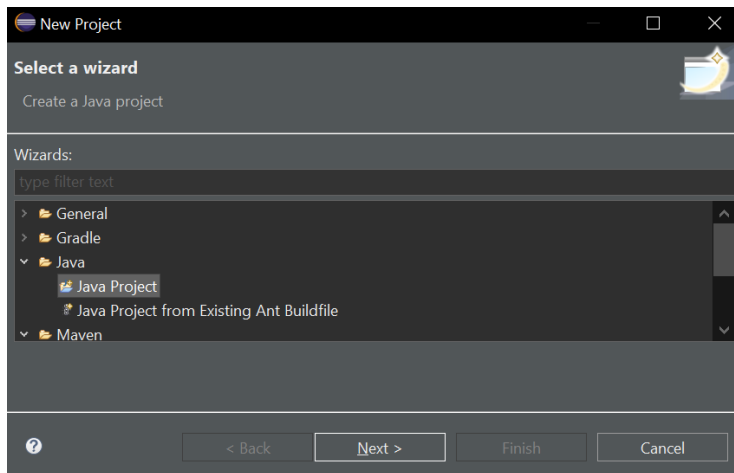
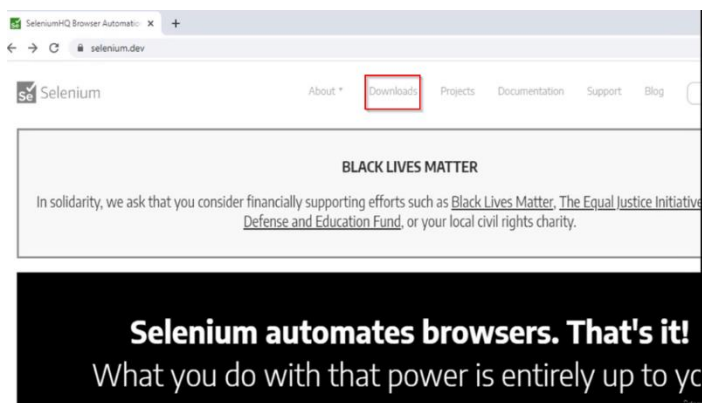


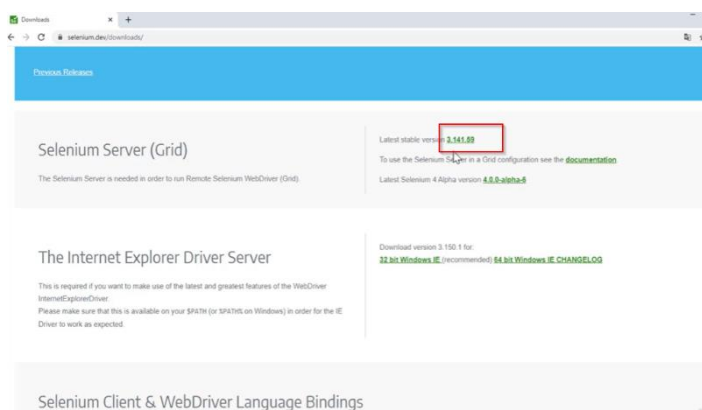
Creación de proyecto y primer script



Se creara un proyecto con java tradicional sin utilizar repositorios externo, para eso lo primero que hacemos es crear un Java Project

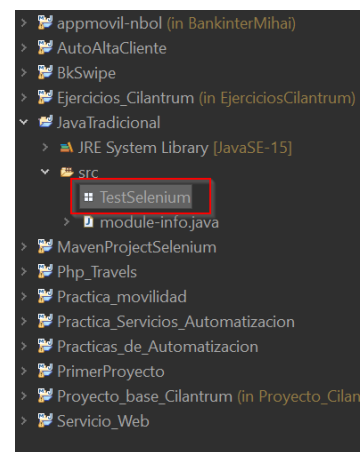


A continuación se pasara a descargar la librería de Selenium desde la propia web para eso vamos a Selenium.dev y dentro pulsamos Downloads



Nos descargamos la ultima versión que exista sobre Selenium Server(Grid)

Dentro del proyecto se crea un paquete que se llama TestSelenium

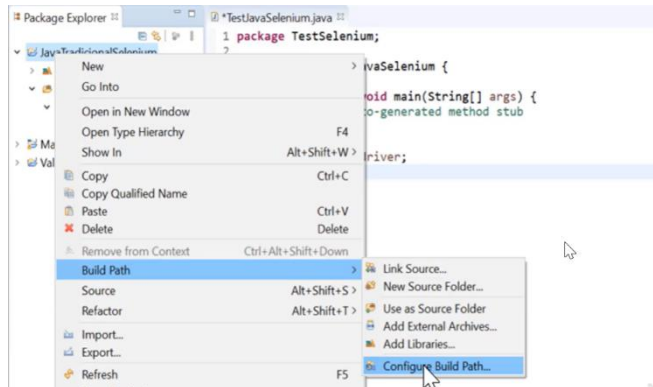


```

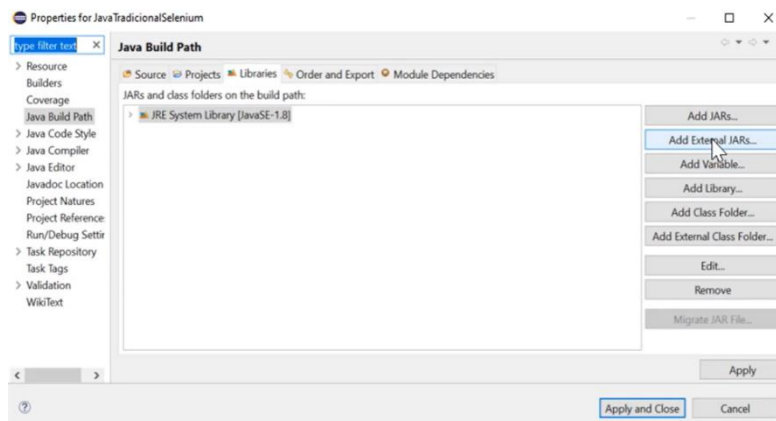
1 package TestSelenium;
2
3 public class TestJavaSelenium {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10
11
12
13 }
14
15 }
16

```

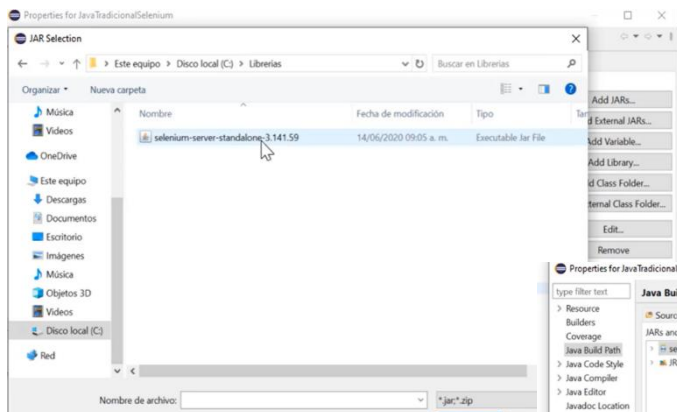
Dentro del paquete se crea la clase main



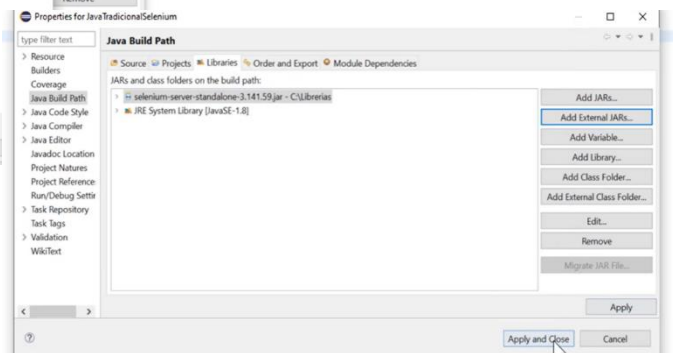
Para importar las librerías de Selenium, se hace click derecho sobre el proyecto uy nos vamos a Build Path/Configure Build Path..



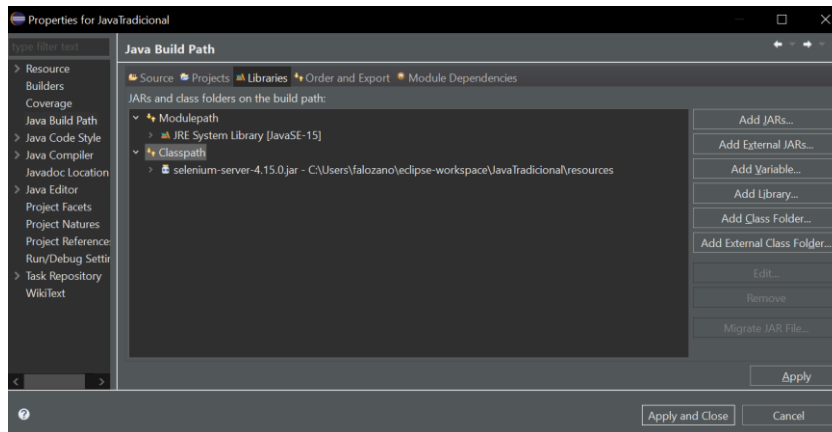
En la ventana que se abre pulsamos sobre la pestaña de Libraries y luego a Add External JARs...



Se busca la librería descargada con anterioridad

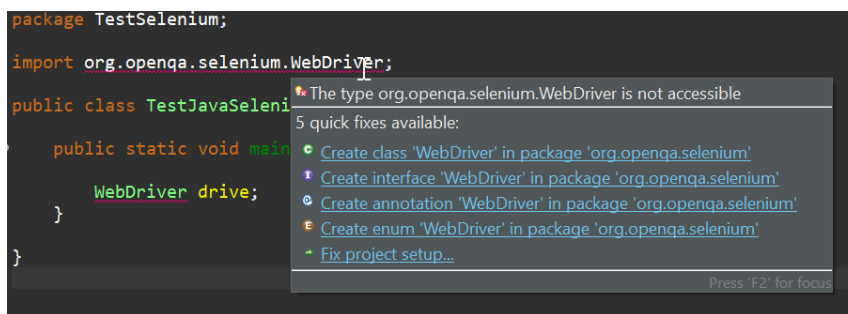


Para finalizar Apply and Close

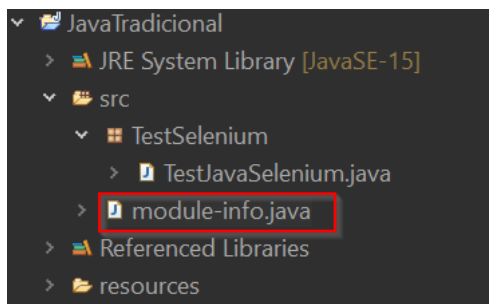


Si se muestra
Classpath, se añade
ahí

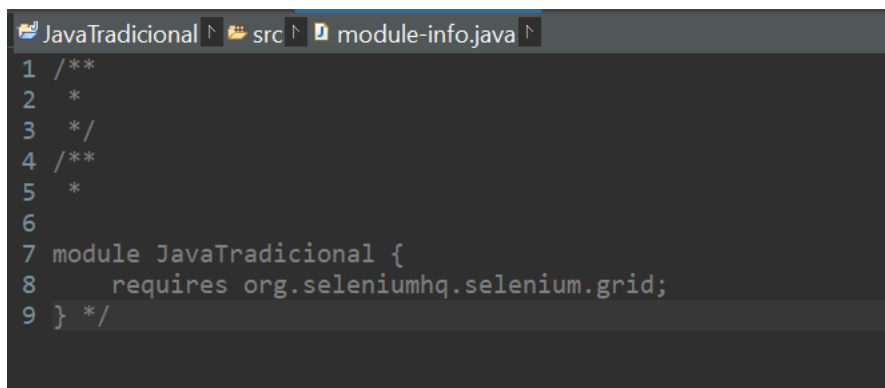
Si al importar la librería te da este error



En module-info.java



Comentamos todas las líneas y guardamos



El siguiente paso es crear el chromedriver <https://chromedriver.chromium.org/downloads>

Es el que se encargara de controlar el navegador Chrome en la ejecución

```
package TestSelenium;

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

public class TestJavaSelenium {

    public static void main(String[] args) {

        WebDriver driver;

        System.setProperty("webdriver", "resources\\chromedriver.exe");
        driver= new ChromeDriver();
        driver.get("http://www.google.es");

        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        driver.close();
    }

}
```

Este código está escrito en Java y utiliza Selenium WebDriver para realizar una prueba básica de automatización en un navegador Chrome. Vamos a desglosar el código paso a paso:

Explicación detallada:

1. Package y Importaciones:

- package TestSelenium;; Indica el paquete al que pertenece la clase TestJavaSelenium.
- import org.openqa.selenium.WebDriver; y import org.openqa.selenium.chrome.ChromeDriver;; Importan las clases necesarias de Selenium WebDriver para trabajar con un navegador web y específicamente con el controlador de Chrome.

2. Declaración de Variables:

- WebDriver driver;; Declara una variable llamada driver del tipo WebDriver. Esta variable se utilizará para interactuar con el navegador.

3. Configuración del Sistema para el Controlador de Chrome:

- System.setProperty("webdriver", "resources\\chromedriver.exe");; Configura el sistema para utilizar el controlador de Chrome (chromedriver.exe). Asegúrate de que la ruta proporcionada sea correcta y apunte al archivo ejecutable del controlador de Chrome.

4. Inicialización del Objeto ChromeDriver:

- driver = new ChromeDriver();; Inicializa el objeto driver como una instancia de ChromeDriver. Esto abre una instancia del navegador Chrome.

5. Navegación a una URL:

- driver.get("http://www.google.es");; Navega a la URL proporcionada ("http://www.google.es"). En este caso, se abre el sitio web de Google.

6. Espera de 5 Segundos:

- `Thread.sleep(5000);`: Hace que el hilo de ejecución espere durante 5 segundos. Esto se utiliza a veces para propósitos de demostración o para dar tiempo a que se cargue la página. En una prueba real, deberías considerar usar esperas explícitas para esperar a que elementos específicos de la página estén disponibles.

7. Manejo de Excepciones:







- `try {...} catch (InterruptedException e) {...}`: Maneja la excepción `InterruptedException` que podría ocurrir durante la espera. En este caso, simplemente imprime el rastro de la excepción.




8. Cerrar el Navegador:

- `driver.close();`: Cierra la ventana del navegador. Si estás utilizando `driver.quit();`, cerraría todas las ventanas abiertas del navegador. En este caso, solo hay una ventana abierta.

Este código básicamente inicia un navegador Chrome, navega a la página de Google, espera 5 segundos y luego cierra el navegador. Es un ejemplo simple de automatización web utilizando Selenium WebDriver con Java.

Como recomendación guardar el driver y la librería dentro del mismo proyecto en un directorio llamado `resources` así es más fácil identificar donde se encuentra

	<code>.settings</code>	01/12/2023 8:37	Carpeta de archivos	
	<code>bin</code>	01/12/2023 8:38	Carpeta de archivos	
	<code>resources</code>	01/12/2023 8:47	Carpeta de archivos	
	<code>src</code>	01/12/2023 8:38	Carpeta de archivos	
	<code>.classpath</code>	01/12/2023 8:37	Archivo CLASSPATH	1 KB
	<code>.project</code>	01/12/2023 8:37	Archivo PROJECT	1 KB

	<code>chromedriver.exe</code>	30/10/2023 11:38	Aplicación	16.468 KB
	<code>chromedriver-win64.zip</code>	01/12/2023 8:24	Carpeta comprimida ...	8.251 KB
	<code>selenium-server-4.15.0.jar</code>	01/12/2023 8:35	Archivo JAR	38.022 KB