Selenium Level 2 Framework Guidelines

Created Date: Oct 26, 2021

Updated Date: -/-

Updated Version: 1.0

Author: Thang Nguyen

LogiGear®

Contents

Automation Environment Setup 3

I. Setup Java 3

II. Setup TestNG 3

Framework Structure 13

1. Common 14
2. PageObjects 15
3. DataObjects 16
4. Testcases 17
5. Executables 17
6. Core 17
7. Grid 18
8. TestData 18
9. TestReport 18

Project Abbreviation 19

There are some abbreviations used to naming locator in project: 19

Run 20

I. Running with Local mode 21

II. Running with Remote mode 22

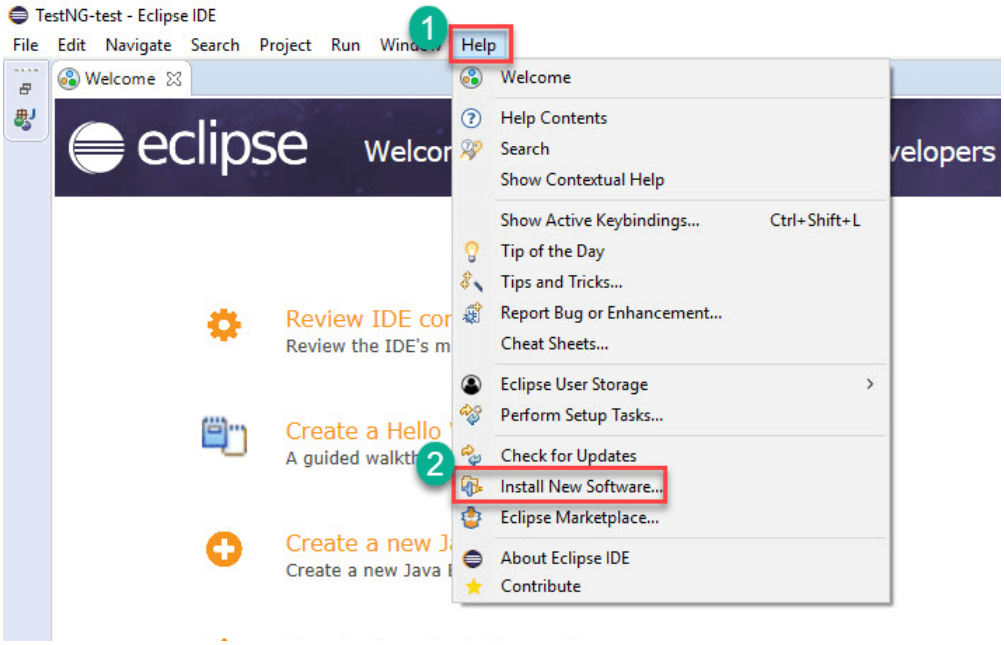
Automation Environment Setup

I. Setup Java

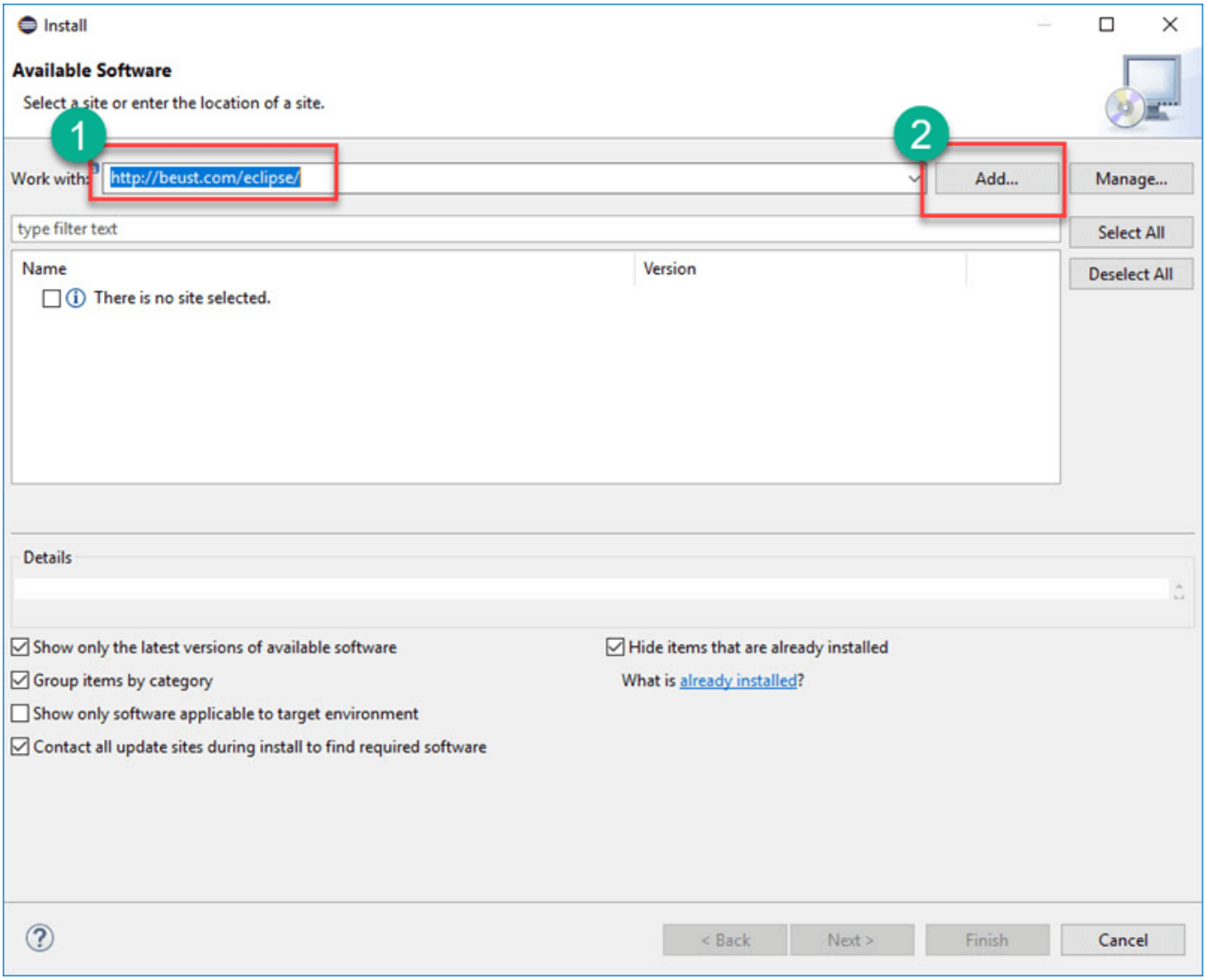
- Window must be installed Java 8
- Link download:
  - JDK (version: 8u301): http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
  - JRE (version: 8u301): http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
- After installing Java, path in system variables (System Properties > Environment Variables) should be " C:\Program Files\Java\jdk1.8.0_301\bin" (window 64 bit)

II. Setup TestNG

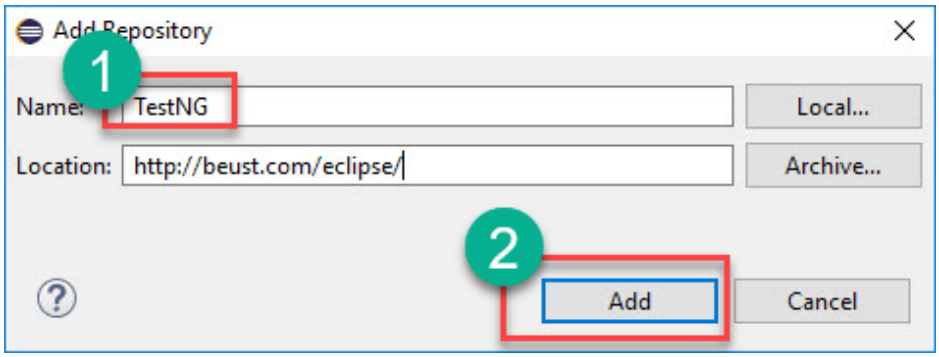
- Launch Esclipse
- On the menu bar > Click on 'Help' > Select 'Install new software option…'



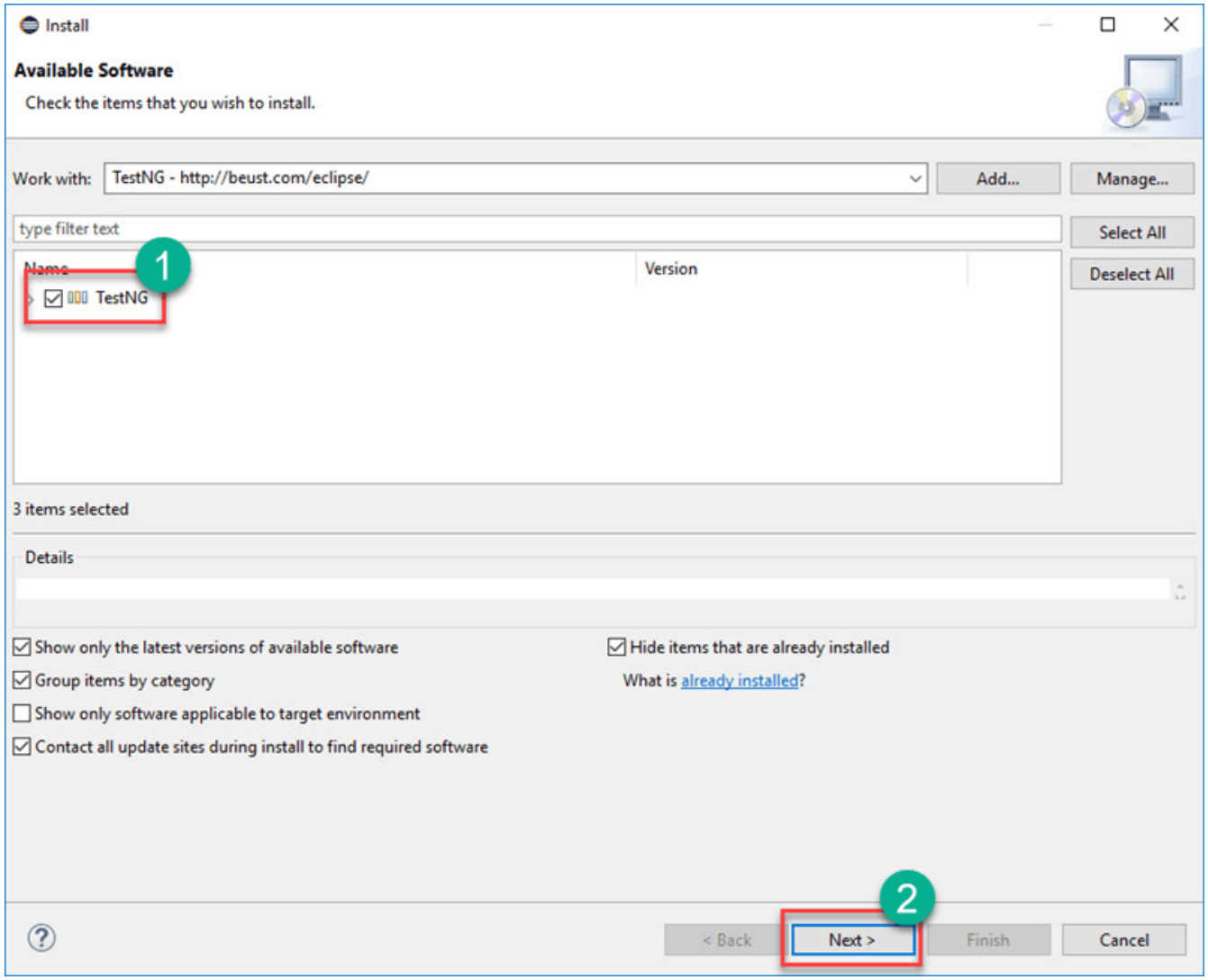
- In the Install dialog box > Enter http://dl.bintray.com/testng-team/testng-eclipse-release/ in the Work with box



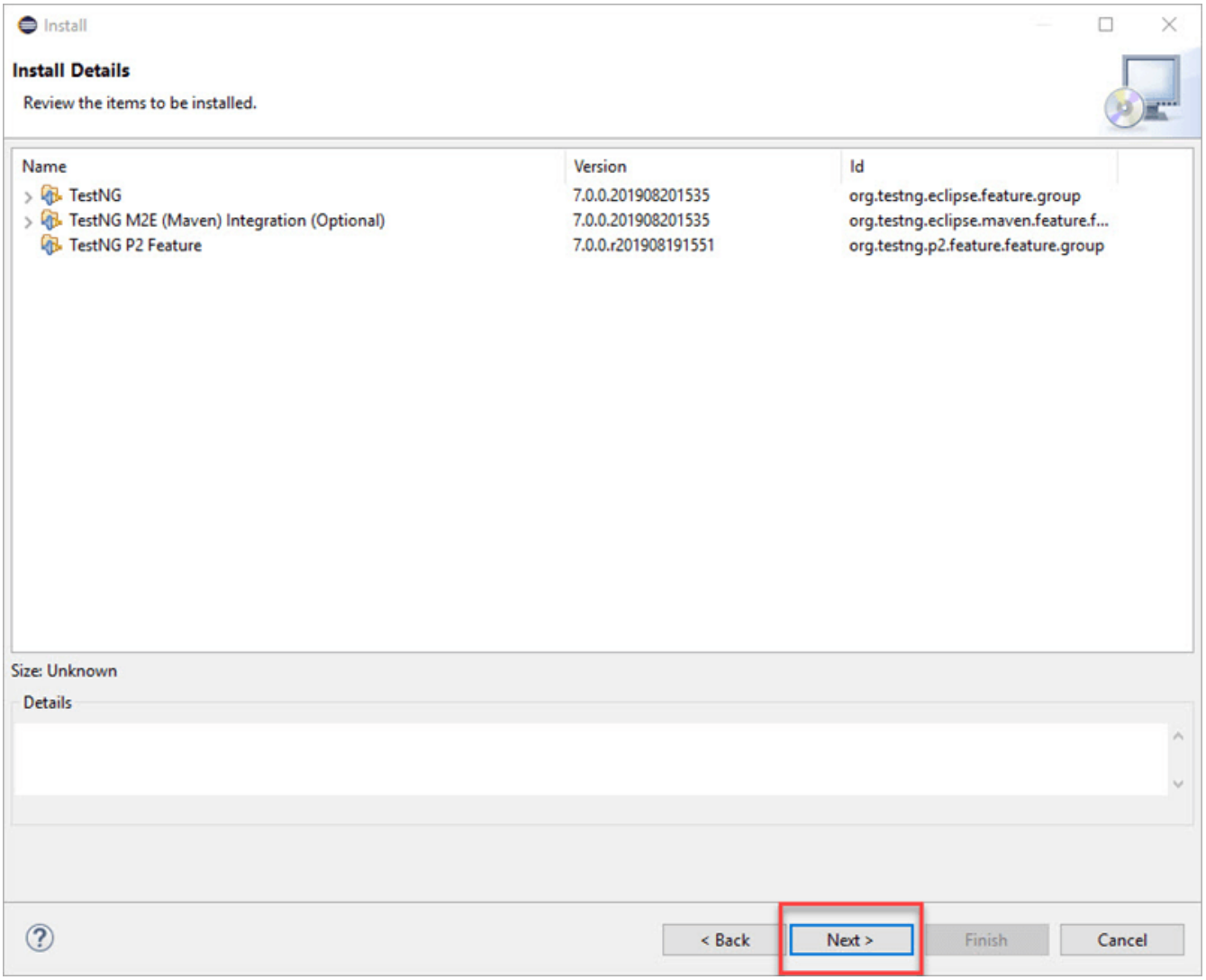
- In Add repository dialog > Enter "TestNG" in Name Text Field > Click on Add button.


Add Repository
1
Name:
TestNG
Local...
Location:
http://beust.com/eclipse/
Archive...
2
Add
Cancel

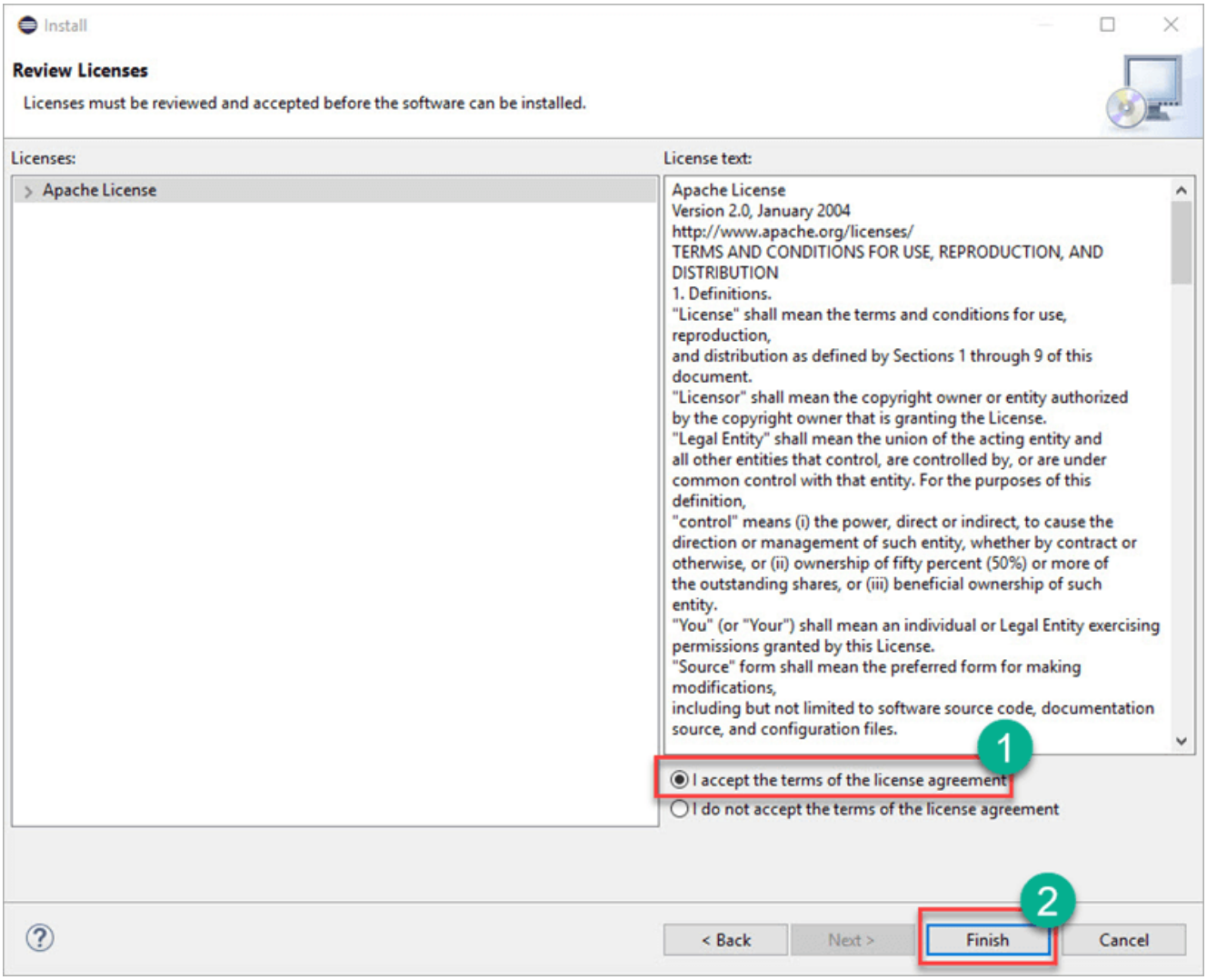
- Select the "TestNG" checkbox > Click on the "Next" button.



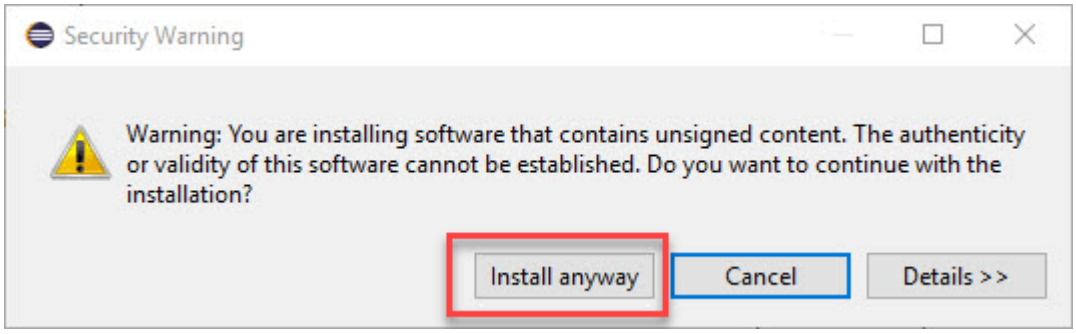
- It will give you a review of Items to be installed. Click on Next.



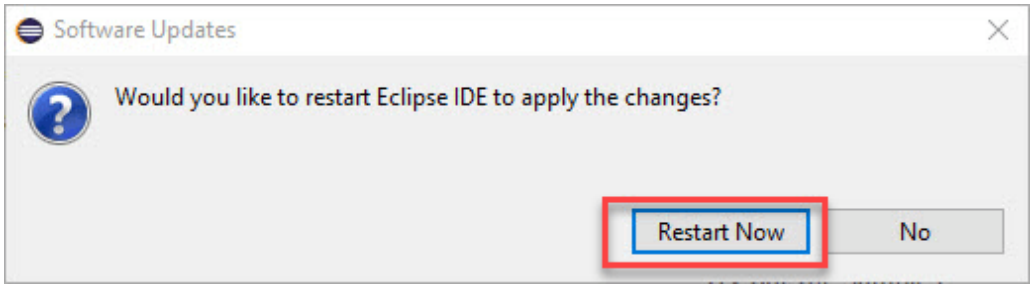
- Select the radio button "I accept the terms of the license agreement" > Click on Finish. Waiting for it. It will take time depending on your Internet speed.



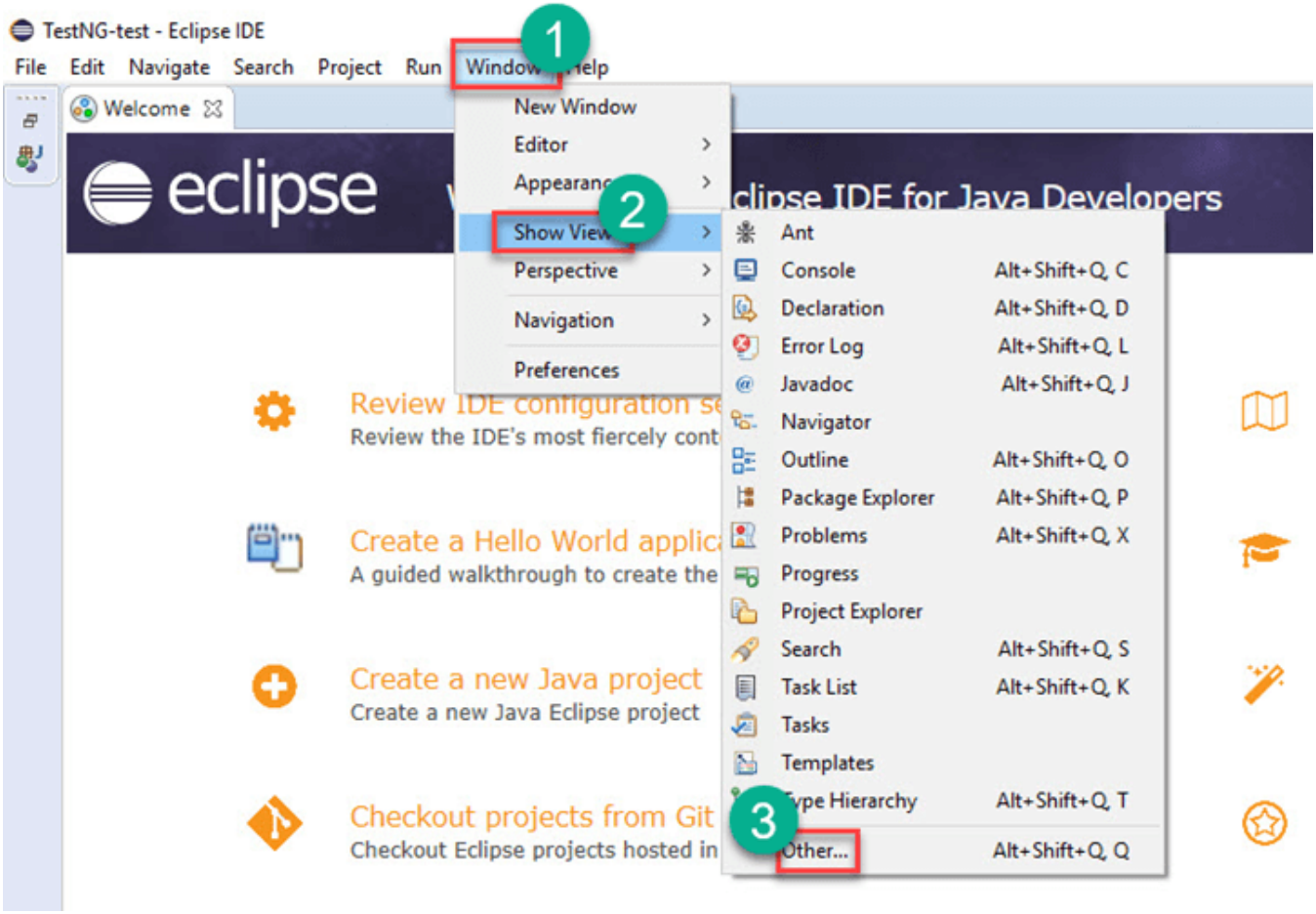
- If you encounter a Security warning, just click "Install Anyway".



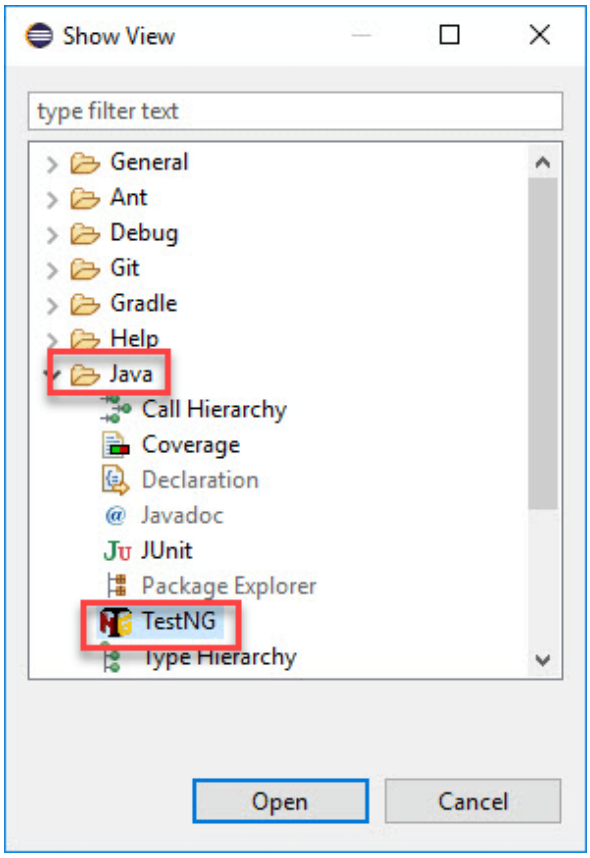
- Wait for the installation to finish. When Eclipse prompts you for a restart, click "Restart now."



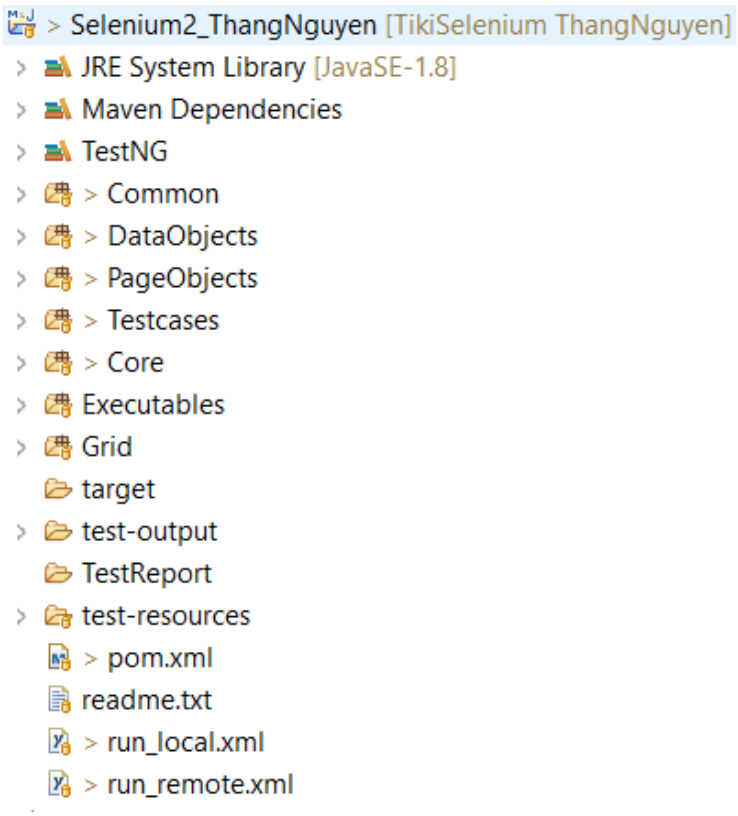
- After the restart, verify if TestNG was indeed successfully installed. Click Window > Show View > Other.



- Then open the Java directory and see if TestNG is included. Now TestNG is installed on you Esclipse

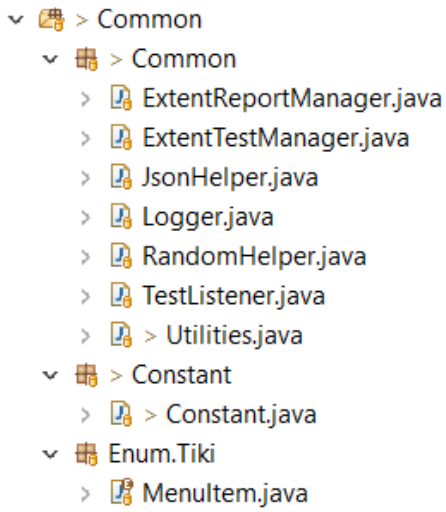


Framework Structure



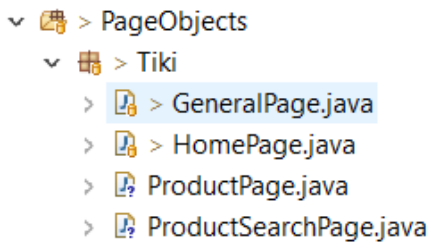
1. **Common**

In the framework we divided the Common into 3 packages: Common, Constant and Enum.



- **Common:** contains Utilities.java, JsonHelper.java, RandomHelper.java and ExtendReport java classes which are often used for generating report HTML and report Logger.
- **Constant:** contains some variable constants defined as URL, wait time out, ...
- **Enum:** contains enums of project

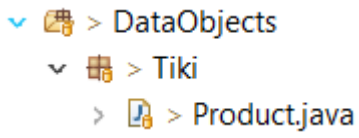
2. PageObjects



- We divided it into two parts pages and dialogs to contain locators and methods corresponding to it's page and it's dialog.

```
public class ProductPage extends GeneralPage {

    private final Element breadCrumbProductName = new Element("//div[@class='breadcrumb']/a
    private final Element lblProductName = new Element("//h1[@class='title']");
    private final Element lblProductPrice = new Element("(//div[contains(@class,'product-p

    public ProductPage waitForLoading(Product product) {
        breadCrumbProductName.getDynamicElement(product.getName()).waitForVisibility(Consta
        lblProductName.waitForVisibility(Constant.DEFAULT_TIMEOUT);
        lblProductPrice.waitForVisibility(Constant.DEFAULT_TIMEOUT);
        return this;
    }

    public String getProductName() {
        return lblProductName.getText();
    }

    public String getProductPrice() {
        return lblProductPrice.getText();
    }

}
```

3. DataObjects

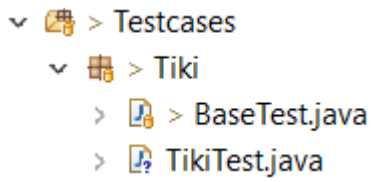


- DataObjects is used for get and set data in project.

```java
public class Product {

    private String name;
    private String price;

    public Product(String name, String price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public String getPrice() {
        return price;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPrice(String price) {
        this.price = price;
    }
```

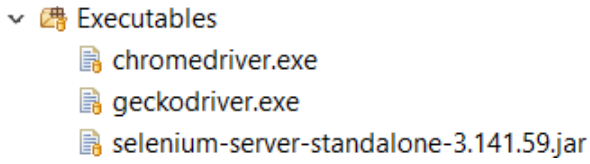
4. **Testcases**

- Contains all test cases.



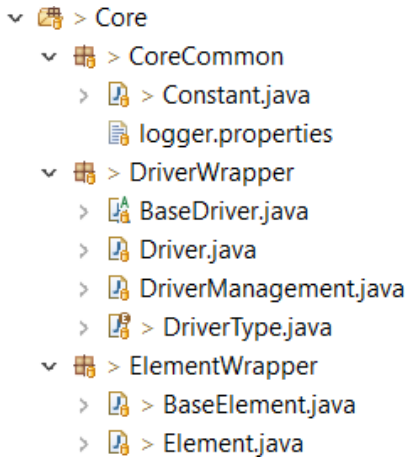
5. **Executables**

- Contains executables as chrome.exe, firefox.exe, selenium-sever-standalone.jar



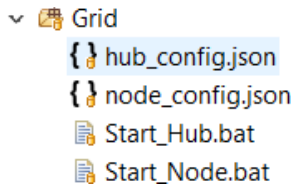
6. **Core**

- Contains Driver Wrapper and Element Wrapper.



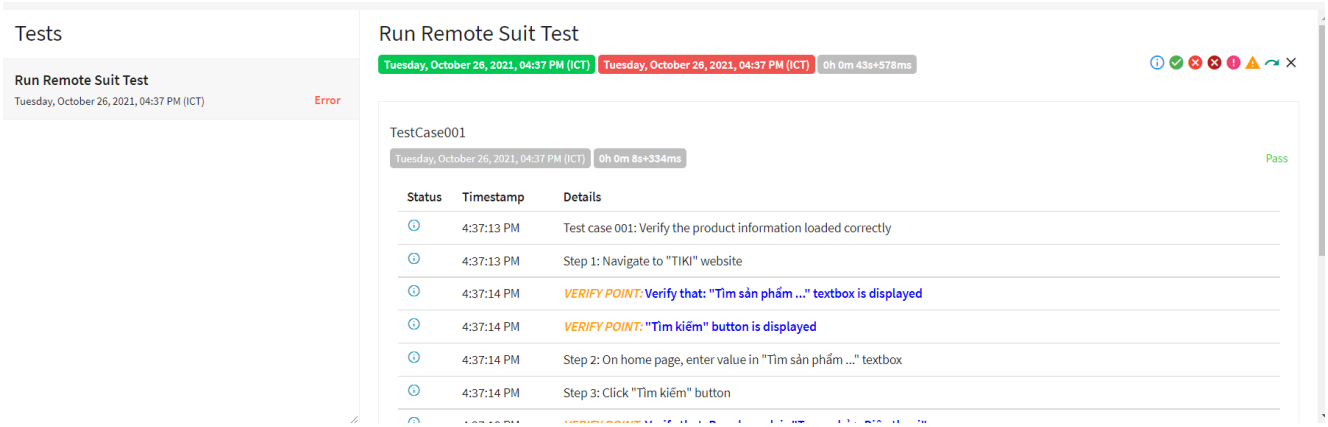
7. Grid

- Contains hub and not bat files which are used for running in remote mode.



8. TestReport

- After running test cases, TestReport and log file will be created automatically if it does not existand contains HTML file report, also screenshot if need.



Project Abbreviation

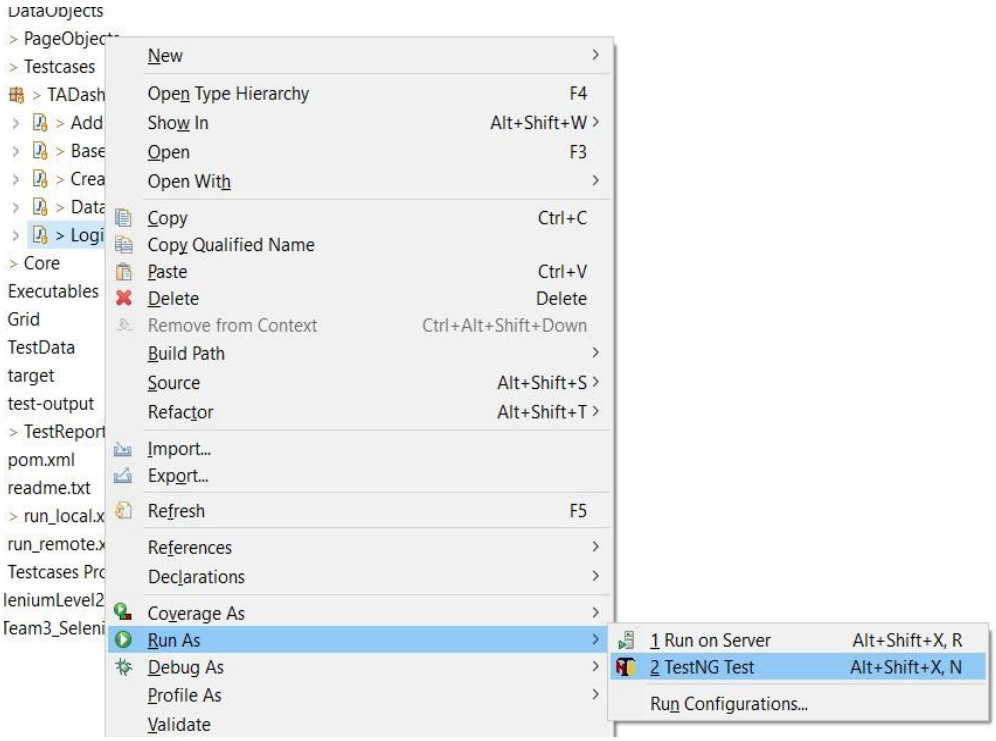
There are some abbreviations used to naming locator in project:

- ***Link***: ***lnk***
- ***Label: lbl***
- ***Textbox: txt***
- ***Button: btn***

Run

You can run test cases by two ways:

- **Basic**: Right click to the test case or test suite you want then click Run as TestNG Test. It's only for running on **Local** and does not support run parallel.



- **Execution files**: There are two files "run-local.xml" to run TCs in **Local** mode and "run-remote.xml" to run TCs in **Remote** mode. They are located in "Selenium1_NganNguyen\run-local.xml" and "Selenium1_NganNguyen\run-remote.xml". Both are also supported parallel test execution.

I. Running with Local mode

To run with Local mode, “run-local.xml” should be configured as the image below then right click the test suite and select Run as TestNG Test.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This file is created for running test suit -->
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<!-- Run parallel with number of threads -->
<suite name="Suite" thread-count="2" parallel="methods">

    <!-- Supported browser names: Chrome; Firefox -->
    <parameter name="broswer" value="Chrome"></parameter>

    <!-- Value parameters: Local, Remote -->
    <parameter name="runningMode" value="Local"></parameter>

    <listeners>
        <listener class-name="Common.TestListener" />
    </listeners>

    <test name="Run Local Suit Test">
        <classes>
            <class name="Tiki.TikiTest"><methods><include name="TestCase001"></include></methods></class>
            <class name="Tiki.TikiTest"><methods><include name="TestCase002"></include></methods></class>
        </classes>
    </test>

</suite> <!-- Suite -->
```

✓ `<parameter name="broswer" value="Chrome"></parameter>`

You can choose browser and replace it in **Value** parameter: Chrome or Firefox (Only support 2 of browses in project).

✓ `<suite name="Suite" thread-count="2" parallel="methods">`

You can set number of threads you want to run: 1,2,3,4, …

II. Running with Remote mode

- Go to “Grid” -> Open file “Start_Hub.bat” to start a host then open “Start_Node.bat” to register a node.
- Configure “run-remote.xml” as the image below then right click the test suite and select Run as TestNG Test.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This file is created for running test suit -->
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<!-- Run parallel with number of threads -->
<suite name="Suite" thread-count="2" parallel="methods">

    <!-- Supported browser names: Chrome; Firefox -->
    <parameter name="broswer" value="Chrome"></parameter>

    <!-- Value parameters: Local, Remote -->
    <parameter name="runningMode" value="Remote"></parameter>

    <listeners>
        <listener class-name="Common.TestListener" />
    </listeners>

    <test name="Run Remote Suit Test">
        <classes>
            <class name="Tiki.TikiTest"><methods><include name="TestCase001"></include></methods></class>
            <class name="Tiki.TikiTest"><methods><include name="TestCase002"></include></methods></class>
        </classes>
    </test>

</suite> <!-- Suite -->
```

✓ `<parameter name="broswer" value="Chrome"></parameter>`

You can choose browser and replace it in **Value** parameter: Chrome or Firefox (Only support 2 of browses in project).

✓ `<suite name="Suite" thread-count="2" parallel="methods">`

You can set number of threads you want to run: 1,2,3,4, …