## Install Selenium with Maven

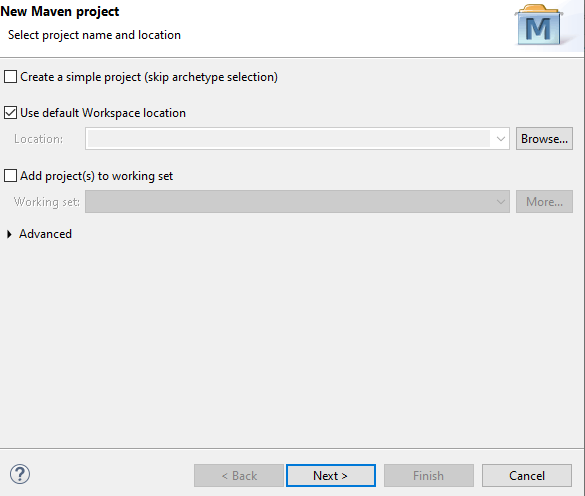


### 1.Install java: download latest java version from java website

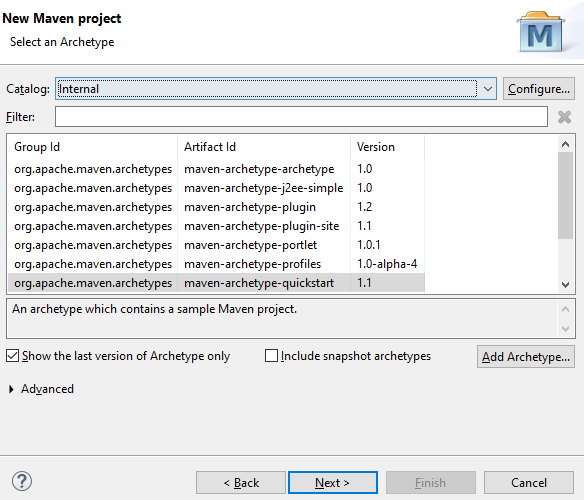
### 2.Download Eclipse and install Eclipse Java Developer version

 **//Eclipse symbol will be like this**

### 3.Open Eclipse and go to File -> New -> Maven project and then follow below picture steps

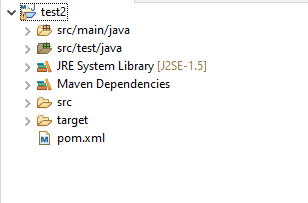


**Choose catalog internal and keep others default**



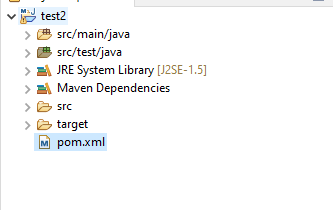
Give grouid , artifat id other information as user defined like test2

Then finally below GUI will appear with new maven project



### 4.Acquire main dependency for selenium

**Double click below pom.xml file and paste following command at dependency section ny keeping “junit” section just paste below dependency below junit section in dependency** . Keep other section untouched , only delete existing entry from dependency and paste the following command



* If you are using **ChromeDriver** use either of the **<artifactId>selenium-java</artifactId>** or **<artifactId>selenium-server</artifactId>**. Avoid using both at the same time.

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-chrome-driver</artifactId>

<version>3.141.59</version>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-server</artifactId>

<version>3.141.59</version>

</dependency>

<dependency>

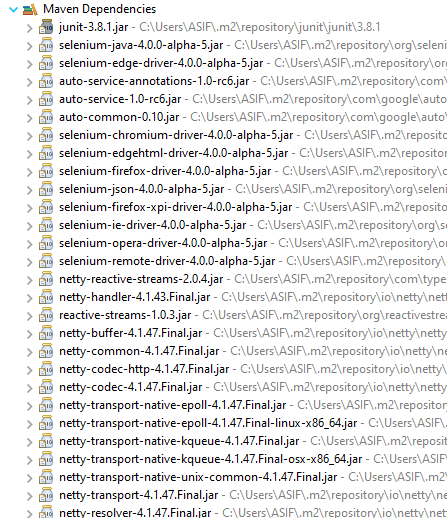
<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-api</artifactId>

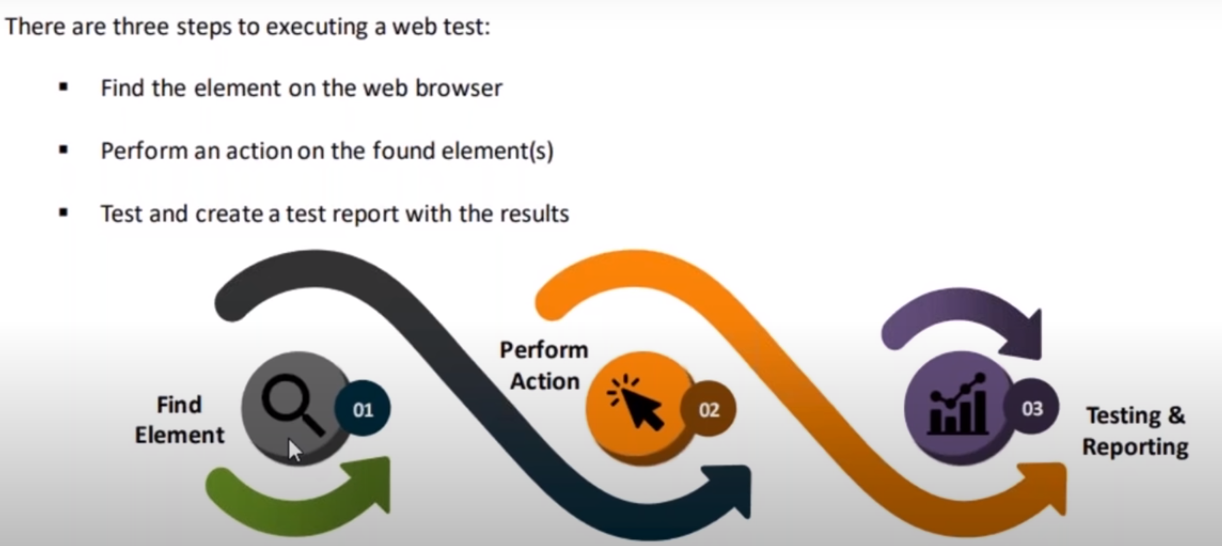
<version>3.141.59</version>

</dependency>

After that Eclipse will do some build process and download dependencies and then GUI dependency section will look below



## Automation test steps



### 1.Find Element from WebSite

We can find element from website by following seledctors:

**.ID**

**.Name**

**.ClassName**

**.Link Name**

**.Link text**

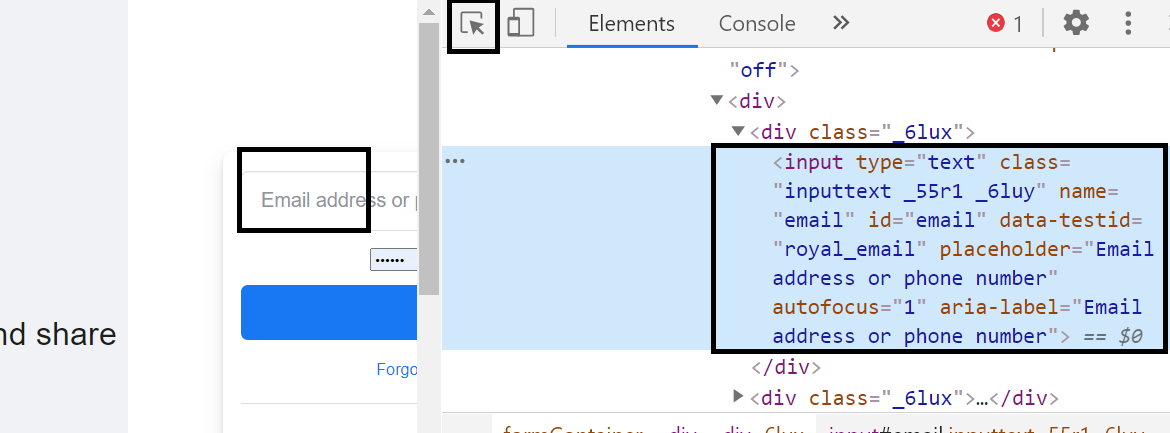
**.Partial link text**

**.XPATH**

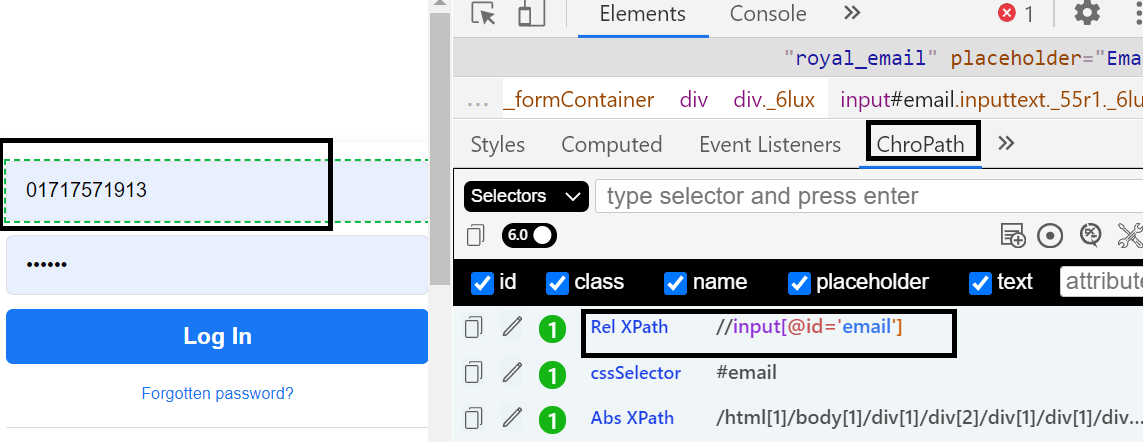
**Syntax:**

**WebElement button = driver.findElement(By.class(“CWak9”))**

**Example :Go to facebook.com website and right click go to inspect and press Ctl+shift+c or click “Select at element on the page” icon on top most left of the inspect page. Then u can choose any element in the website on left and u can see corresponding on the right.**

****

**But className , Name can be duplicate and so it Is better to use Xpath selector to get the unique selector from a website. Xpath is always unique. To get Xpath u need to download the xpath extension on your browser. Like In google chrome u need to install extension “Chorpath” . go to chrome webstore and add ChorePath**

****

**Here, Xpath for userid text box is** //input[@id='email']

**WebElement button =** driver.findElements(By.xpath("//input[@id='email']"));

### 2.Perform Action

**Mouse Actions:**

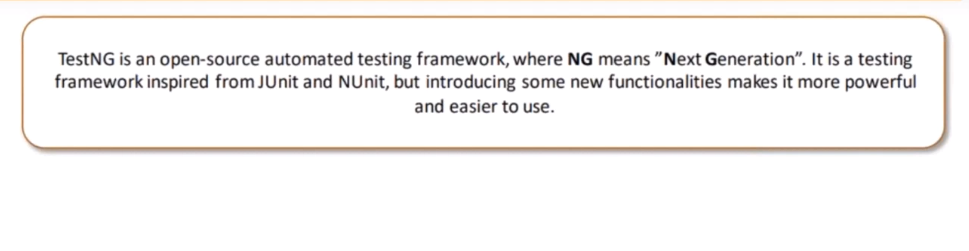
1. **doubleClick()**: Performs double click on the element
2. **clickAndHold()**: Performs long click on the mouse without releasing it
3. **dragAndDrop()**: Drags the element from one point and drops to another
4. **moveToElement()**: Shifts the mouse pointer to the center of the element
5. **contextClick()**: Performs right-click on the mouse

**Keyboard Actions:**

1. **sendKeys()**: Sends a series of keys to the element
2. **keyUp()**: Performs key release
3. **keyDown()**: Performs keypress without release

### 3.Testing and reporting in Selenium using TestNG

#### What is TestNG ?



**Main feature of TestNG is Annotations, whereby using annotations it is easy to create cases and give them priority.**

#### **Hands on : Setting up TestNG**

1. **Install TestNG at eclipse; follow this link** [**https://dzone.com/articles/how-to-install-testng-in-eclipse-step-by-step-guid**](https://dzone.com/articles/how-to-install-testng-in-eclipse-step-by-step-guid)
2. **Check your chrome browser version and download corresponding version from chrome website**

**Then follow below step**

**Eclipse -> Project explorer -> projectname -> src/main/java -> App.java**

Then write below command to import below dependencies.

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.By;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

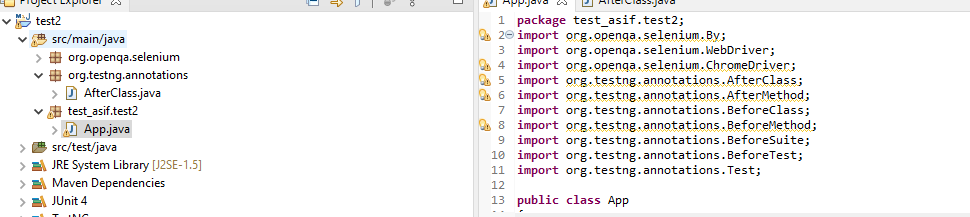
**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

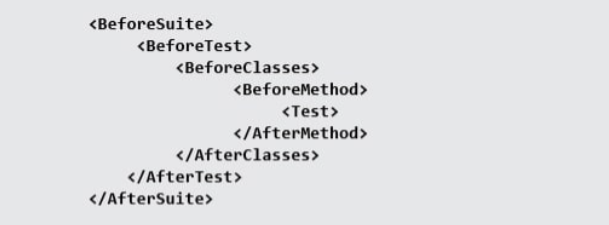
**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;



#### Annotations in TestNG

**The Workflow of TestNG is Shown Below:**



#### **Example of TestNg of annotations:**

**package** test\_asif.test2;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.ChromeDriver;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**public** **class** App

{

WebDriver driver;

@BeforeTest

**public** **void** before() {

System.***out***.println("Before test");

}

@Test

**public** **void** test() {

System.***out***.println("Test");

}

@BeforeSuite

**public** **void** beforesuit() {

System.***out***.println("Before suit");

}

@BeforeClass

**public** **void** beforeclass() {

System.***out***.println("Before class");

}

}

**This is the output**

