To implement a java application on Selenium Java driver to search product on amazon.com and scrap it.

This post includes basic description of how selenium web-driver work. I tried to keep this post simpler, so that a new java programmer can start coding to scrap a webpage.

I used eclipse and maven to build this java application, but same approach can be applied with other IDEs.

Prerequisites:

Jdk1.8 (http://www.oracle.com/technetwork/java/javase/downloads/index.html?lipi=urn%3Ali%3Apage%3Ad_flagship3_pulse_read%3Be3kIHb3SSbeWHNi6UqCMIg%3D%3D)

Eclipse IDE (http://www.eclipse.org/downloads/eclipse-packages/)

Firefox browser preferably version > 46.0

Geckodriver.exe-if using Selenium-Java webdriver version 3.0 or higher, and firefox web browser. (https://github.com/mozilla/geckodriver/releases?lipi=urn%3Ali%3Apage%3Ad_flagship3_pulse_read%3Be3kIHb3SSbeWHNi6UqCMIg%3D%3D)

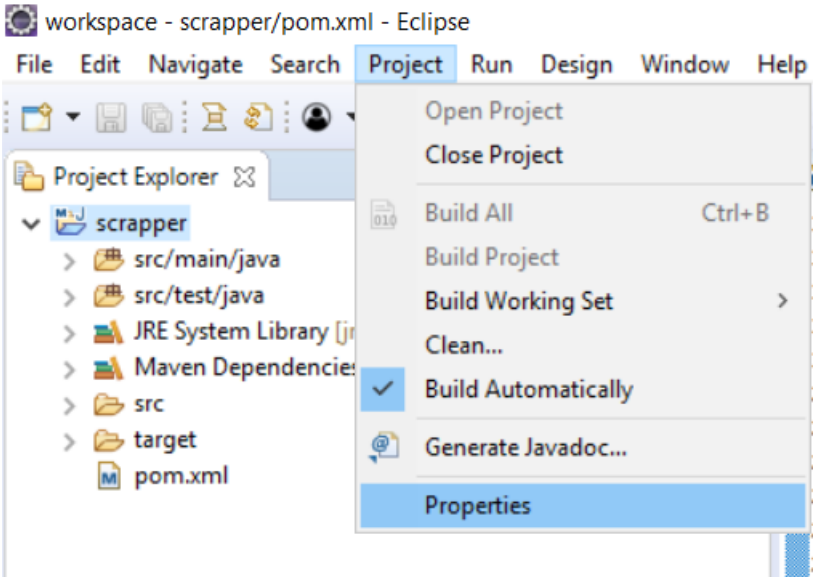
Let's start building your first scrap code

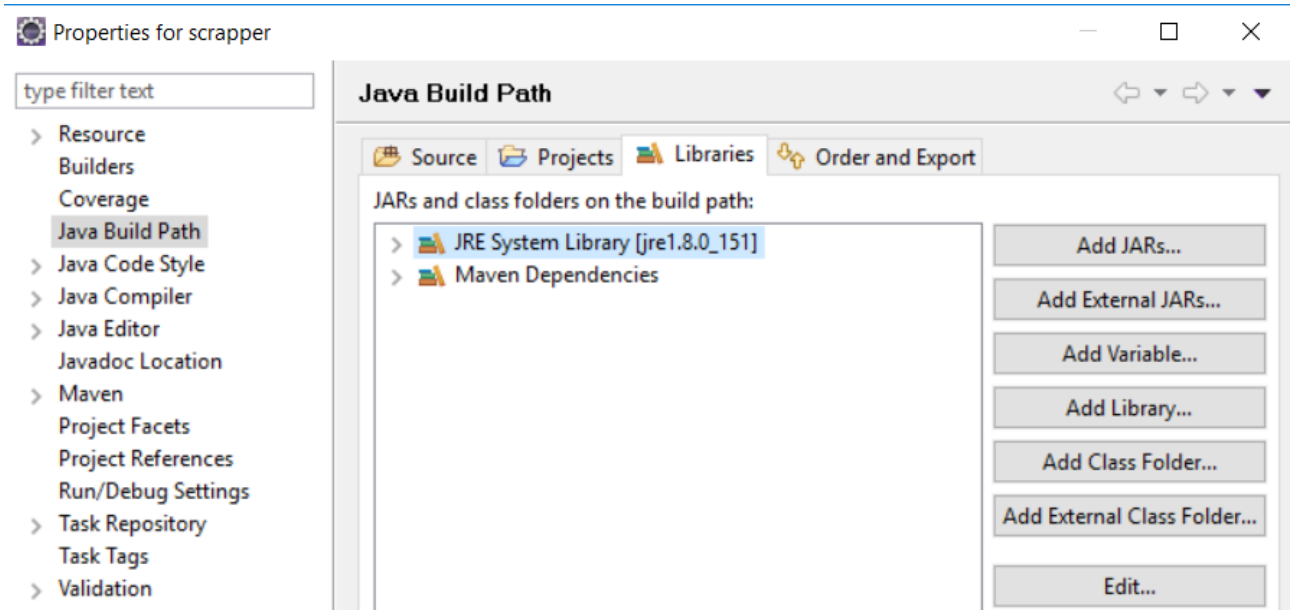
1 Configure Maven in Eclipse:

Open Eclipse and Click on Help -> Eclipse Marketplace

In the search field, search with text maven and select m2e the click install and complete installation

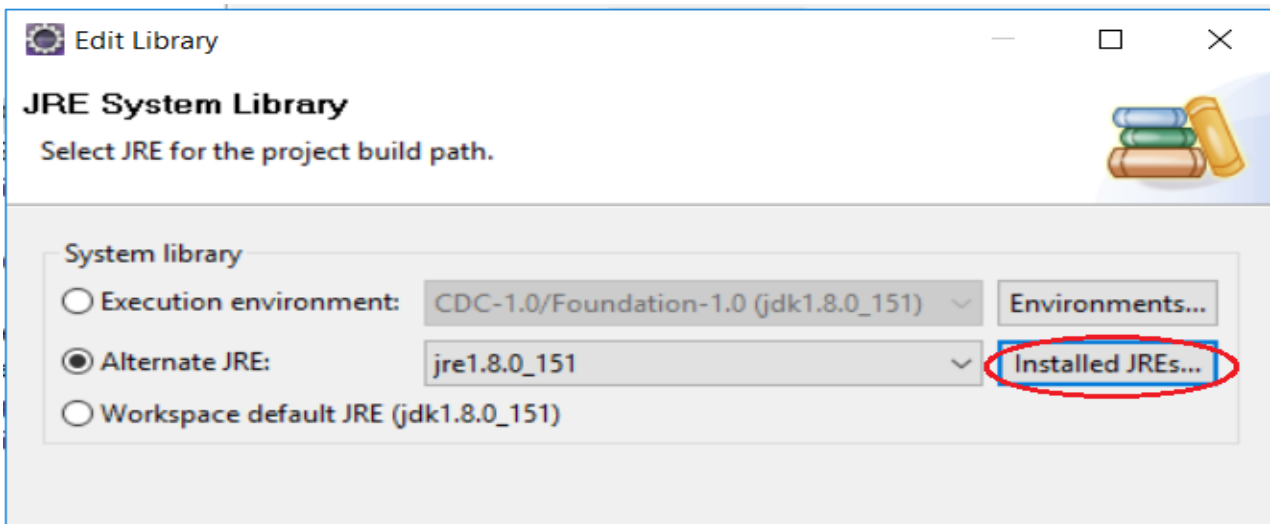
2 Check Java Build Path





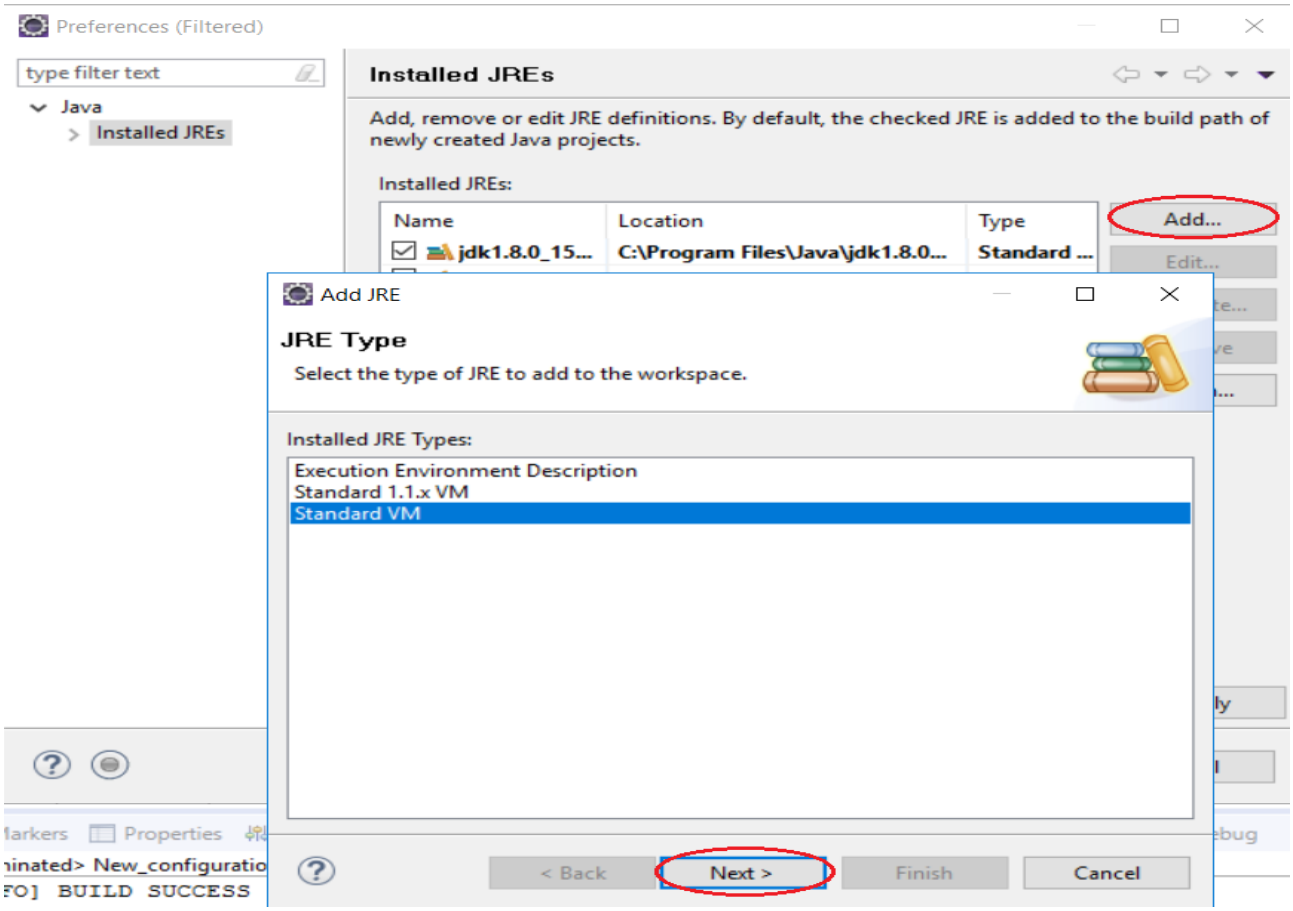
2.1 Verify JRE version

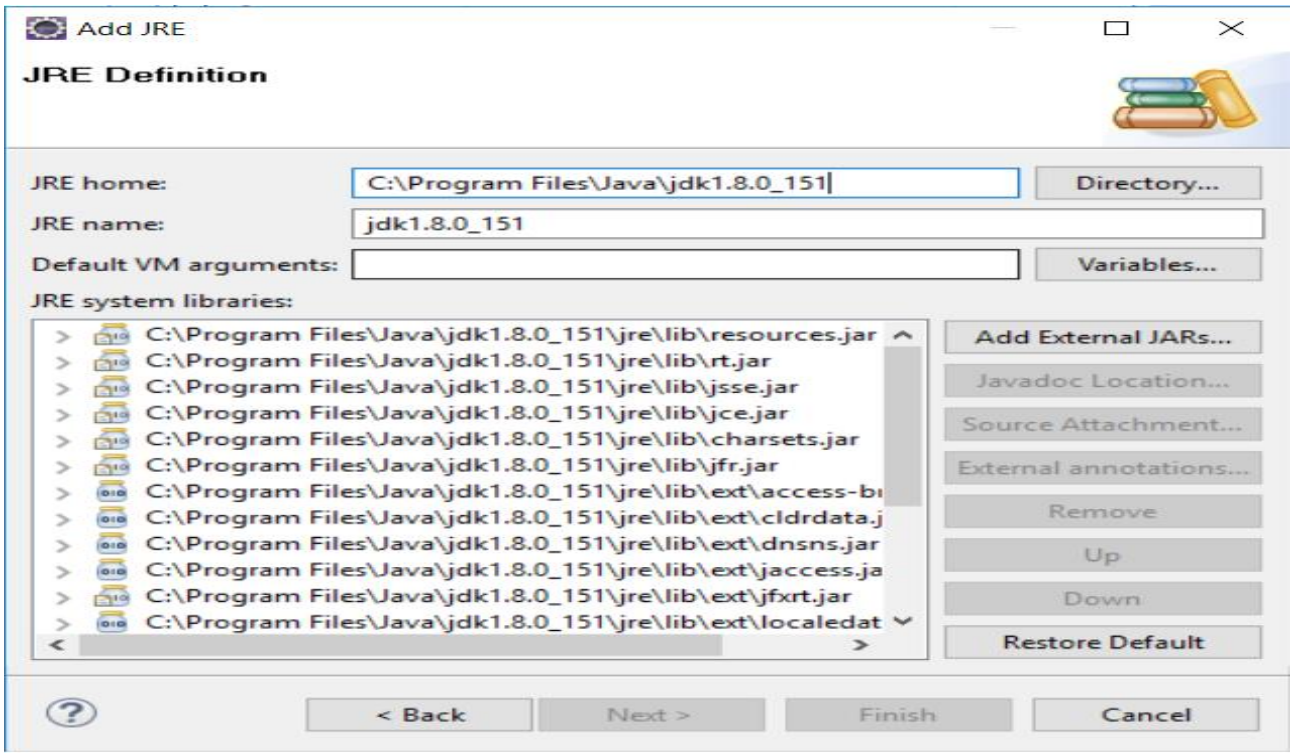
If JRE version is not correct, which you are expecting, edit it here.



Note: I will always be better to given JDK path, instead of JRE path.

C:\Program Files\Java\jdk1.8.0_151

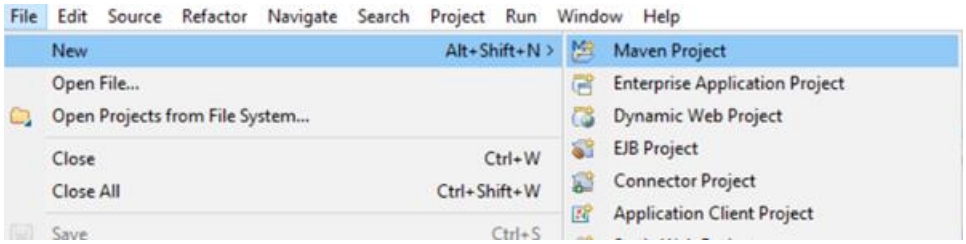

Preferences (Filtered)
type filter text
Java
Installed JREs
Installed JREs
Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.
Installed JREs:
Name
Location
Type
jdk1.8.0_15...
C:\Program Files\Java\jdk1.8.0...
Standard ...
Add...
Edit...
Add JRE
JRE Type
Select the type of JRE to add to the workspace.
Installed JRE Types:
Execution Environment Description
Standard 1.1.x VM
Standard VM
< Back
Next >
Finish
Cancel


Add JRE
JRE Definition
JRE home:
C:\Program Files\Java\jdk1.8.0_151
Directory...
JRE name:
jdk1.8.0_151
Default VM arguments:
Variables...
JRE system libraries:
C:\Program Files\Java\jdk1.8.0_151\jre\lib\resources.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\rt.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\jsse.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\jce.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\charsets.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\jfr.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\ext\cldrdata.j
C:\Program Files\Java\jdk1.8.0_151\jre\lib\ext\dnsns.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\ext\jaccess.ja
C:\Program Files\Java\jdk1.8.0_151\jre\lib\ext\jfxrt.jar
C:\Program Files\Java\jdk1.8.0_151\jre\lib\ext\localedat
Add External JARs...
Javadoc Location...
Source Attachment...
External annotations...
Remove
Up
Down
Restore Default
< Back
Next >
Finish
Cancel

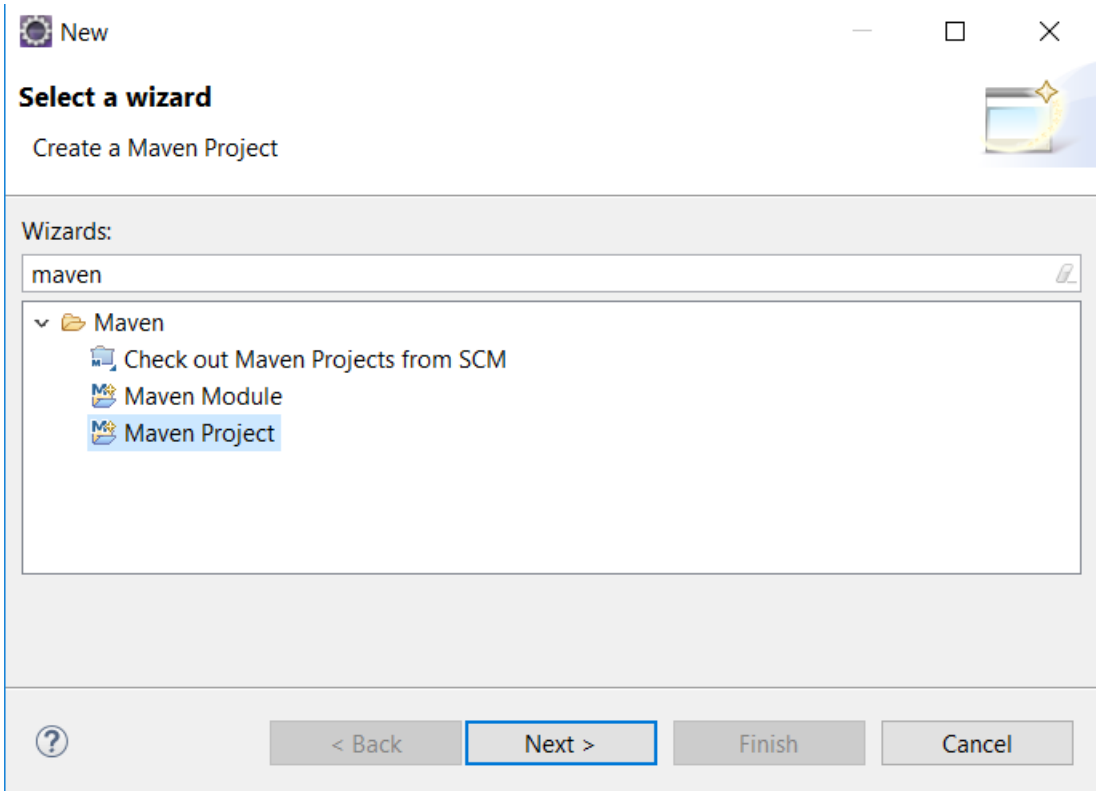
3 Create new Maven Project:

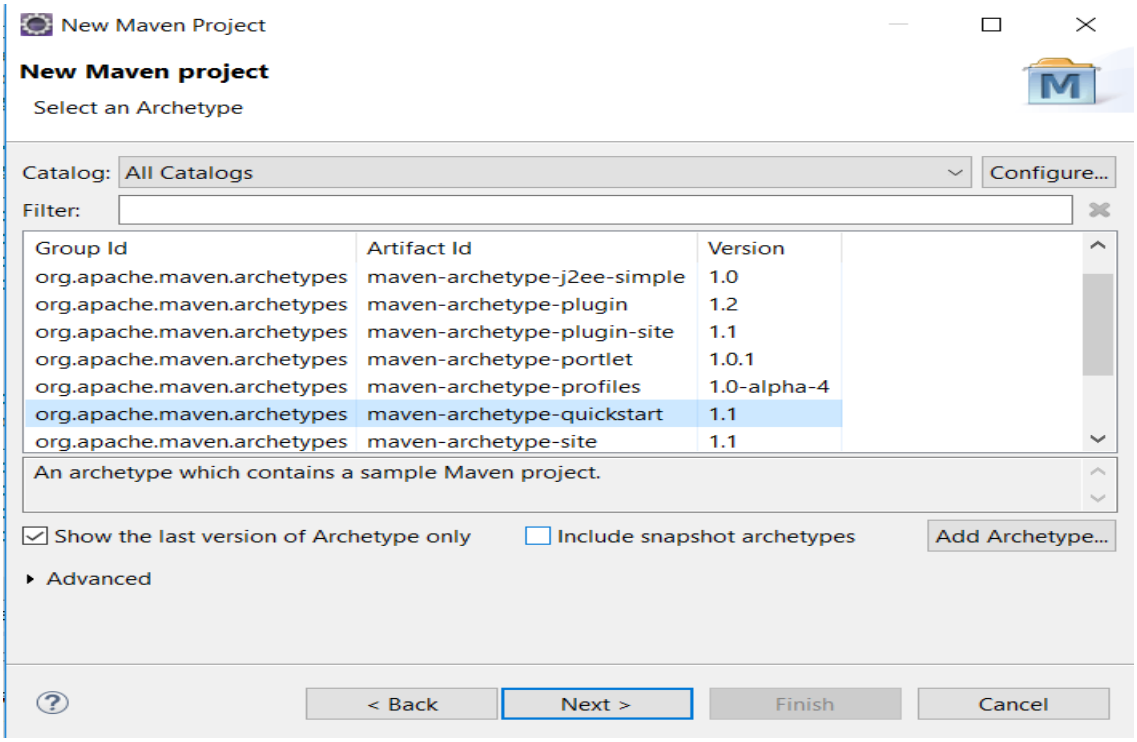
Click on File --> New --> Project

Select Maven --> Maven Project option and click on Next as shown below

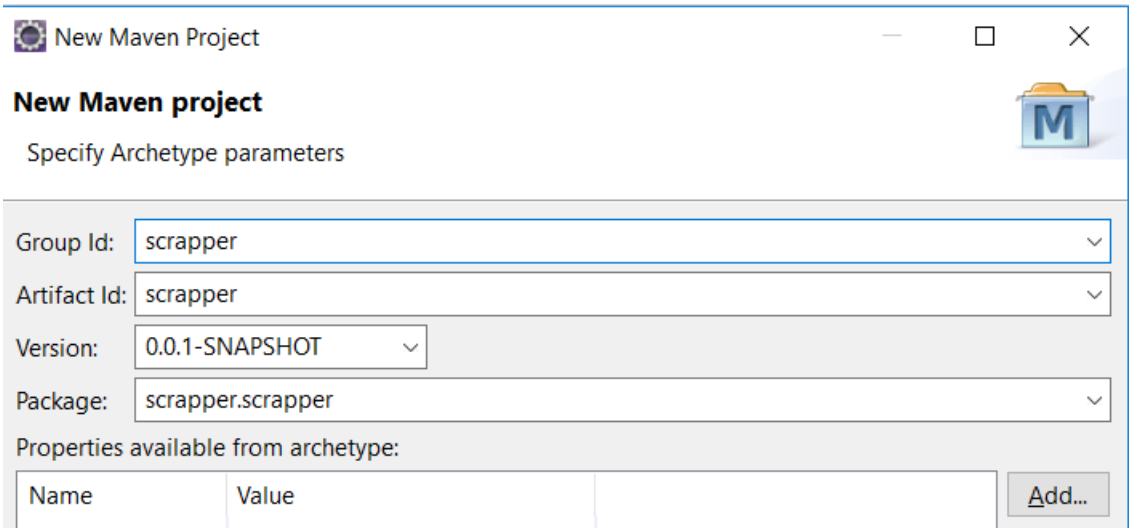


Or Select from wizard

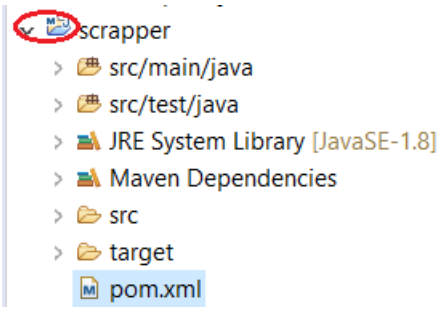




Set GroupId and ArtifactId with meaningful name.



Maven Project created successfully. You can verify ‘M’ icon on project folder.



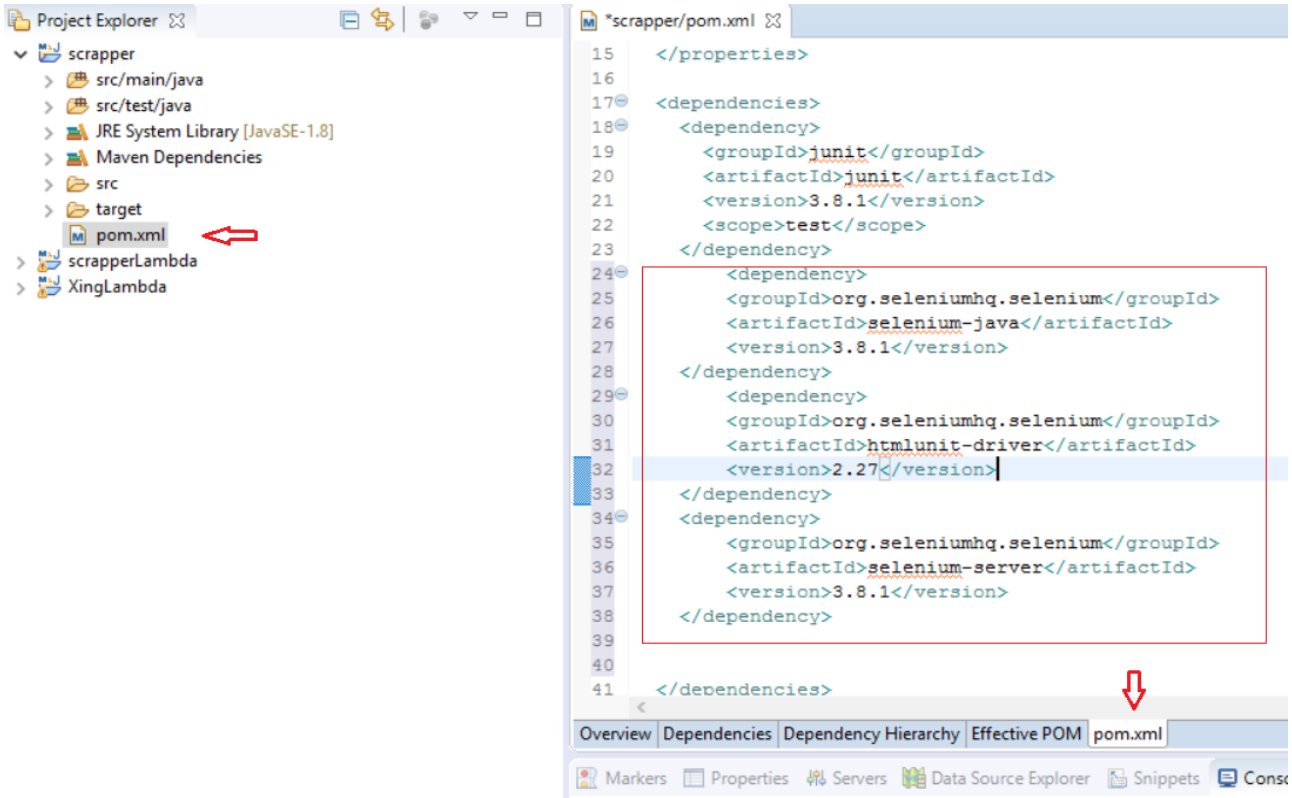
4 Add dependency in pom.xml

Search for latest version of jar

Open https://mvnrepository.com/ and search for dependencies.

```
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.8.1</version>
</dependency>
    <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>htmlunit-driver</artifactId>
    <version>2.27</version>
</dependency>

<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-server</artifactId>
    <version>3.8.1</version>
</dependency>
```

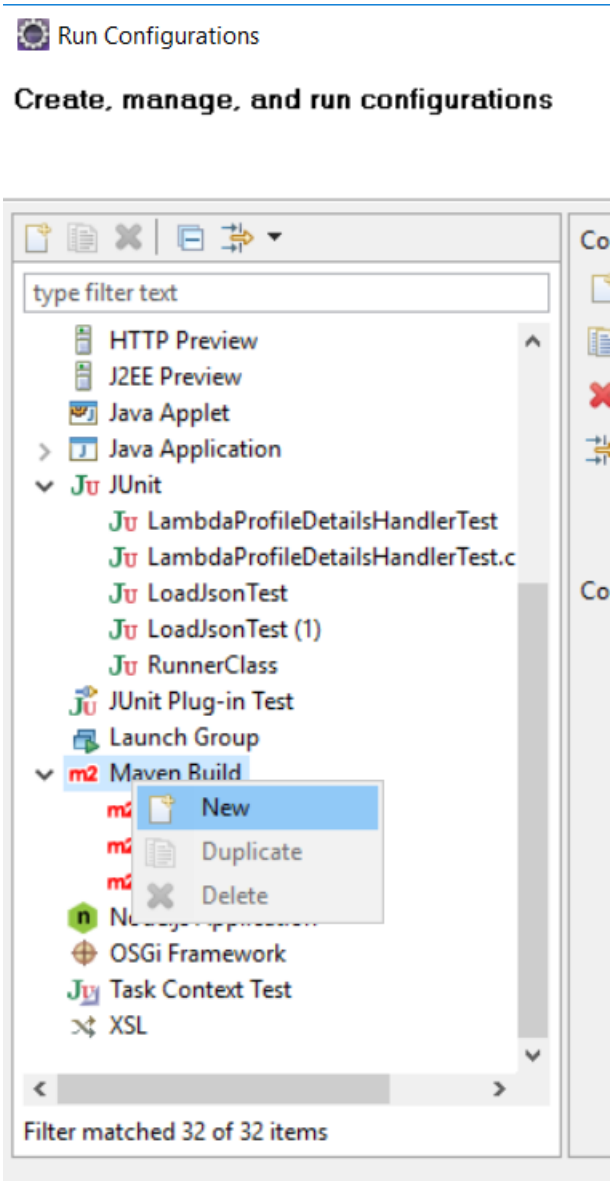
Open pom.xml and add above code as below:



5 Run Maven build

Select pom.xml, right click-> Run As -> Run Configuration

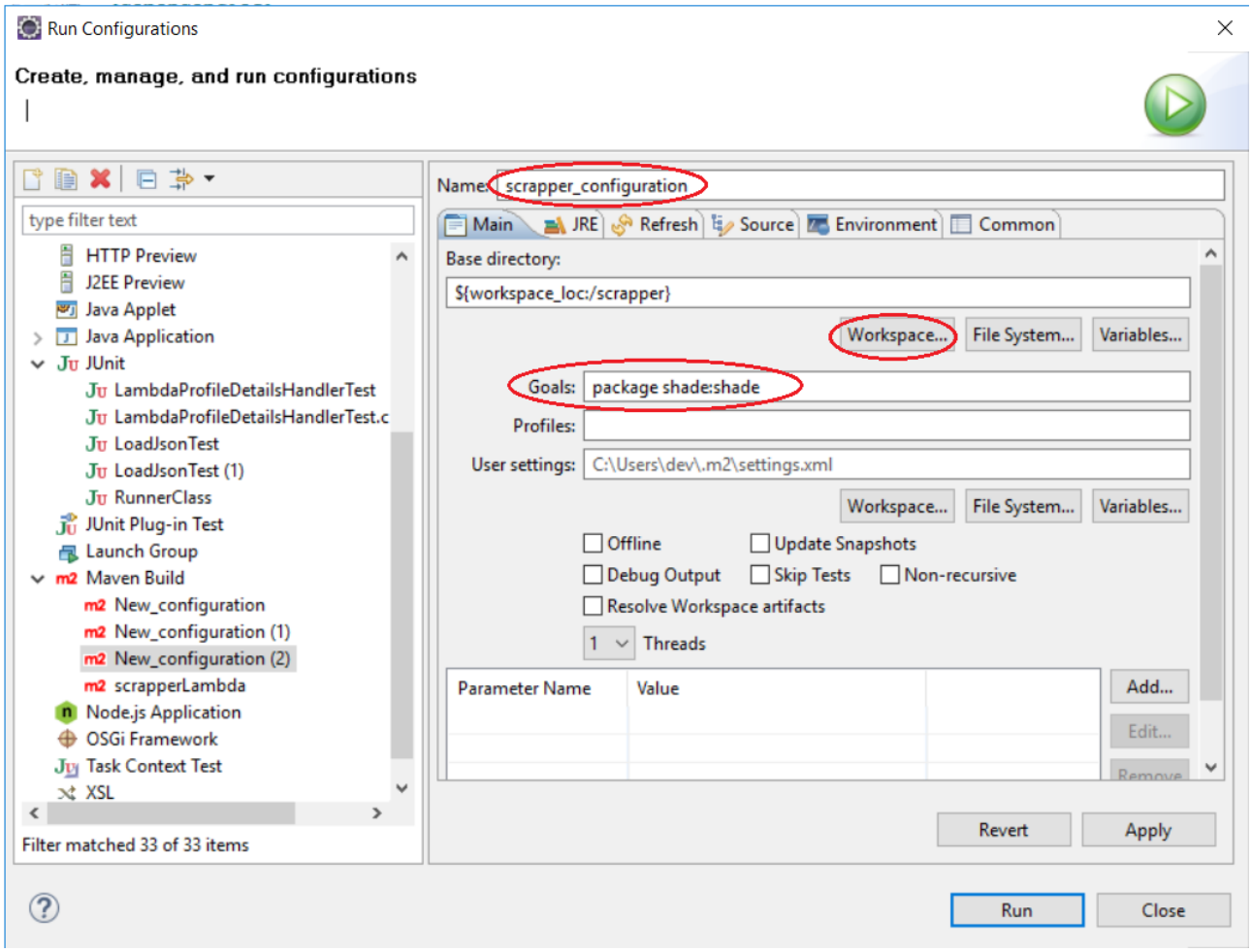
Create new Configuration



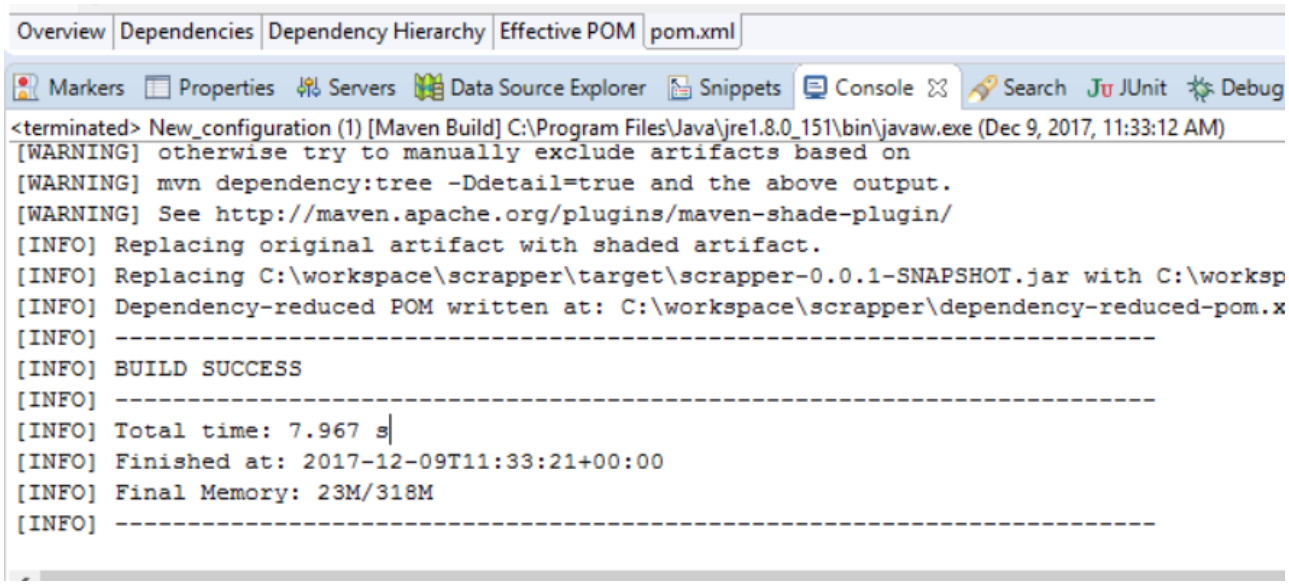
Put meaningfull configuration name.

Select you project from workspace.

You can set Goal as: **package shade:shade.** It will also help you to create jar file for further deployment. For more option in Maven please check on its site.



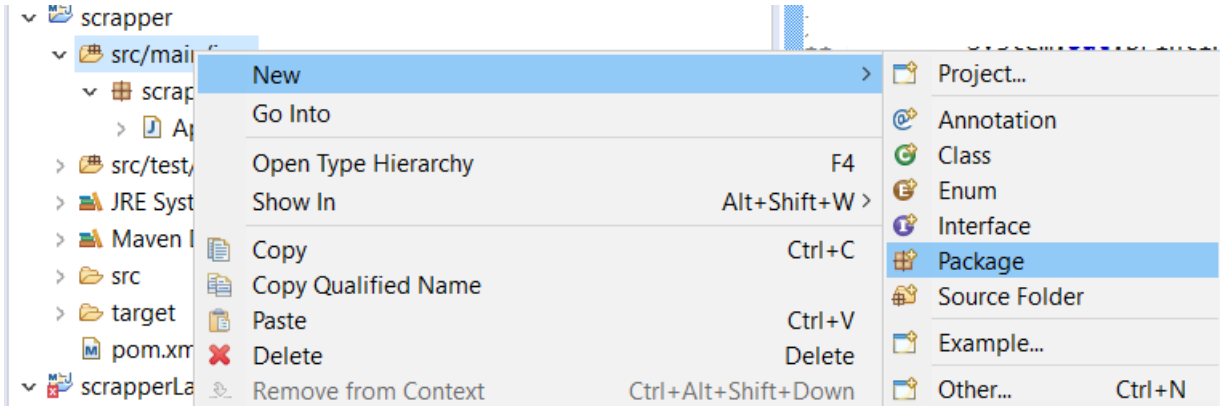
And Click Run. It will install all dependencies (jars) into your system. Check on console for output. It must show successful message. If there is any issue, please check you jdk path, admin access. You can ignore any warning in console.



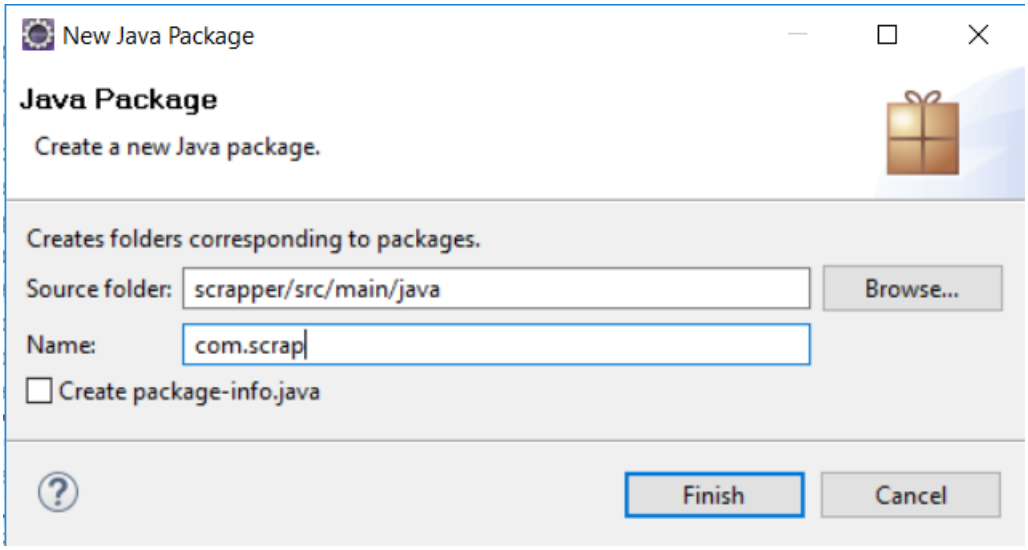
6 Create a new package

Create new Java package for your scrapping project. Right click on ‘src/main/java’ folder.

I used simple main class with main method, where I will invoke my scrapping method. You can also write Test class, and implement as per Junit framework.

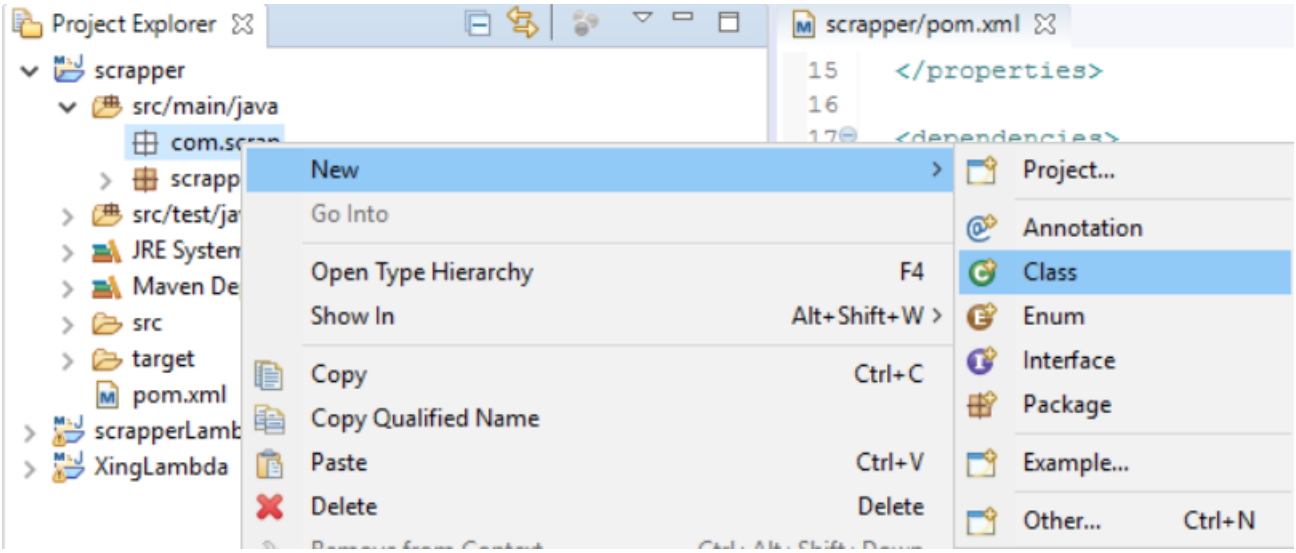


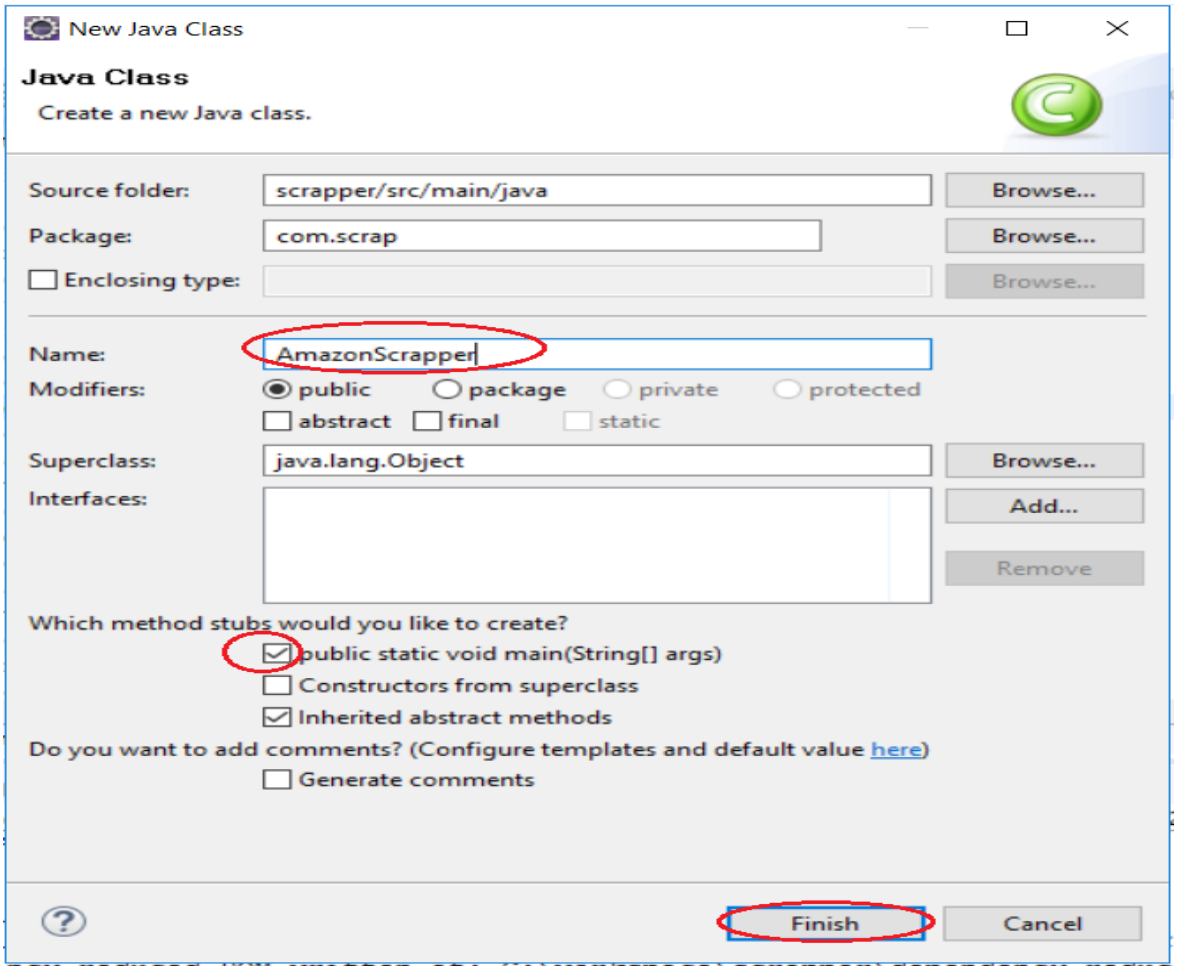
6.1 Define your package name. and click Finish.



7 Create Classes

7.1 Create main java file in your package.





7.2 Create new method to Setup your driver

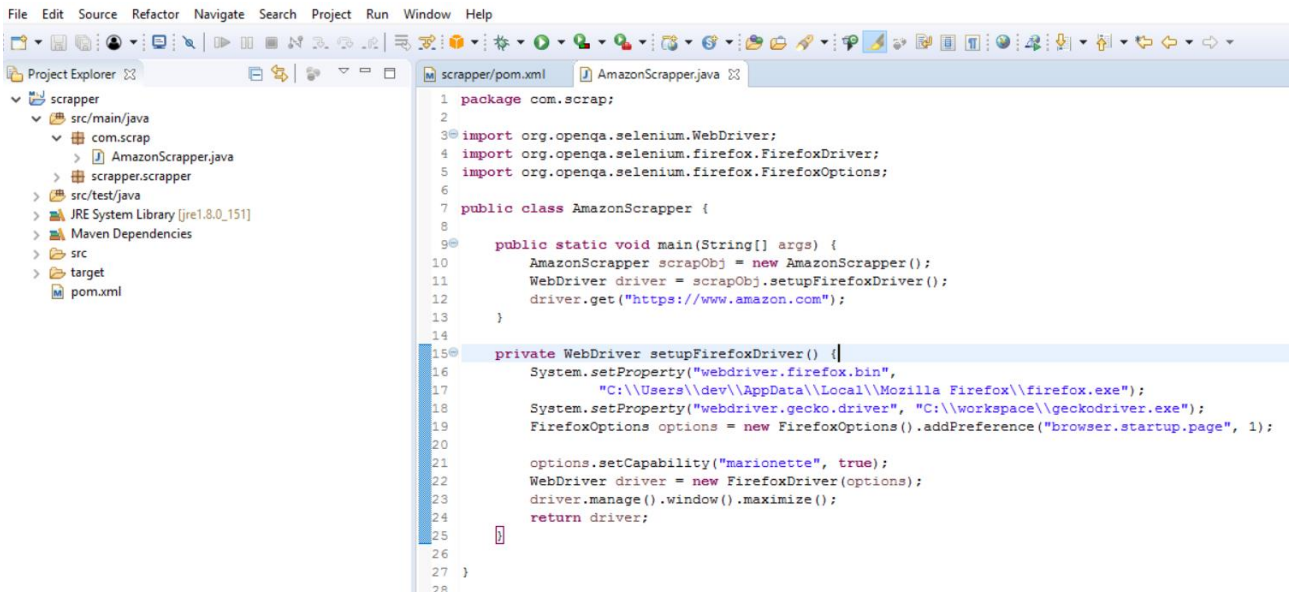
Write method to setup your WebDriver. And get the WebDriver instance. On WebDriver instance, you can invoke various method, which will fetch data from website.

To import any class in your java file, you can simply press Ctrl+Shift+O (it will import all missing class) or Ctrl+Shift+M (it will import selected missing class).

```
private WebDriver setupFirefoxDriver() {
	System.setProperty("webdriver.firefox.bin",
			"C:\\.... \\Mozilla Firefox\\firefox.exe");

	System.setProperty("webdriver.gecko.driver", "C:\\..\\geckodriver.exe");


	FirefoxOptions options = new FirefoxOptions()
				.addPreference("browser.startup.page", 1);
	options.setBinary(firefoxBinary);

	options.setCapability("marionette", true);

	WebDriver driver = new FirefoxDriver(options);
	driver.manage().window().maximize();
	return driver;
}
```

Verify path for Firefox binary file and geckodirver.exe. Always use double backward slash (\\), or single forward slash (/) when specifying path of folders.

7.3 Invoke your method by your class object as below.

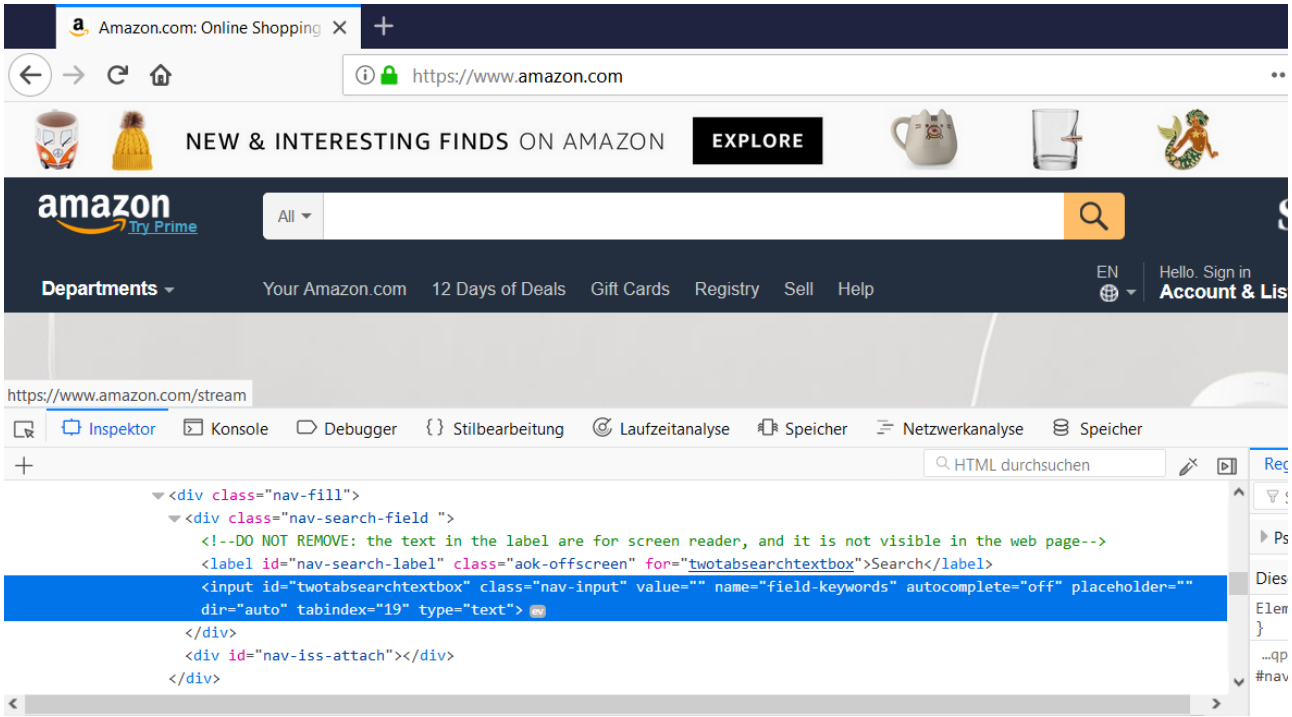


To hit any URL, you can call driver.get(String s), by passing any website url. Here I am calling amazon.com

And here it is your first interection with website through selenium. You can perform many operation, find element on websites by using WebDriver method. Please check its APIs at https://seleniumhq.github.io/selenium/docs/api/java/

7.4 Let's search and read some data from amazon.

To use amazon search box, you need to first get its xpath or html tag properties. When we will have it unique identifiers, we can tell WebDriver to find out on webpage and send our search key.



Right click on search box, select 'Inspection', you can see the highlighted code. Here, <input> tag for search box has id attribute. Which we can assume that, it will be unique name on this page. If it is not unique, we might get more than one element in our findElements() result.

```
WebElement searchElem = driver.findElement(By.id("twotabsearchtextbox"));
```

or we can use xPath for finding element. You right click on source code, and copy the xPath. xPath is always helpful, when we don't have unique identifier for any element on web-page.

For example, there is no id for search button.

Let's search iPhone 7

```
searchElem.sendKeys("iPhone 7");
```

and press search button.(for this we again need to find button id).

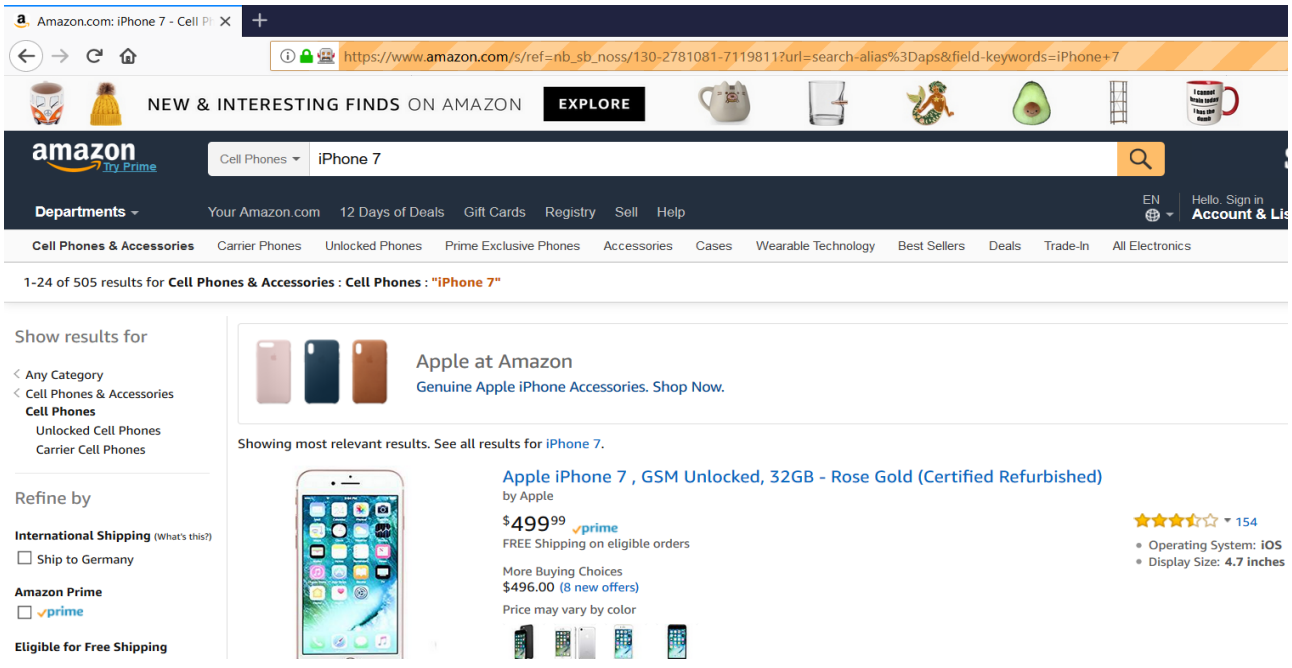
driver.findElement(By.xpath("/html/body/div[1]/header/div/div[1]/div[3]/div/form/div[2]/div/input")).click();

scrapper/pom.xml | AmazonScrapper.java

```
1  package com.scrap;
2
3  import org.openqa.selenium.By;
4  import org.openqa.selenium.WebDriver;
5  import org.openqa.selenium.WebElement;
6  import org.openqa.selenium.firefox.FirefoxDriver;
7  import org.openqa.selenium.firefox.FirefoxOptions;
8
9  public class AmazonScrapper {
10
11     public static void main(String[] args) {
12         AmazonScrapper scrapObj = new AmazonScrapper();
13         WebDriver driver = scrapObj.setupFirefoxDriver();
14         driver.get("https://www.amazon.com");
15
16         try {
17             Thread.sleep(3000);
18         } catch (InterruptedException e) {
19             //do nothing
20         }
21
22         WebElement searchElem = driver.findElement(By.id("twotabsearchtextbox"));
23         searchElem.sendKeys("iPhone 7");
24         driver.findElement(By.xpath("/html/body/div[1]/header/div/div[1]/div[3]/div/form/div[2]/div/input")).click();
25
26     }
27
28     private WebDriver setupFirefoxDriver() {
29         System.setProperty("webdriver.firefox.bin",
```

See code, I used Thread.sleep(3000). This is very important to understand that, before finding any element, page must be loaded on browser. Sometime a webpage might take longer time to open. We can ask to sleep for some second, before our code go to next line. Use wisely the time in sleep() method.

It will give you result displayed on amazon with iPhone product.



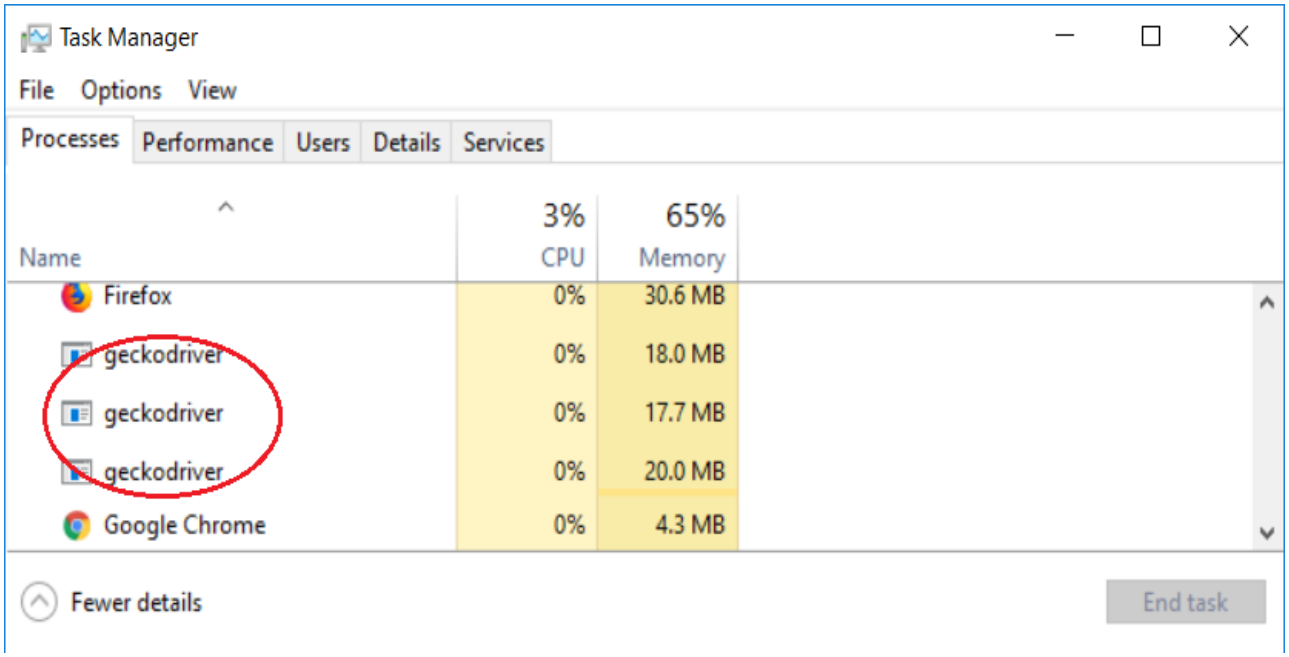
This is how you can scrap any webpage.

7.5 Close your WebDriver instance

Always close you WebDriver instance. Otherwise, **geckodriver** will be loaded into your system memory, and it might create memory issue. You can verify in your task manager, if there is any instance present, and remove it from there.

driver.close(); **//**It will close only your browser

driver.quit(); **//**it will remove geckodriver process from system. It must be called after close().



7.6 How to handle dynamic page load?

You also noticed that amazon page is dynamically loaded. When we scroll down, it's get loaded with new products. At the end of product list, NextPage button will be loaded on webpage. If you want to traverse to next page, you must need to get it loaded, before clicking it. You can use below code, to scroll down your webpage.

```
((JavascriptExecutor)driver).executeScript("window.scrollBy(0,250)", "");
```

It means your page will go down by 250 pixel. Set this value as per your need. Or you can call this function in while loop, if page is longer.

7.7 Handle Login

```
	private void login(WebDriver driver) throws Exception {
		driver.findElement(By.id("nav-link-accountList")).click();
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			//do nothing
		}
		//User name
		driver.findElement(By.id("ap_email")).sendKeys("<user id>");

		//Enter Password
		driver.findElement(By.id("ap_password")).sendKeys("<password>");

		// Click on 'Sign In' button
		driver.findElement(By.id("signInSubmit")).click();
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
	}
```

In above code: `driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);`

It allows code to wait for 10 seconds, if particular elements are not found on webpage.

Further task:

If you want to save the product results from amazon, you can simply use findElement() or findElements() method by using unique identifiers in html code. E.g. given list on amazon must be in some tag or <li> tag. If we know the unique identifiers, you can use in our code and can save into our file or in any DB.

There is another way to this. You can simply get all HTML code into your system by using **driver.getPageSource()**, and later you can use DOMParser to pick relevant data.

Please share your feedback, and let me know, if anything missing in the code.

Later, I will post, how you can save scrapped data into CSV or in RDBMS.

Happy Learning!!!

Code is available on github at

https://github.com/arun2code/ScrappingInJava