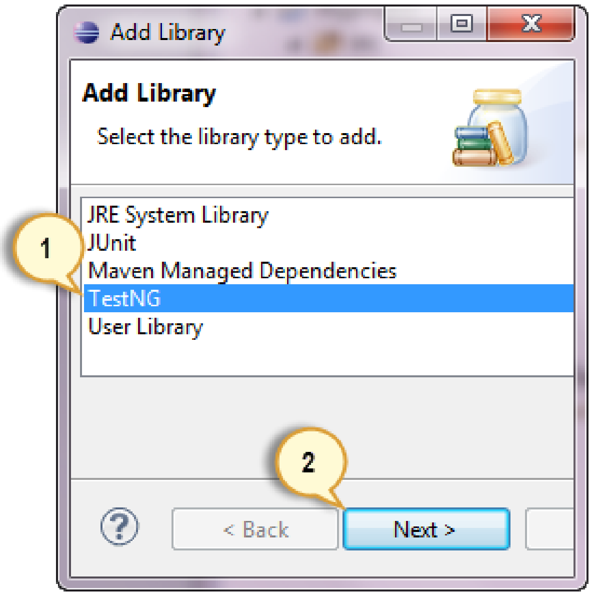

New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
1
2
< Back
Next >
Finish
Cancel

Project name: FirstTestNGProject

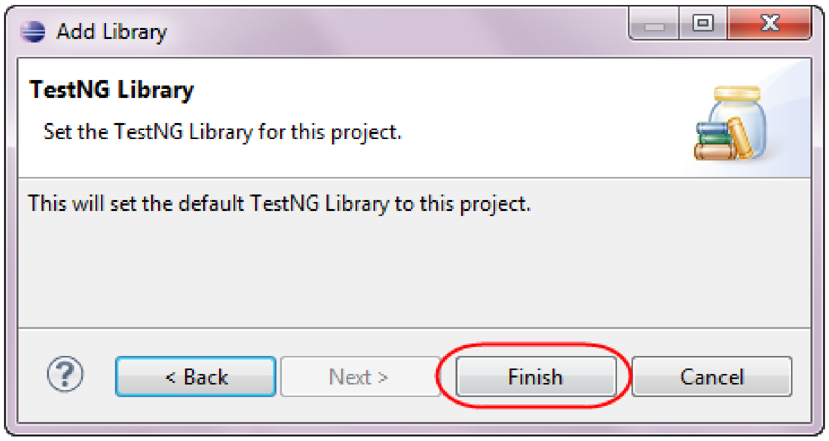
Step 3: We will now start to import the TestNG Libraries onto our project. Click on the "Libraries" tab, and then "Add Library…"



Step 4: On the Add Library dialog, choose "TestNG" and click Next.

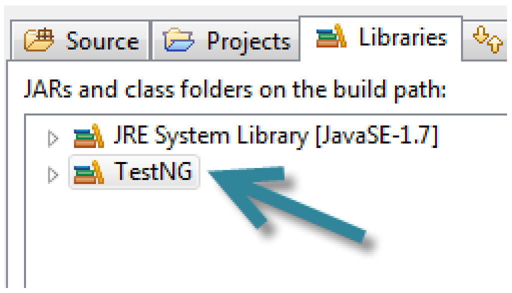


Step 5: Click Finish.

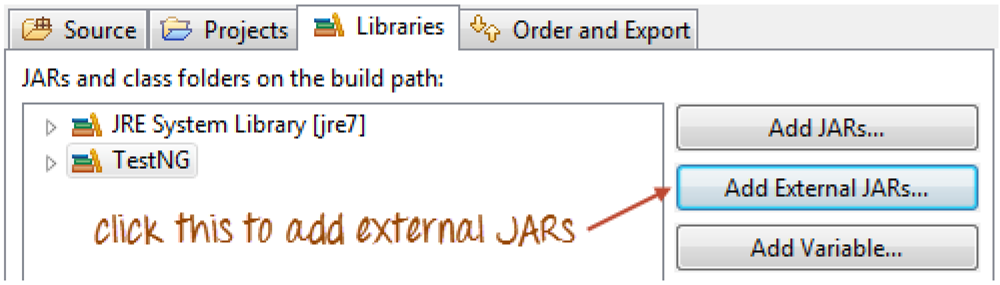


This will set the default TestNG Library to this project.

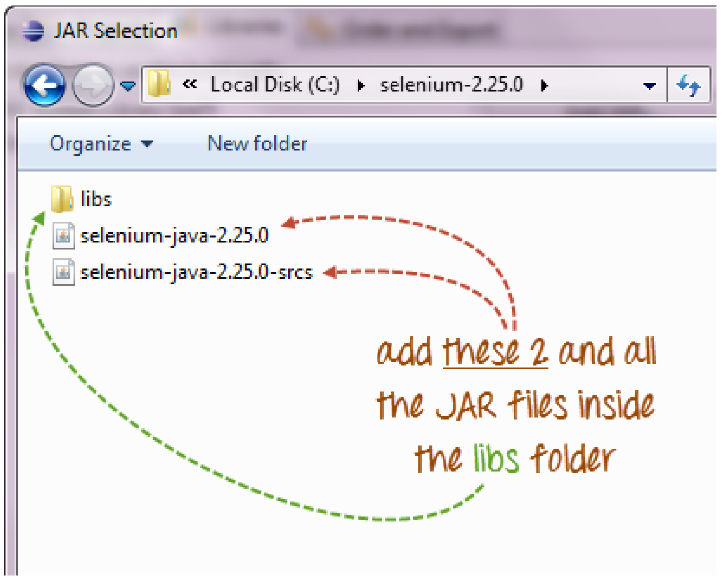
You should notice that TestNG is included on the Libraries list.


Source
Projects
Libraries
JARs and class folders on the build path:
JRE System Library [JavaSE-1.7]
TestNG

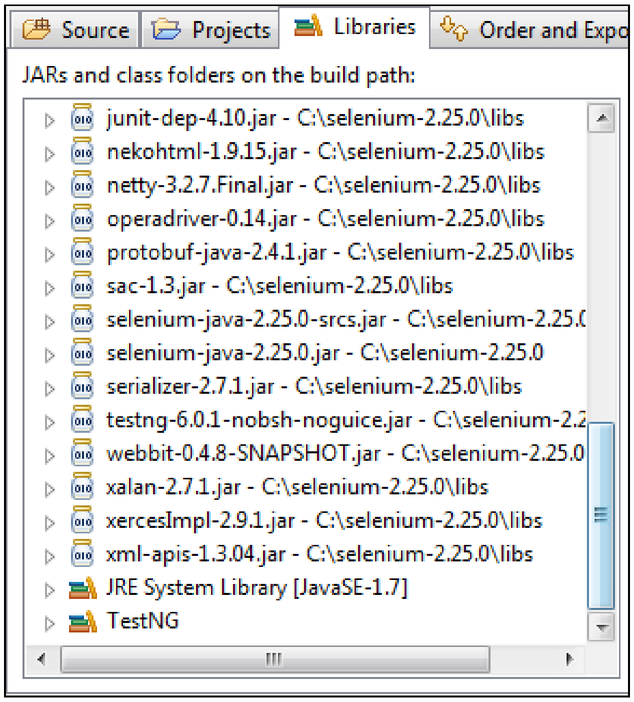
Step 6: We will now add the JAR files that contain the Selenium API. These files are found in the Java client driver that we downloaded from http://docs.seleniumhq.org/download/ when we were installing Selenium and Eclipse in the previous chapters.


Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [jre7]
TestNG
Add JARs...
Add External JARs...
Add Variable...
click this to add external JARS

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


JAR Selection
Local Disk (C:) ▸ selenium-2.25.0 ▸
Organize
New folder
libs
selenium-java-2.25.0
selenium-java-2.25.0-srcs
add these 2 and all the JAR files inside the libs folder

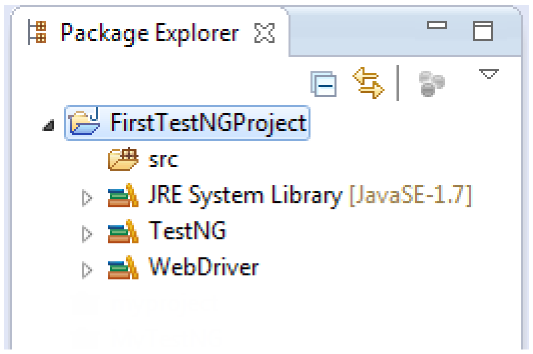
After adding the external JARs, your screen should look like this.



JARs and class folders on the build path:

- junit-dep-4.10.jar - C:\selenium-2.25.0\libs
- nekohtml-1.9.15.jar - C:\selenium-2.25.0\libs
- netty-3.2.7.Final.jar - C:\selenium-2.25.0\libs
- operadriver-0.14.jar - C:\selenium-2.25.0\libs
- protobuf-java-2.4.1.jar - C:\selenium-2.25.0\libs
- sac-1.3.jar - C:\selenium-2.25.0\libs
- selenium-java-2.25.0-srcs.jar - C:\selenium-2.25.0
- selenium-java-2.25.0.jar - C:\selenium-2.25.0
- serializer-2.7.1.jar - C:\selenium-2.25.0\libs
- testng-6.0.1-nobsh-noguice.jar - C:\selenium-2.2
- webbit-0.4.8-SNAPSHOT.jar - C:\selenium-2.25.0
- xalan-2.7.1.jar - C:\selenium-2.25.0\libs
- xercesImpl-2.9.1.jar - C:\selenium-2.25.0\libs
- xml-apis-1.3.04.jar - C:\selenium-2.25.0\libs
- JRE System Library [JavaSE-1.7]
- TestNG

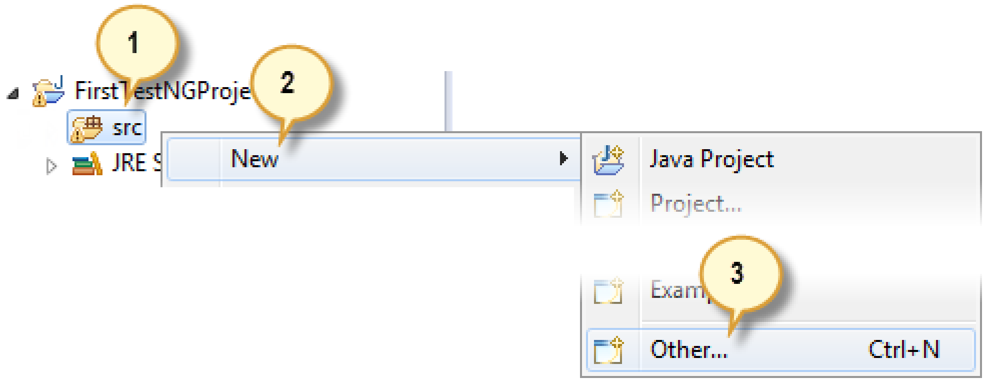
Step 7: Click Finish and verify that our FirstTestNGProject is visible on Eclipse's Package Explorer window.



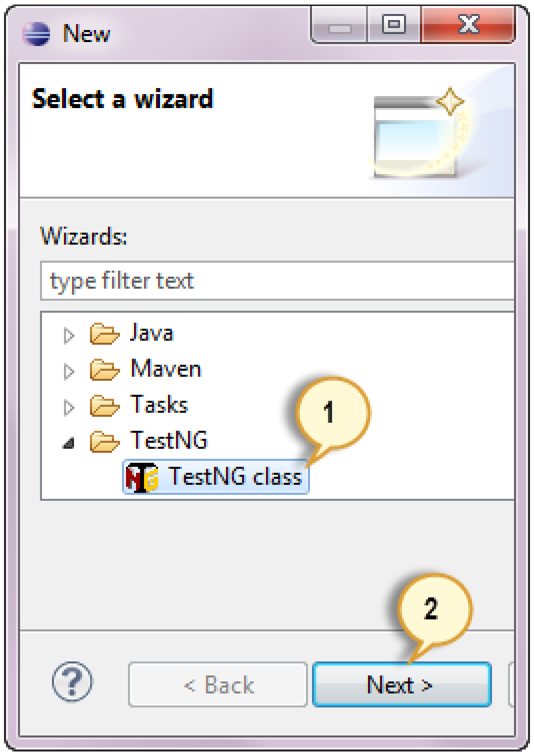
Creating a New TestNG Test File

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

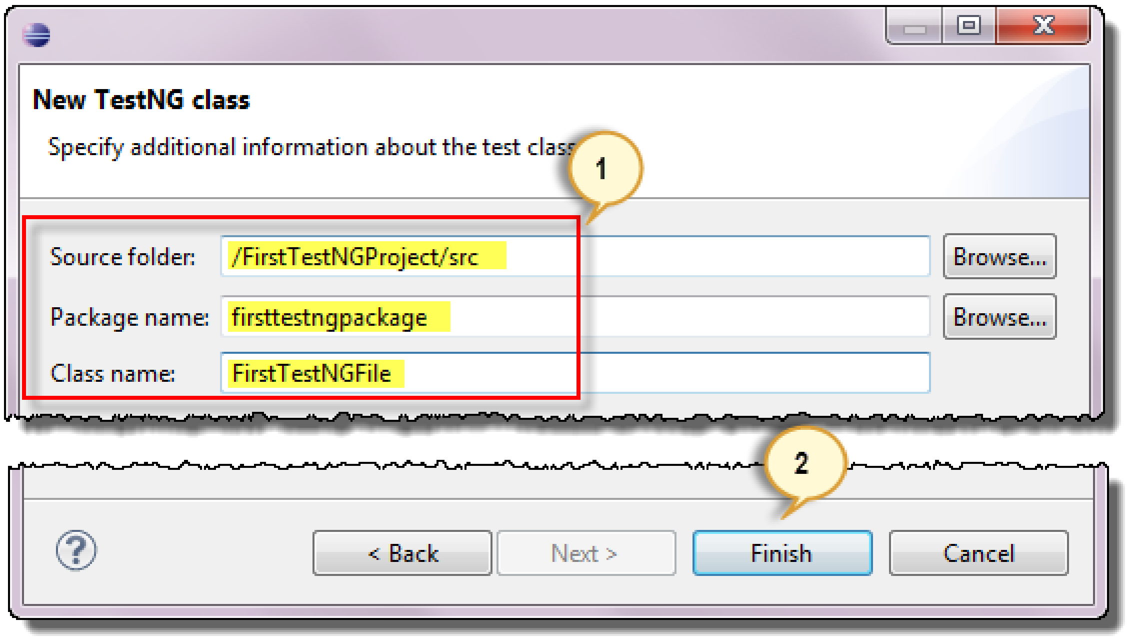
Step 1: Right-click on the "src" package folder then choose New > Other…



Step 2: Click on the TestNG folder and select the "TestNG class" option. Click Next.



Step 3: Type the values indicated below on the appropriate input boxes and click Finish. Notice that we have named our Java file as "FirstTestNGFile".



New TestNG class

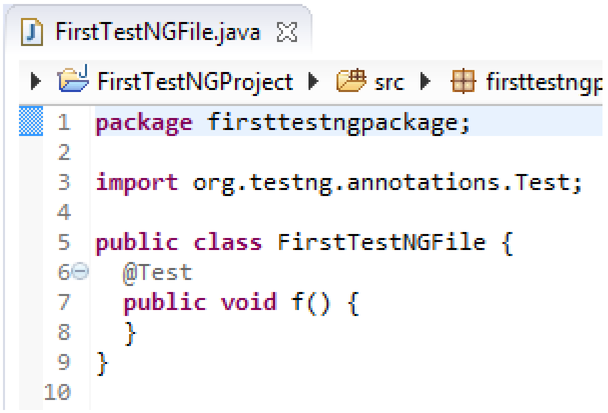
Specify additional information about the test class

Source folder: /FirstTestNGProject/src

Package name: firsttestngpackage

Class name: FirstTestNGFile

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



```java
package firsttestngpackage;

import org.testng.annotations.Test;

public class FirstTestNGFile {
  @Test
  public void f() {
  }
}

```

Annotations in TestNG

@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.

Benefits of using Annotations

1. It identifies the methods it is interested in by looking up annotations. Hence method names are not restricted to any pattern or format.
2. We can pass additional parameters to annotations.
3. Annotations are strongly typed, so the compiler will flag any mistakes right away.
4. Test classes no longer need to extend anything (such as Test Case, for JUnit 3).