Selenium from September 17th recording

This document will be rewritten in the future.
In current form it should still guide you through setting up multiple selenium projects from the Sep 17th Selenium Refresher lecture and recording.

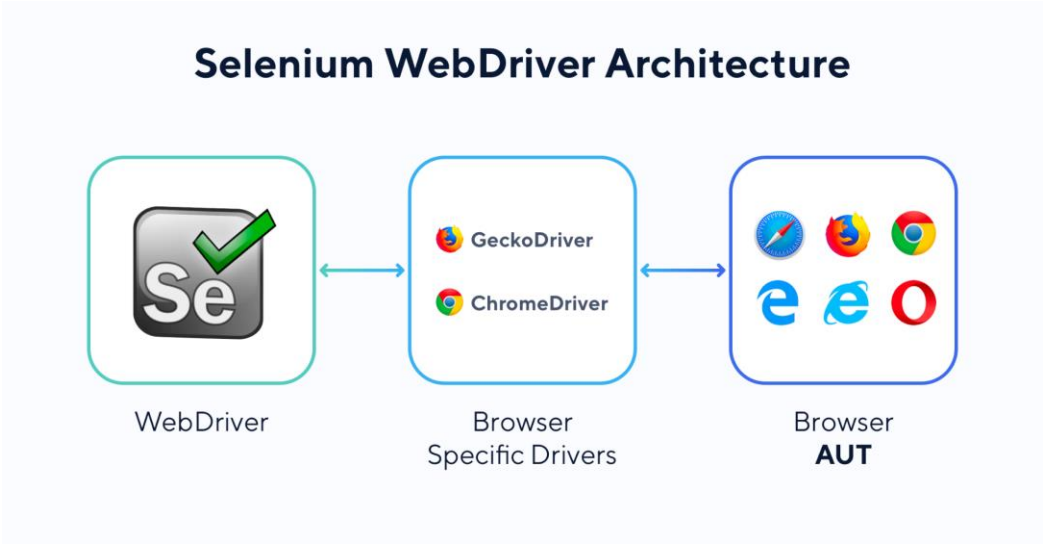
Create Maven project (simple-selenium-setup)
Update pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
         <modelVersion>4.0.0</modelVersion>
         <groupId>com.revature</groupId>
         <artifactId>simple-selenium-setup</artifactId>
         <version>0.0.1-SNAPSHOT</version>

         <properties>
                  <maven.compiler.source>1.8</maven.compiler.source>
                  <maven.compiler.target>1.8</maven.compiler.target>
         </properties>

         <dependencies>
                  <dependency>
                           <groupId>org.seleniumhq.selenium</groupId>
                           <artifactId>selenium-java</artifactId>
                           <version>3.141.59</version>
                  </dependency>
         </dependencies>
</project>
```

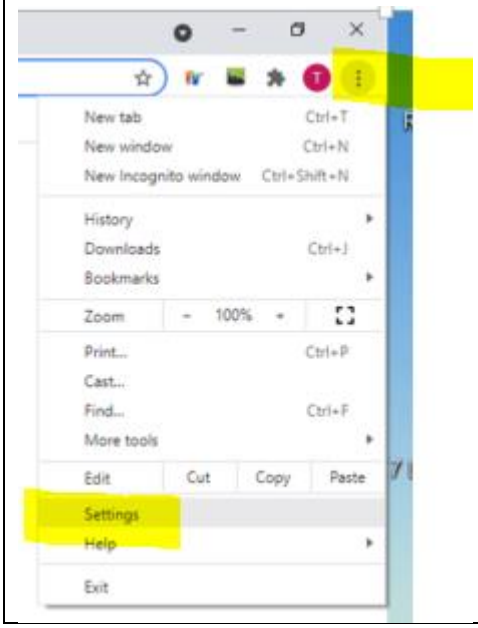
Note: The selenium dependency (with the webdriver) ONLY automates the browser.
Need a test frame work plus selenium to do actual testing. (cucumber, testNG, Jupiter, JUnit)

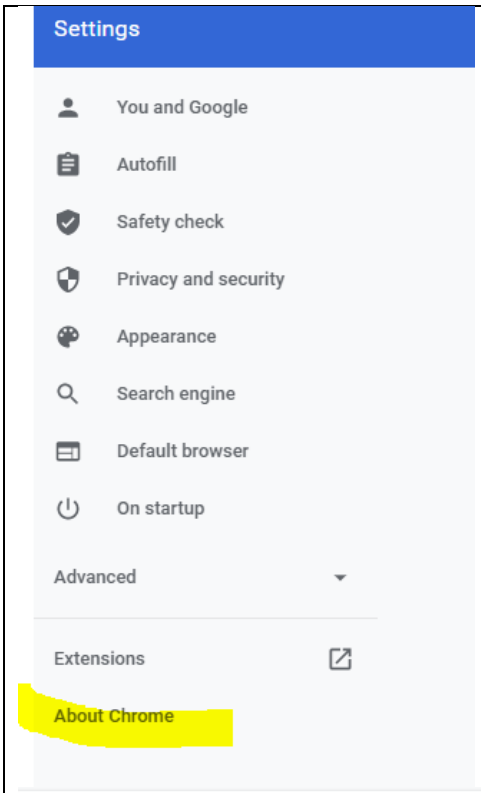
Cucumber – BDD behavior driver development.

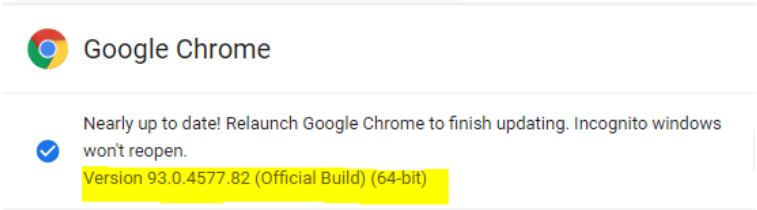
Google search selenium webdriver



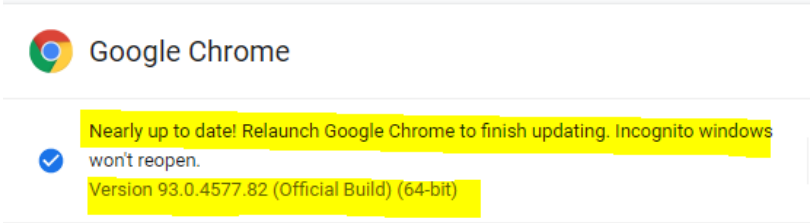
Download Webdriver
Check Chrome version first



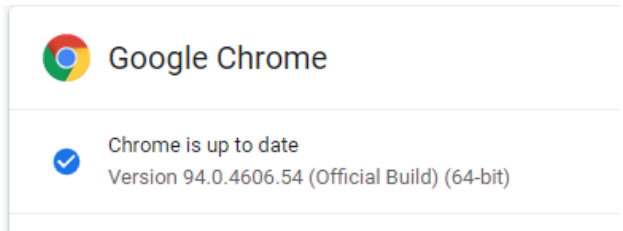




Need the webdriver for your version of Chrome



You may need to relaunch your browser to get the latest version



Go to:
https://chromedriver.chromium.org/

All versions available in Downloads

- Latest dev release: ChromeDriver 95.0.4638.10
- Latest beta release: ChromeDriver 94.0.4606.41
- Latest stable release: ChromeDriver 93.0.4577.63

Current Releases

- If you are using Chrome version 95, please download ChromeDriver 95.0.4638.10
- If you are using Chrome version 94, please download ChromeDriver 94.0.4606.41
- If you are using Chrome version 93, please download ChromeDriver 93.0.4577.63
- If you are using Chrome version 92, please download ChromeDriver 92.0.4515.107

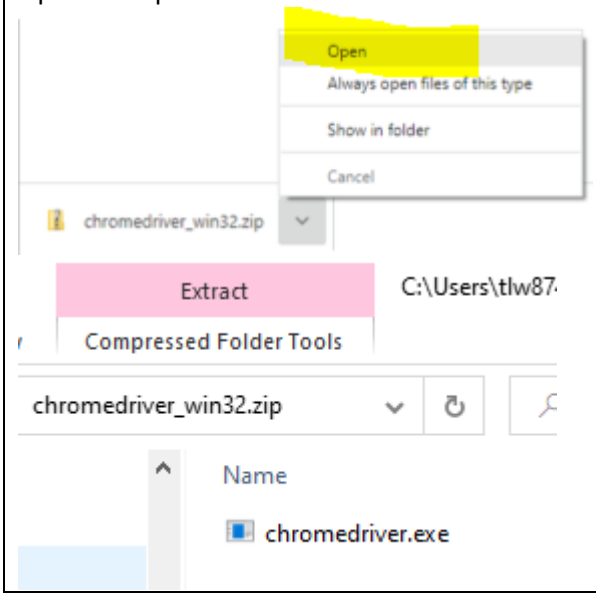
Find the version that starts with your Chrome version in my case 94.0…
Download the version

Index of /94.0.4606.41/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2021-09-09 09:57:33	9.42MB	9a00e20a5e5646c009c0ad74a446df10
chromedriver_mac64.zip	2021-09-09 09:57:36	7.80MB	f89f291ccc799a9de32e832807ce7e42
chromedriver_mac64_m1.zip	2021-09-09 09:57:38	7.15MB	69b92932297c2ce2764aa4c7744e1c17
chromedriver_win32.zip	2021-09-09 09:57:41	5.72MB	c853398ecdb35290e60e57a833e61752
notes.txt	2021-09-09 09:57:45	0.00MB	b88b7fe40b1af6891a9f57da13e08914

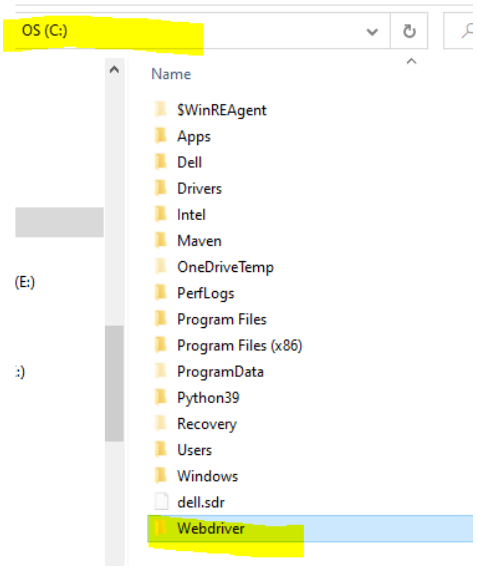
For Windows down load the win32.zip file.

Open the zip file



You want to place the chromedriver.exe in a folder on your computer. You can create a folder on your Desktop or some other location.

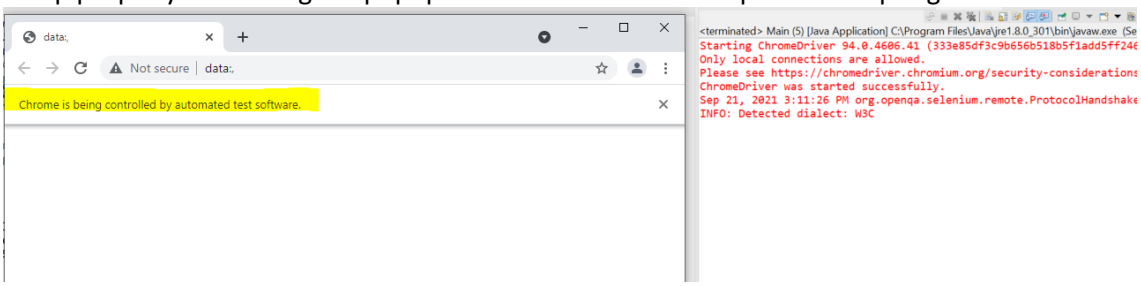
For convenience you should create the webdriver folder in the C:\ driver root folder.



Using simple-selenium-setup project creates a project using selenium with NO testing frame work. It just automates the browser.

```
package com.revature.app;

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.chrome.ChromeOptions;

public class Main {

	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");

		// To make use of this option, we would actually need to pass it into the ChromeDriver when we
instantiate it
		WebDriver driver = new ChromeDriver(options);

}
```

Run the above snippet of code with your directory location to ensure selenium and the driver are setup properly. You will get a popup chrome window and output to the Spring Tool console.




```
Starting ChromeDriver 94.0.4606.41 (333e85df3c9b656b518b5f1add5ff246365b6c24-refs/branch-heads/4606@{#845}) on port 32592
Only local connections are allowed.
```

```
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Sep 21, 2021 3:11:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
```

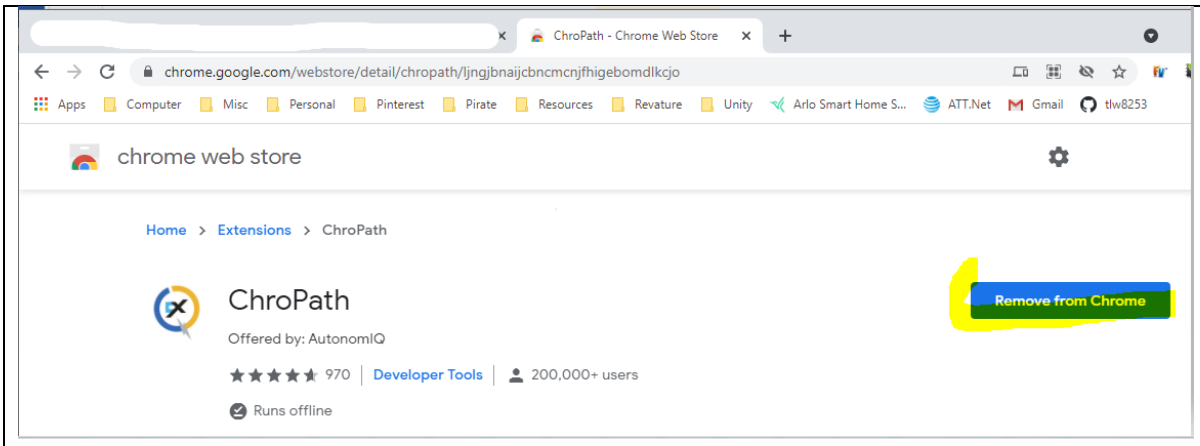
Make sure to add driver.quit(); to the end of the processing to take care of hanging chrome browser tasks. Else you would need to start the task manager and end them manually.

```
package com.revature.app;

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.chrome.ChromeOptions;

public class Main {

	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver", "C:/Webdrivers/chromedriver.exe");

		// To make use of this option, we would actually need to pass it into the ChromeDriver when we
instantiate it
		WebDriver driver = new ChromeDriver();

		driver.quit();
	}
}
```

Log into Facebook

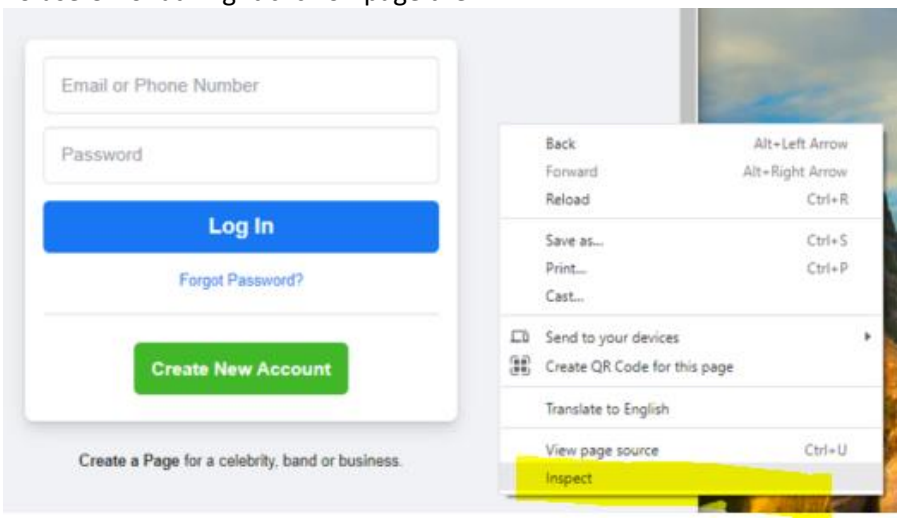
```
package com.revature.app;

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.chrome.ChromeOptions;

public class Main {

  public static void main(String[] args) {
	System.setProperty("webdriver.chrome.driver", "C:/Webdrivers/chromedriver.exe");

	// To make use of this option, we would actually need to pass it into the ChromeDriver when we instantiate it
	WebDriver driver = new ChromeDriver();
	driver.get("http://facebook.com");


	try {
		Thread.sleep(5000);
	} catch (InterruptedException e) {
		e.printStackTrace();
	}


	driver.quit();
  }
}
```

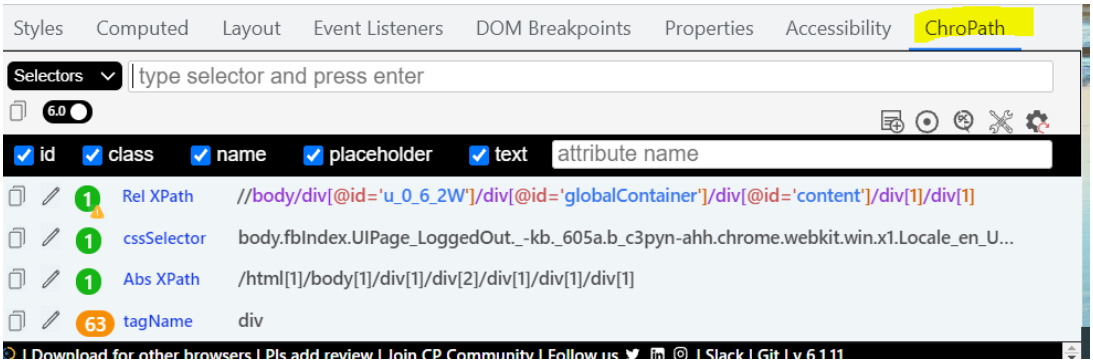
chropath – used to find page elements
Google search for chropath
https://chrome.google.com/webstore/detail/chropath/ljngjbnaijcbncmcnjfhigebomdlkcjo



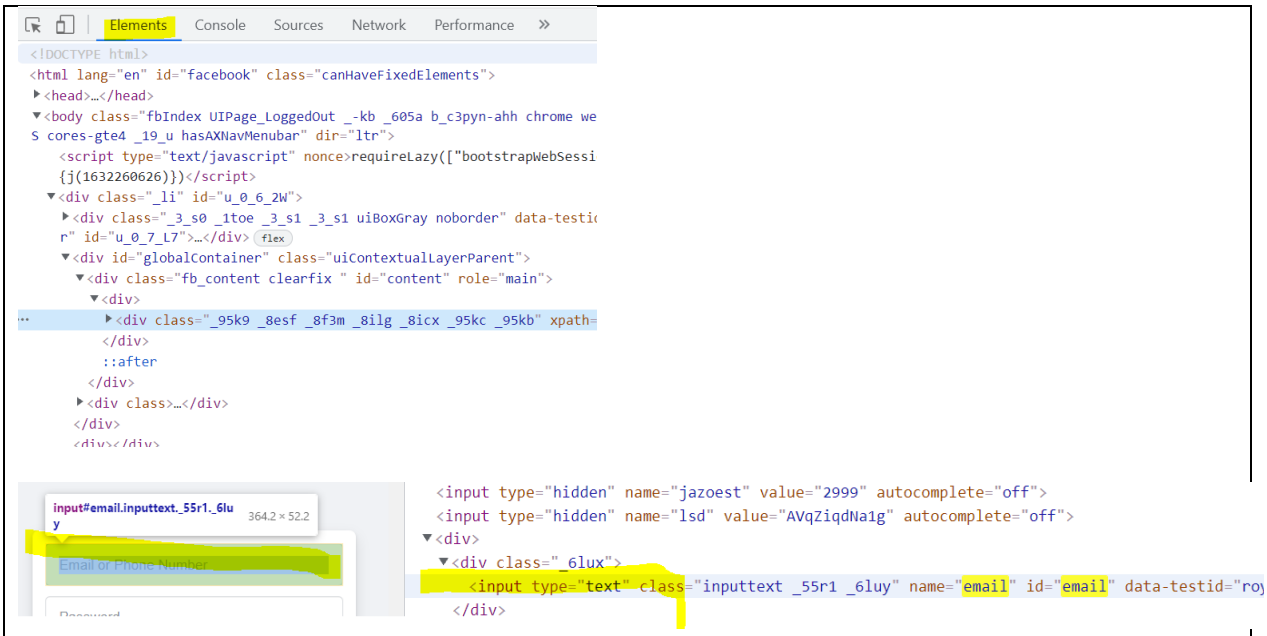
Verify that ChroPath is already installed (from a previous lecture) if not install it.

To use ChroPath right click on page then





Development console element tab



Highlighting dom elements in the element tab highlights area on the webpage

Can select elements by name, id, placeholder, etc.
Locators are how to select elements (easy, xpath selectors, css selectors)

```
WebElement emailInput = driver.findElement(By.name("email"));
```

USING: ChromPath
Element / xpath search
Use single quote
//input[@class='inputtext _55r1 _6luy']
Select an input element "@" called class with value between single quotes

```
public class Main {

  public static void main(String[] args) {
    System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--incognito", "--headless");
    WebDriver driver = new ChromeDriver(options);
    driver.get("http://facebook.com");

//NOTE: comment out all but one variable name emailInput to test
    WebElement emailInput = driver.findElement(By.name("email"));
    WebElement emailInput = driver.findElement(By.id("email"));

  // Examples of the "hard" locators (Xpath and CSS selectors)
  WebElement emailInput = driver.findElement(By.xpath("//*[@name='email']"));
  // Generally there is a CSS equivalent of Xpath slectors
  WebElement emailInput = driver.findElement(By.cssSelector("*[name='email']"));
```

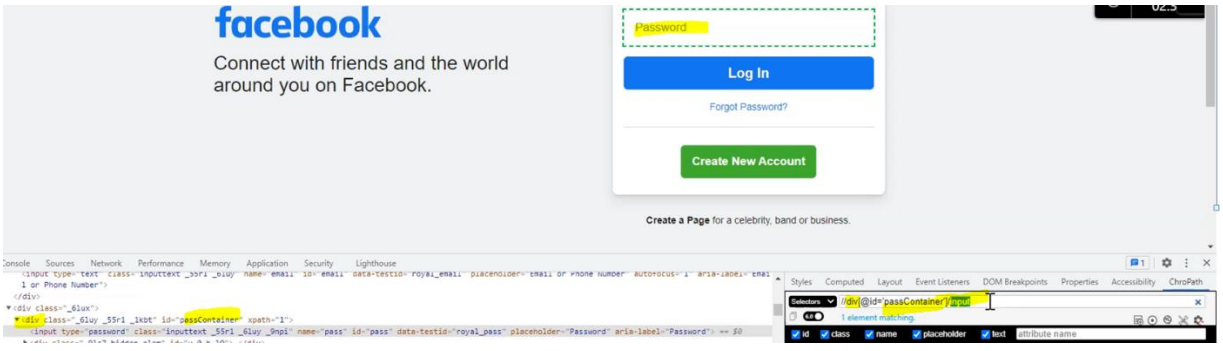
```
//Note: should avoid using className since they are not guaranteed to be consistent across
//page loads and possibly used in many places.  Stick to By.id
    WebElement emailInput = driver.findElement(By.className("inputtext _55r1 _6luy"));


  WebElement h2Header = driver.findElement(By.xpath("//*[contains(text(), 'Connect
with')]")); // partial match for the text of an element
  System.out.println(h2Header.getText());



    // Type into emailInput
    emailInput.sendKeys("some.longin@some.com");
    // Type into passwordInput
    passwordInput.sendKeys("dfdljklkjlkjfds");
    loginButton.click();


  }
}
```

Xpath cheat sheet
https://devhints.io/xpath

XPath v. CSS
XPath is slower than CSS (but still fast)
XPath is apparently about 10 times slower, but they're both so fast, it probably really doesn't matter
CSS selectors are faster, but less flexible
XPath selectors allow you to traverse from a child element to a parent element, vice versa, etc.
XPath allows you to select an element based on the text of that element

Using xpath to select child elements.
Use directory structure notation like here we are using root of:
Div with id of passContainer then / input

```
  // This XPath is saying, select an element with a tag name of input that is a child
  // of a div with an id of passContainer
  WebElement passwordInput = driver.findElement(By.xpath("//div[@id='passContainer']/input"));
```



You can be more specific going down the family tree and want the child name of ‘pass’

//div[@id='passContainer']/input[@name='pass']

Key syntax to complete login to webpage

```
WebElement emailInput = driver.findElement(By.cssSelector("*[name='email']"));
WebElement passwordInput = driver.findElement(By.id("passContainer")).findElement(By.tagName("input"));
WebElement loginButton = driver.findElement(By.name("login"));

emailInput.sendKeys("bach_tran@outlook.com");
passwordInput.sendKeys("dfdljklkjlkjfds");
loginButton.click();
```

Key elements from simple-selenium-setup project

```
package com.revature.app;

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.chrome.ChromeOptions;

public class Main {

	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");
		ChromeOptions options = new ChromeOptions();
		options.addArguments("--incognito", "--headless");
		WebDriver driver = new ChromeDriver(options);
		driver.get("http://facebook.com");

		WebElement emailInput = driver.findElement(By.cssSelector("*[name='email']"));
		WebElement passwordInput = driver.findElement(By.id("passContainer")).findElement(By.tagName("input"));
		WebElement loginButton = driver.findElement(By.name("login"));

		WebElement h2Header = driver.findElement(By.xpath("//*[contains(text(), 'Connect with')]"));
		System.out.println(h2Header.getText());

		emailInput.sendKeys("bach_tran@outlook.com");
		passwordInput.sendKeys("dfdljklkjlkjfds");
		loginButton.click();

		WebElement incorrectPasswordElement = driver.findElement(By.xpath("//div[contains(text(), 'The password
you')]"));
		System.out.println(incorrectPasswordElement.getText());


		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			e.printStackTrace();
		}

		driver.quit();
	}

}
```

Complete class from simple-selenium-setup project

```
package com.revature.app;

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

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

public class Main {

	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");

		// Sometimes you might need to do additional configuration with the settings when running your
ChromeDriver
		// There is an object that you can create called a ChromeOptions object that will contain various
settings
		ChromeOptions options = new ChromeOptions();
		// To enable incognito, I could add an argument to this ChromeOptions object
		options.addArguments("--incognito", "--headless");
		// --incognito is an argument to start up the browser in incognito mode
		// --headless is an argument to not have the browser actually pop up on the screen (important when
running Selenium tests
		// from headless systems, such as an EC2 instance)
		// What we mean by headless is a computer that doesn't have any monitor/display/graphics card

		// To make use of this option, we would actually need to pass it into the ChromeDriver when we
instantiate it
		WebDriver driver = new ChromeDriver(options);

		// WebDriver is an interface for any type of WebDriver
		// ChromeDriver is a class that implements WebDriver
//		WebDriver driver = new ChromeDriver();
		driver.get("http://facebook.com");

		// Examples of the "easy" locators (refer back to the powerpoint for the others)
//		WebElement emailInput = driver.findElement(By.name("email"));
//		WebElement emailInput = driver.findElement(By.id("email"));

		// Examples of the "hard" locators (Xpath and CSS selectors)
//		WebElement emailInput = driver.findElement(By.xpath("//*[@name='email']"));
		WebElement emailInput = driver.findElement(By.cssSelector("*[name='email']"));

		// This XPath is saying, select an element with a tag name of input that is a child of a div with an id
		// of passContainer
//		WebElement passwordInput = driver.findElement(By.xpath("//div[@id='passContainer']/input"));

		// I can chain findElements together to find the child elements that exist within a parent element
		// This example here is equivalent to the xpath example above: "//div[@id='passContainer']/input"
		WebElement passwordInput = driver.findElement(By.id("passContainer")).findElement(By.tagName("input"));

		WebElement loginButton = driver.findElement(By.name("login"));

		// XPath v. CSS
		// XPath is slower than CSS (but still fast)
		// XPath is apparently about 10 times slower, but they're both so fast, it probably really doesn't
matter
		// CSS selectors are faster, but less flexible
		// XPath selectors allow you to traverse from a child element to a parent element, vice versa, etc.
		// XPath allows you to select an element based on the text of that element
//		WebElement h2Header = driver.findElement(By.xpath("//h2[text()='Connect with friends and the world
around you on Facebook.']"));
		WebElement h2Header = driver.findElement(By.xpath("//*[contains(text(), 'Connect with')]")); // partial
match for the text of an element
		System.out.println(h2Header.getText());

		// Type into emailInput
		emailInput.sendKeys("bach_tran@outlook.com");
		// Type into passwordInput
		passwordInput.sendKeys("dfdljklkjlkjfds");
		loginButton.click();

		WebElement incorrectPasswordElement = driver.findElement(By.xpath("//div[contains(text(), 'The password
you')]"));
		System.out.println(incorrectPasswordElement.getText());


		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			e.printStackTrace();
		}

		driver.quit();
	}

}
```

Project: selenium-waits

implicitlyWait()

```
WebDriver driver = new ChromeDriver();

// For any element that is not immediately available when we call findElement
// The implicit wait will kick in
// This implicit wait is configured to wait for up to 5 seconds
// Anything beyond that will result in our usual NoSuchElementException
// If Selenium finds the element before 10 seconds, it will not wait
// the full 10 seconds
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

// Our implicit wait will apply to any code that comes after this configuration
// for this particular WebDriver object

// Implicit waits are called implicit, because we are not explicitly specifying
// what element we are waiting on to appear on the page
// Implicit waits are easy to set up, but are essentially a "black box" that you cannot look inside of
// That is the main drawback of implicit waits
```

Explicit wait

```
WebElement buttonElement = driver.findElement(By.id("btn"));
buttonElement.click();

// Explicit wait

// WebDriverWait object gives us functionality for performing an explicit wait
WebDriverWait webDriverWait = new WebDriverWait(driver, 10);

// Wait until the p tag element appears (for a maximum of 10 seconds)
// If not found, gives a TimeoutException
//        webDriverWait.until(ExpectedConditions.visibilityOfElementLocated(By.tagName("p")));
//        WebElement pElement = driver.findElement(By.tagName("p"));

// This is the same as the above, but more condensed, because the until method returns
// The WebElement we are waiting on once it is available
WebElement pElement = webDriverWait.until(ExpectedConditions.visibilityOfElementLocated(By.tagName("p")));
// Explicit wait checks every 500 milliseconds to see if the element is available
```

New project: simple-page-object-model-example (**Page** Object Model)

Simplified main

```
	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");

		ChromeOptions options = new ChromeOptions();
		options.addArguments("--incognito");
		WebDriver driver = new ChromeDriver(options);

		driver.get("http://facebook.com");

		FacebookLoginPage loginPage = new FacebookLoginPage(driver);

		System.out.println(loginPage.h2Header().getText());
		loginPage.username().sendKeys("bach_tran@outlook.com");
		loginPage.password().sendKeys("asdfsdfsdfdsf");
		loginPage.loginButton().click();
		System.out.println(loginPage.incorrectPassword().getText());

		driver.quit();
	}
```

New package
com.revature.page
Class: FacebookLoginPage

```
package com.revature.page;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

// This is an example of the PAGE OBJECT MODEL design pattern (POM)
// It allow us to create an object representation of a page and keep our Selenium automation framework organized and neat
public class FacebookLoginPage {

	private WebDriver driver;

	private WebElement usernameInput;
	private WebElement passwordInput;
	private WebElement loginButton;
	private WebElement h2Header;
	private WebElement incorrectPasswordElement;

	public FacebookLoginPage(WebDriver driver) {
		this.driver = driver;
	}

	public WebElement username() {
		if (this.usernameInput == null) {
			this.usernameInput = this.driver.findElement(By.cssSelector("*[name='email']"));
		}

		return this.usernameInput;
	}

	public WebElement password() {
		if (this.passwordInput == null) {
			this.passwordInput =
this.driver.findElement(By.id("passContainer")).findElement(By.tagName("input"));
		}

		return this.passwordInput;
	}

	public WebElement loginButton() {
		if (this.loginButton == null) {
			this.loginButton = this.driver.findElement(By.name("login"));
		}

		return this.loginButton;
	}

	public WebElement h2Header() {
		if (this.h2Header == null) {
			this.h2Header = this.driver.findElement(By.xpath("//*[contains(text(), 'Connect with')]"));
		}

		return this.h2Header;
	}
```

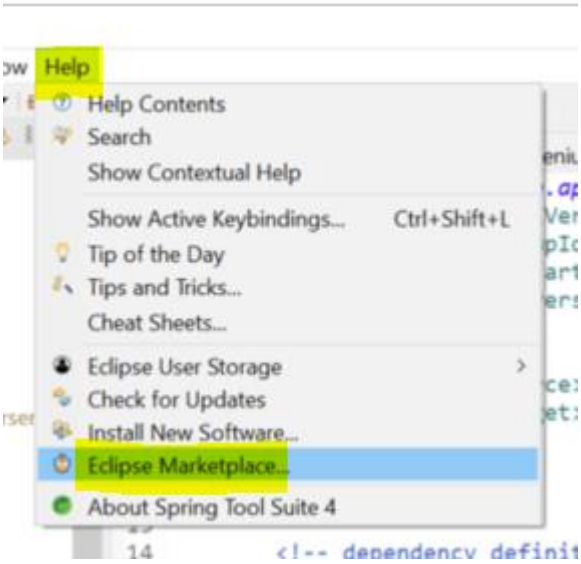
```
          public WebElement incorrectPassword() {
                    if (this.incorrectPasswordElement == null) {
                              this.incorrectPasswordElement = this.driver.findElement(By.xpath("//div[contains(text(), 'The
password you')]"));
                    }

                    return this.incorrectPasswordElement;
          }

}
```

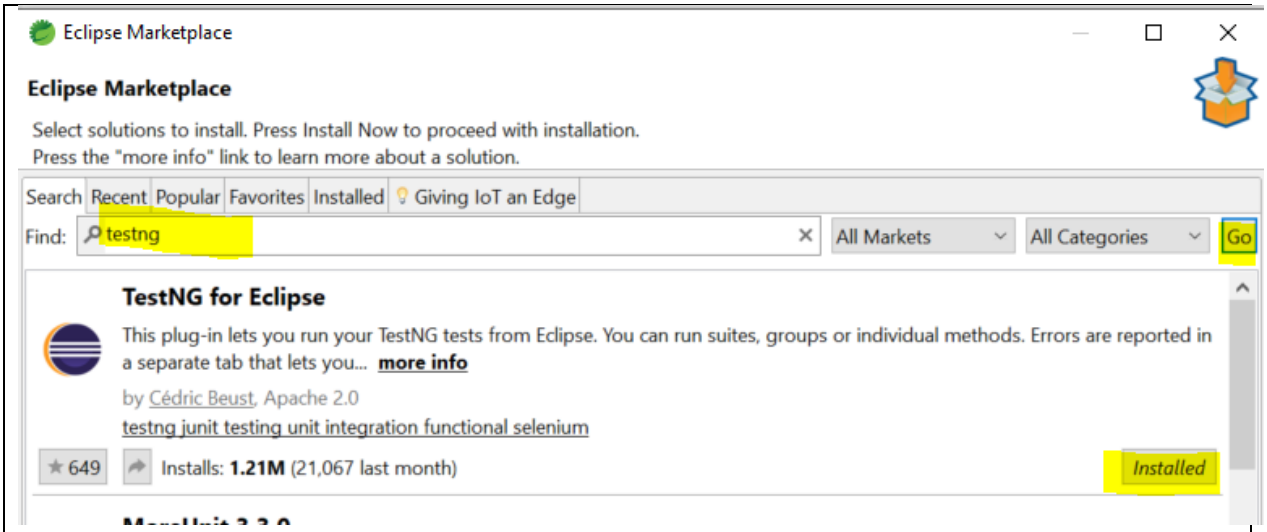
testng-setup
Most feature rich and allows sequencing of test cases

pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
         <modelVersion>4.0.0</modelVersion>
         <groupId>com.revature</groupId>
         <artifactId>testng-setup</artifactId>
         <version>0.0.1-SNAPSHOT</version>

         <properties>
                  <maven.compiler.source>1.8</maven.compiler.source>
                  <maven.compiler.target>1.8</maven.compiler.target>
         </properties>

         <dependencies>
                  <dependency>
                           <groupId>org.seleniumhq.selenium</groupId>
                           <artifactId>selenium-java</artifactId>
                           <version>3.141.59</version>
                  </dependency>

                  <dependency>
                           <groupId>org.testng</groupId>
                           <artifactId>testng</artifactId>
                           <version>7.4.0</version>
                           <scope>test</scope>
                  </dependency>
         </dependencies>
</project>
```

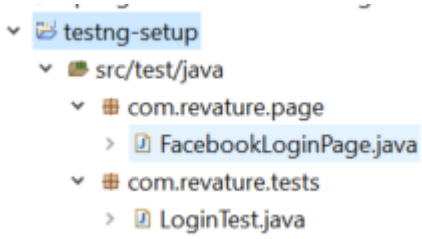
If not already installed, install testNG from the marketplace.



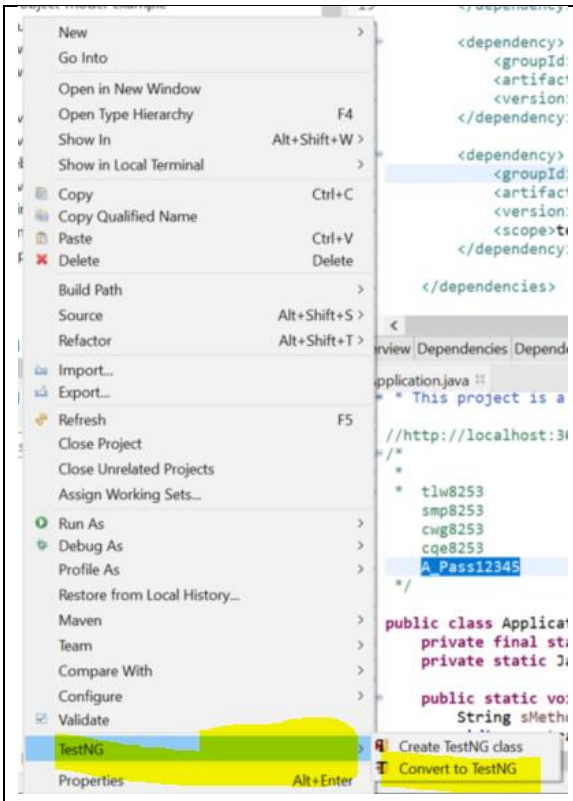


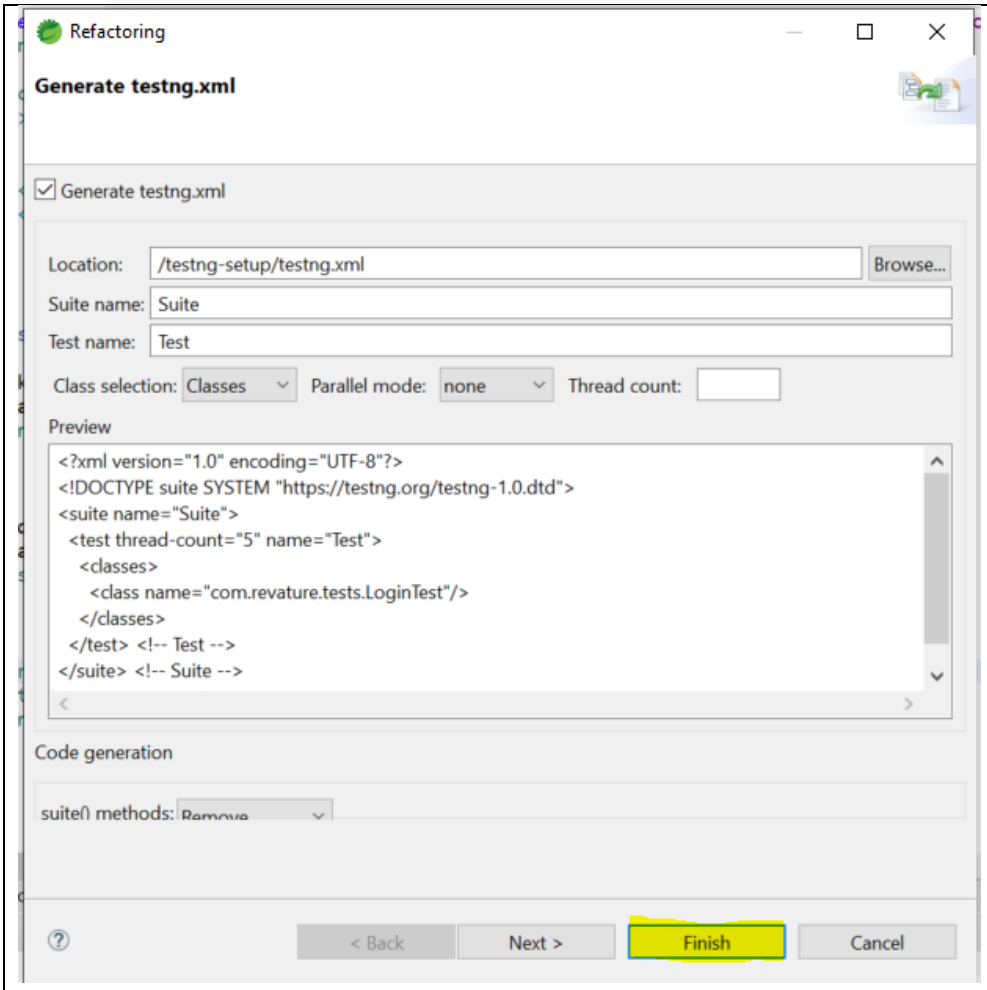
You should be prompted to restart the IDE after install.

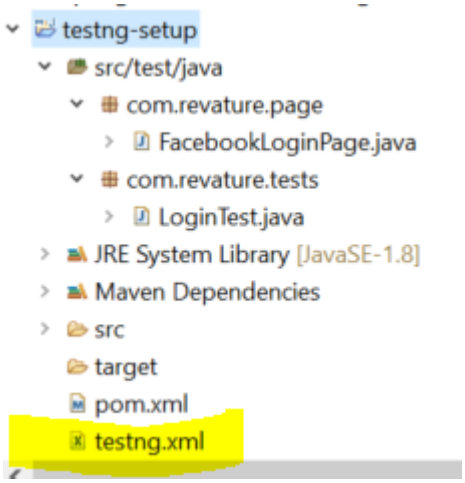
At this point files will be placed under the src/test/java structure.



Right click on project


New
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
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Run As
Debug As
Profile As
Restore from Local History...
Maven
Team
Compare With
Configure
Validate
TestNG
Properties Alt+Enter
Create TestNG class
Convert to TestNG
<dependency>
<groupId
<artifac
<version
</dependency>
<scope>te
</dependencies>
Dependencies
application.java
* This project is a
//http://localhost:3
tlw8253
smp8253
cwg8253
cqe8253
A_Pass12345
public class Applica
private final st
private static J
public static vo
String sMeth





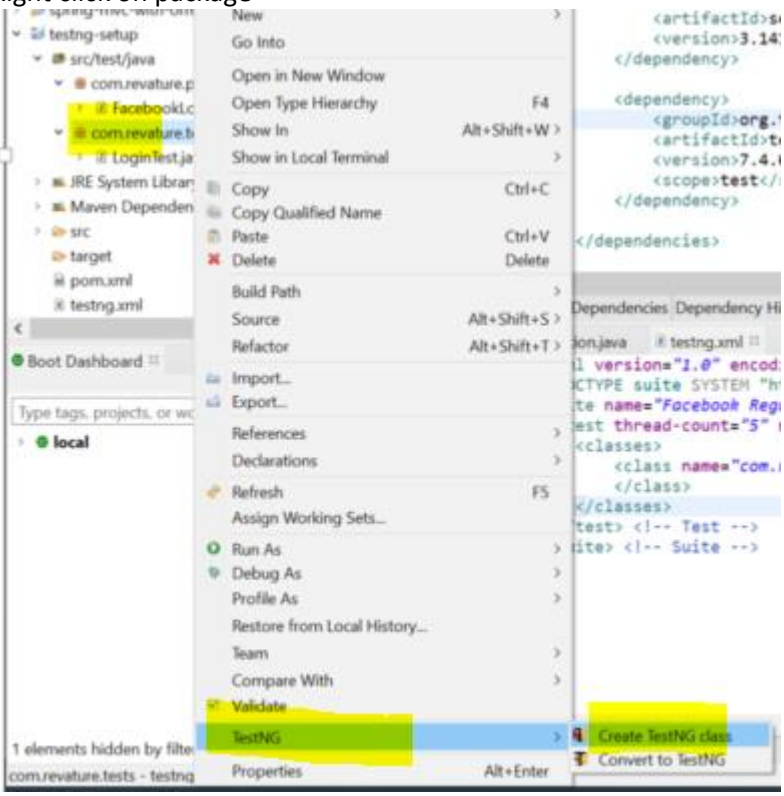
testng.xml

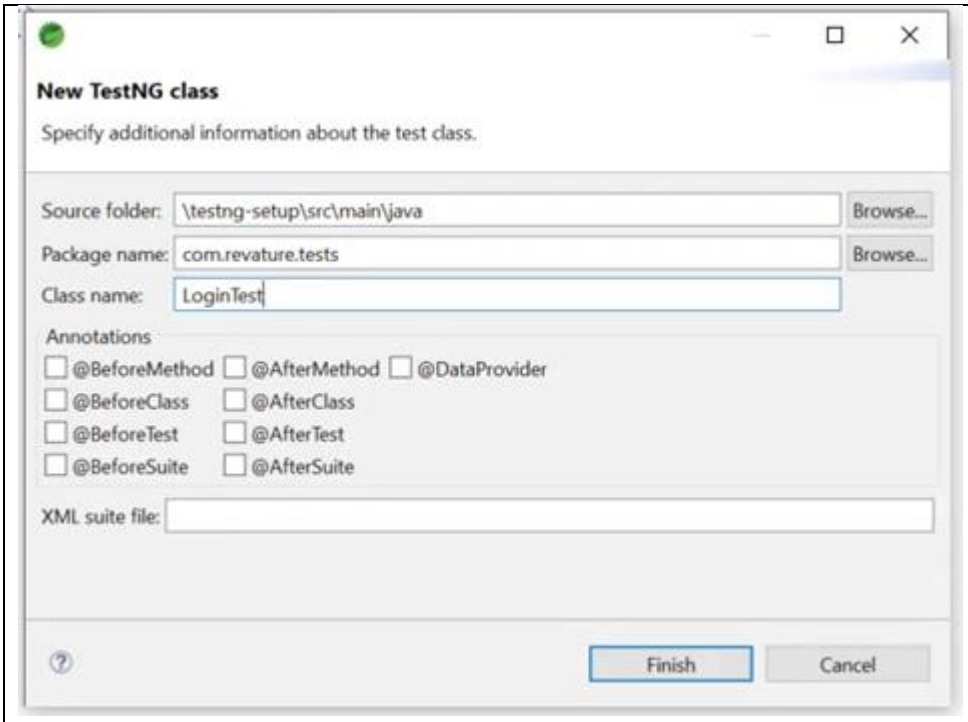
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Facebook Regular User Suite">
  <test thread-count="5" name="Login related tests">
          <classes>
                    <class name="com.revature.tests.LoginTest">
```

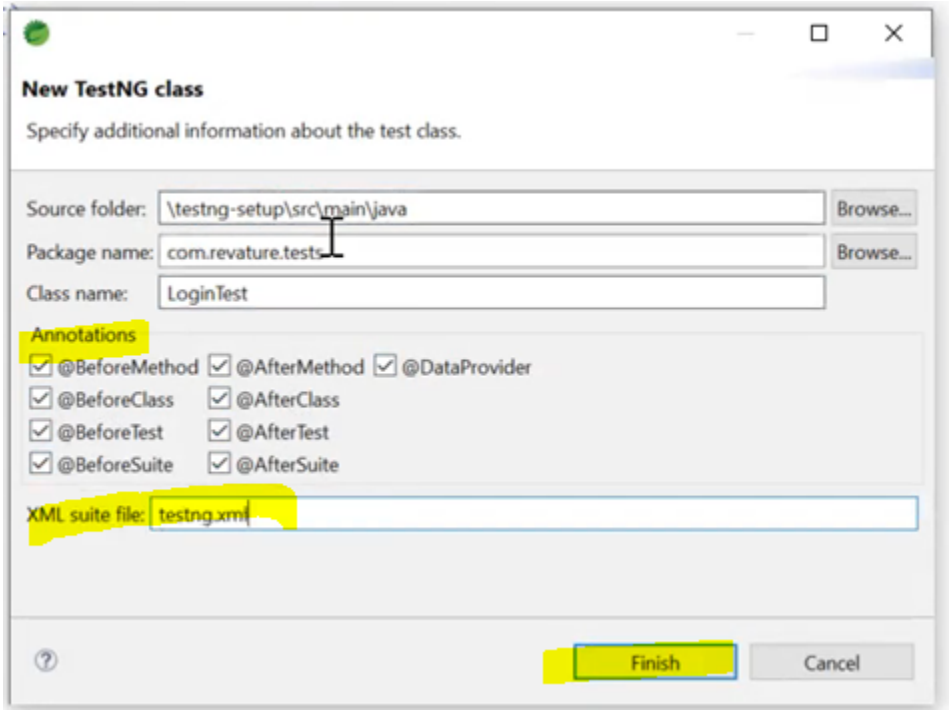
```
                </class>
            </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

<suite> highest level tests hierarchy
<test> next level
<classes> level under <test>
<methods> level under <classes>

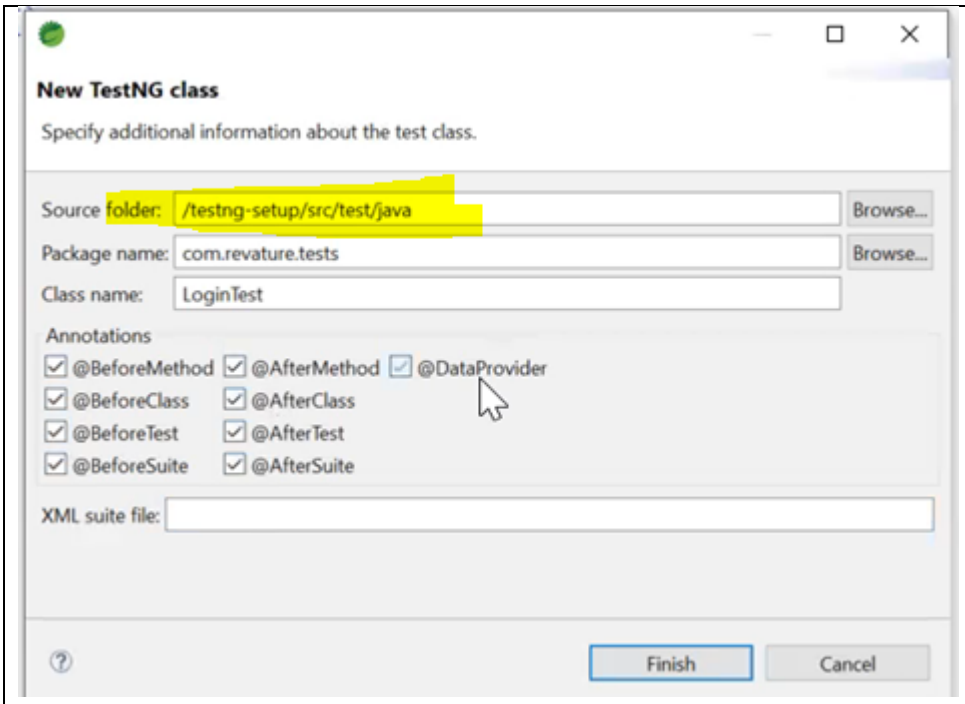
Right click on package







The above did not place the file in the right location so Bach deleted packages and files then add the class file with:


New TestNG class
Specify additional information about the test class.
Source folder: /testng-setup/src/test/java
Browse...
Package name: com.revature.tests
Browse...
Class name: LoginTest
Annotations
@BeforeMethod
@AfterMethod
@DataProvider
@BeforeClass
@AfterClass
@BeforeTest
@AfterTest
@BeforeSuite
@AfterSuite
XML suite file:
Finish
Cancel