Selenium: Automated Web UI Testing

Documentation on the E-Portfolio by Stefanie Neumann

Presentation Date: 2021-11-11

Professor: Dr. Berkling

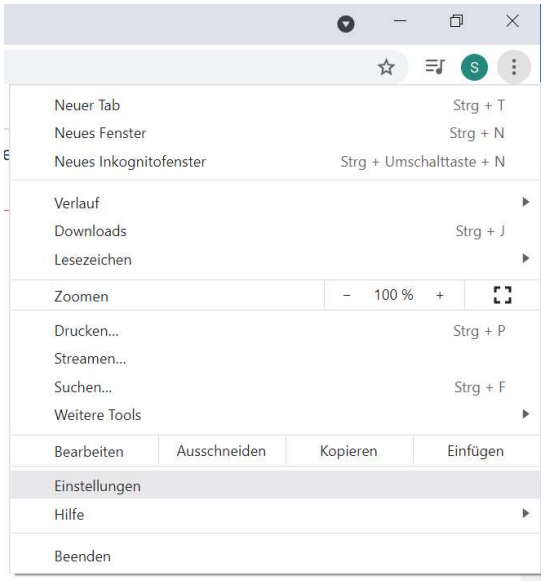
Table of Contents

Preparation

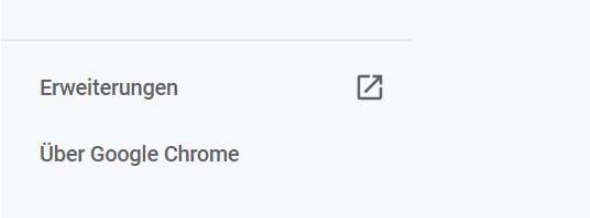
- An IDE that supports Java code has to be installed
- Chrome has to be installed

Install a Chrome Webdriver

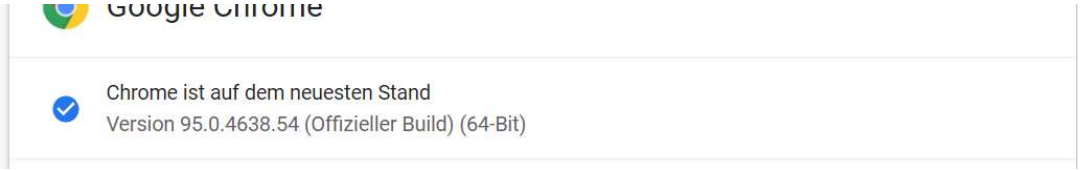
First of all, you have to find out, which Google Chrome version you have installed to know which WebDriver version you need. You can find your installed Chrome version in the Chrome browser settings (see the pictures underneath).



1) Open browser settings



2) Click "Über Google Chrome"



3) Find the installed Chrome version

You may then install the corresponding Chrome WebDriver. Find the latest Chrome WebDriver releases here (https://sites.google.com/chromium.org/driver/downloads).

Download Selenium Webdriver for Java

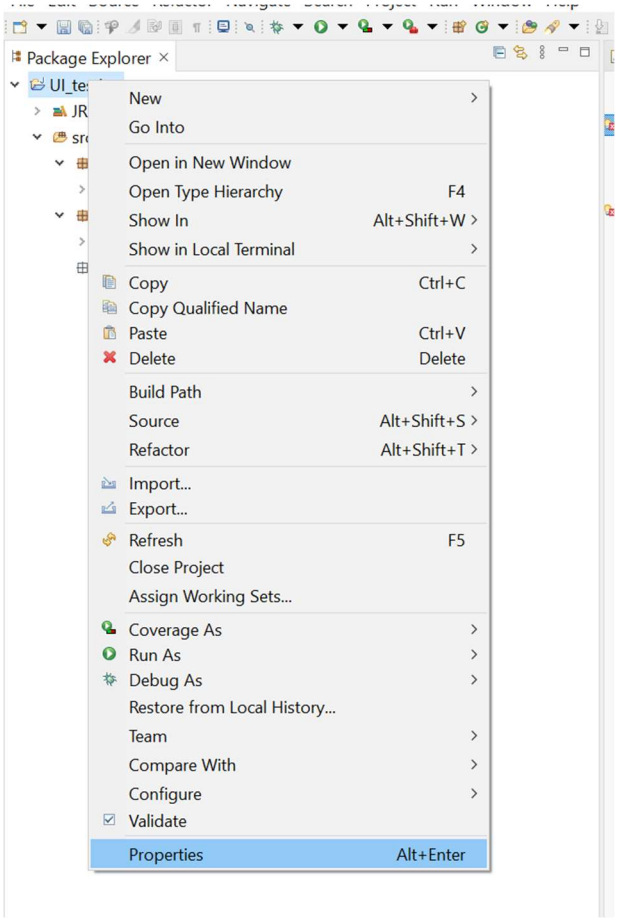
You can download the Selenium Webdriver libraries here: https://www.selenium.dev/downloads/

Make sure to select the language binding for Java if you want to use it in a Java project.

Selenium Webdriver version to download

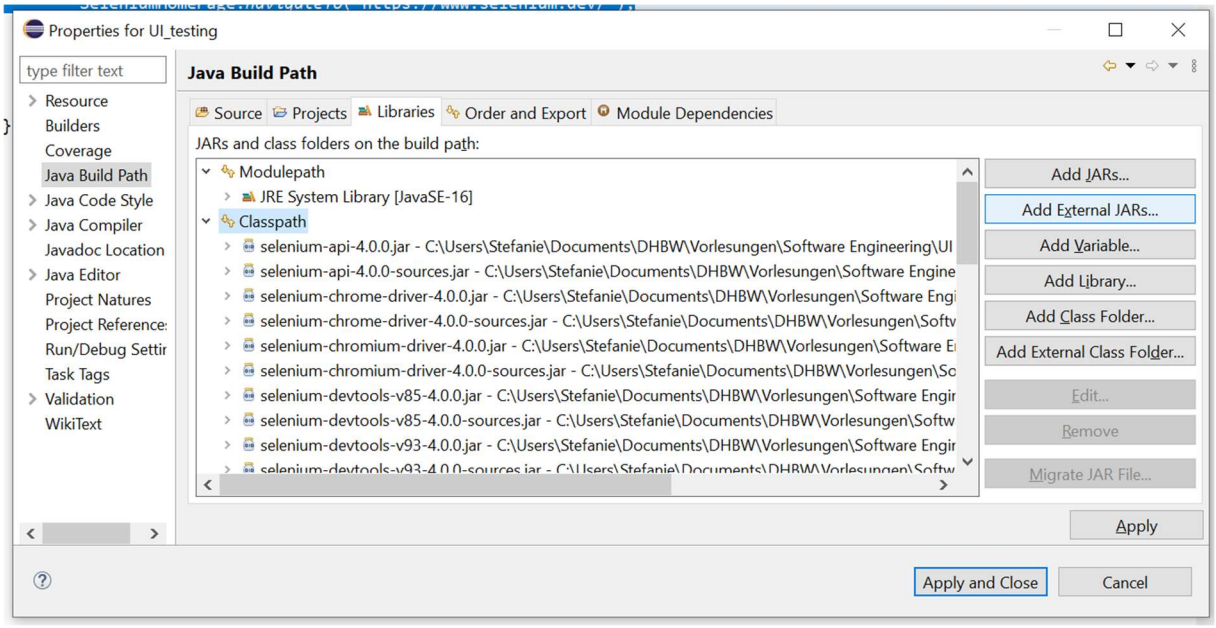
You then have to add the downloaded libraries into your Java project. Make sure to store the download somewhere you can find it again easily. Create a Java project in your wished IDE and add the external libraries to the project.

In order to add libraries in Eclipse, follow the underneath described steps:



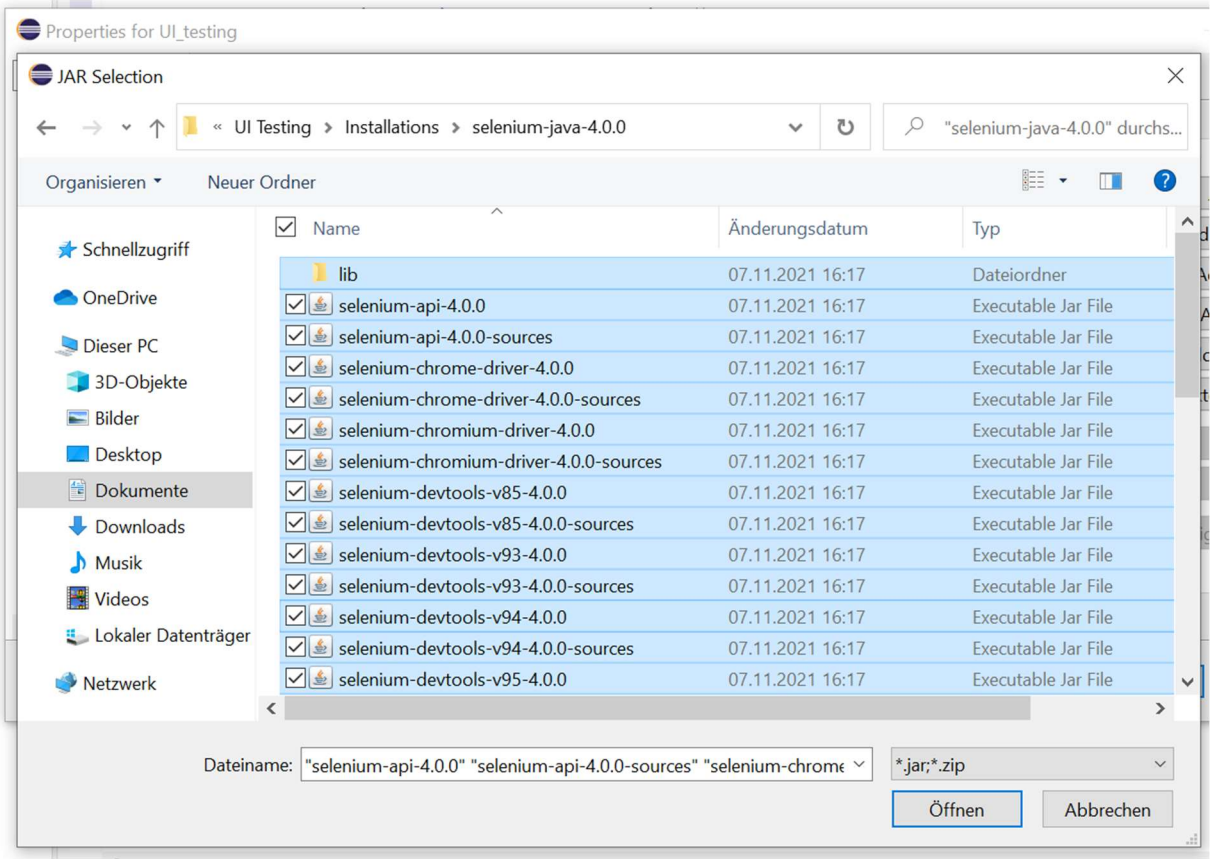
Add libraries into Java project: Step 1

Right click on your project. Select “*Properties*”.



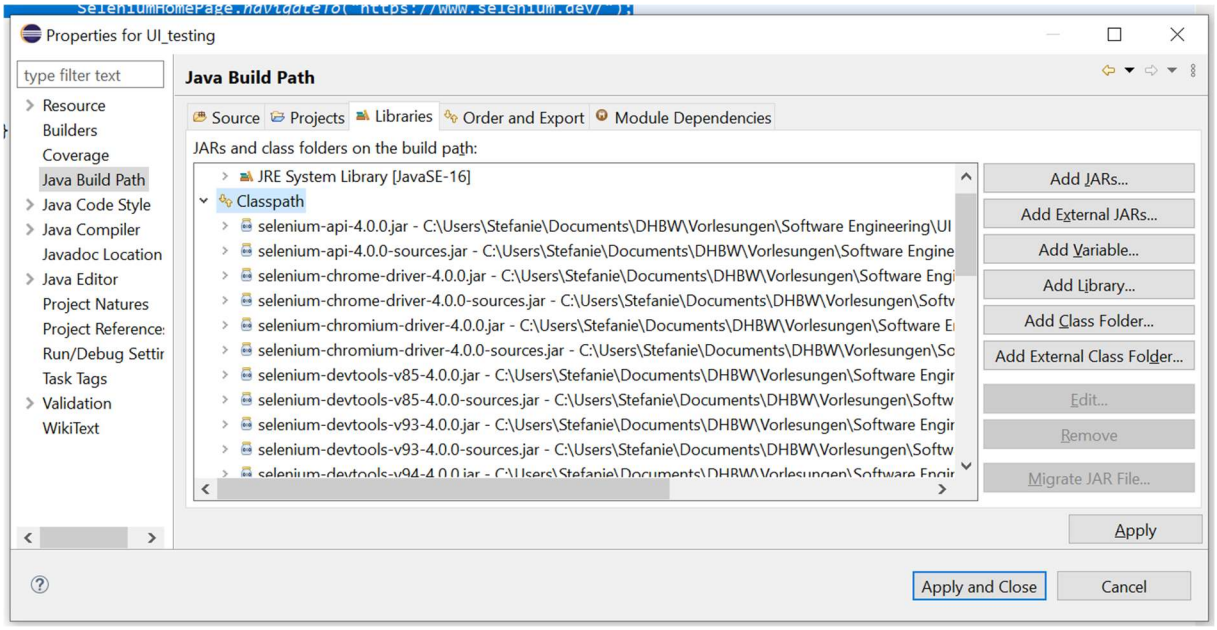
Add libraries into Java project: Step 2

Select *Java Build Path* > *Libraries* > *Classpath* > *Add External JARs*



Add libraries into Java project: Step 3

Make sure to select all files from the “selenium-java-4.0.0” folder and also all files inside of the “lib” folder.

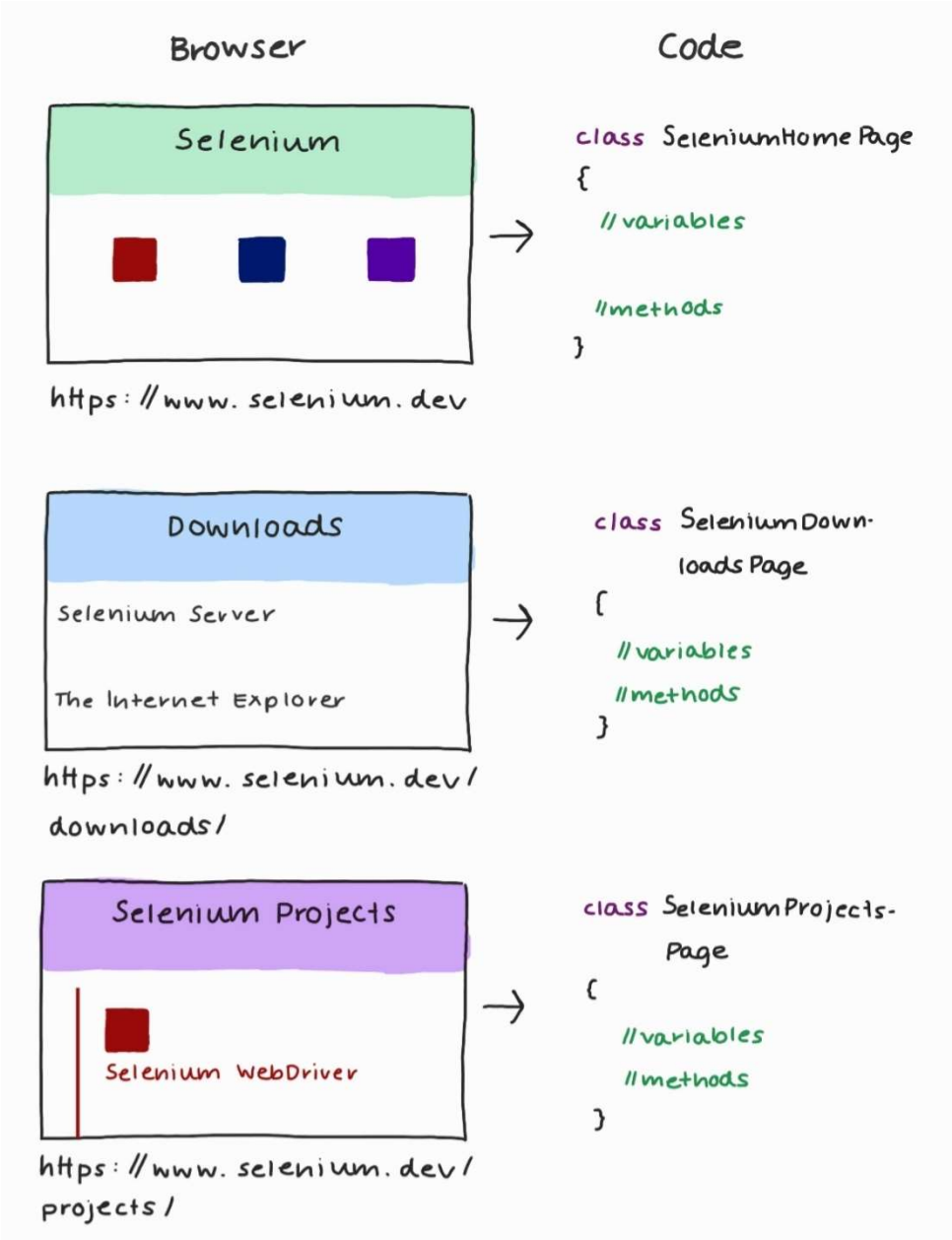


Add libraries into Java project: Step 4

Click “*Apply and Close*”.

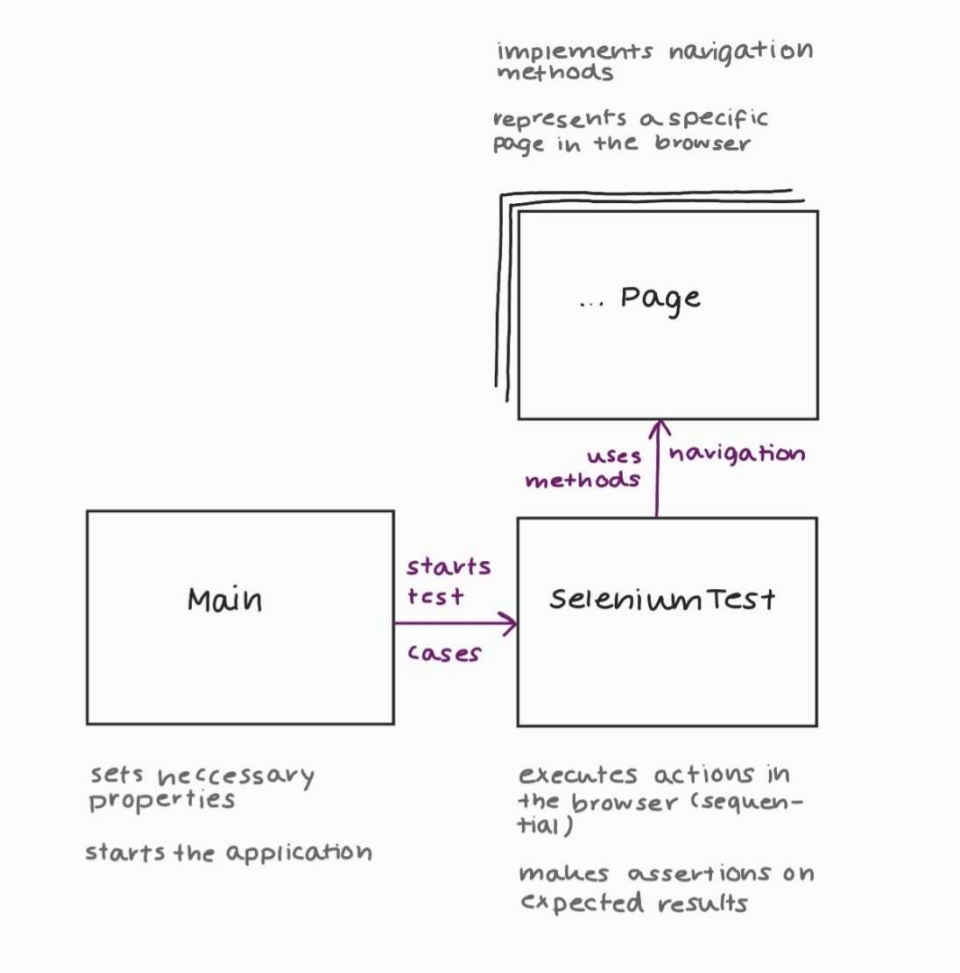
Design

The design of the implementation is guided by the Page Object Model (POM) design pattern. That means that each page of a browser application is represented by a single class in the implementation. This class contains navigation methods specifically for the page it represents and variables needed to search components of the page.



Page Object Model: One class for each page

Tests are implemented separately. A test simply is a sequence of actions that are performed in the browser. Test methods use the navigation methods implemented in the page classes to execute actions in the browser. After execution it has to be checked whether the desired result has occurred. For that an assertion has to be made.

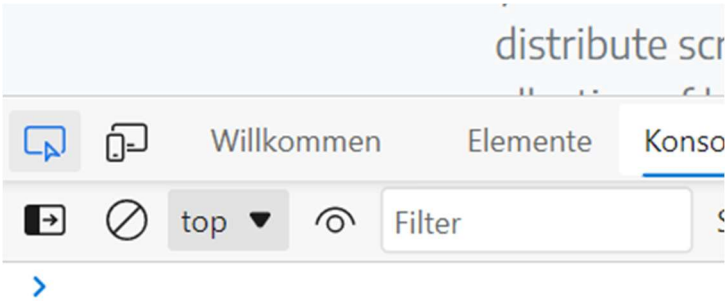


Page Object Model: Connection between classes

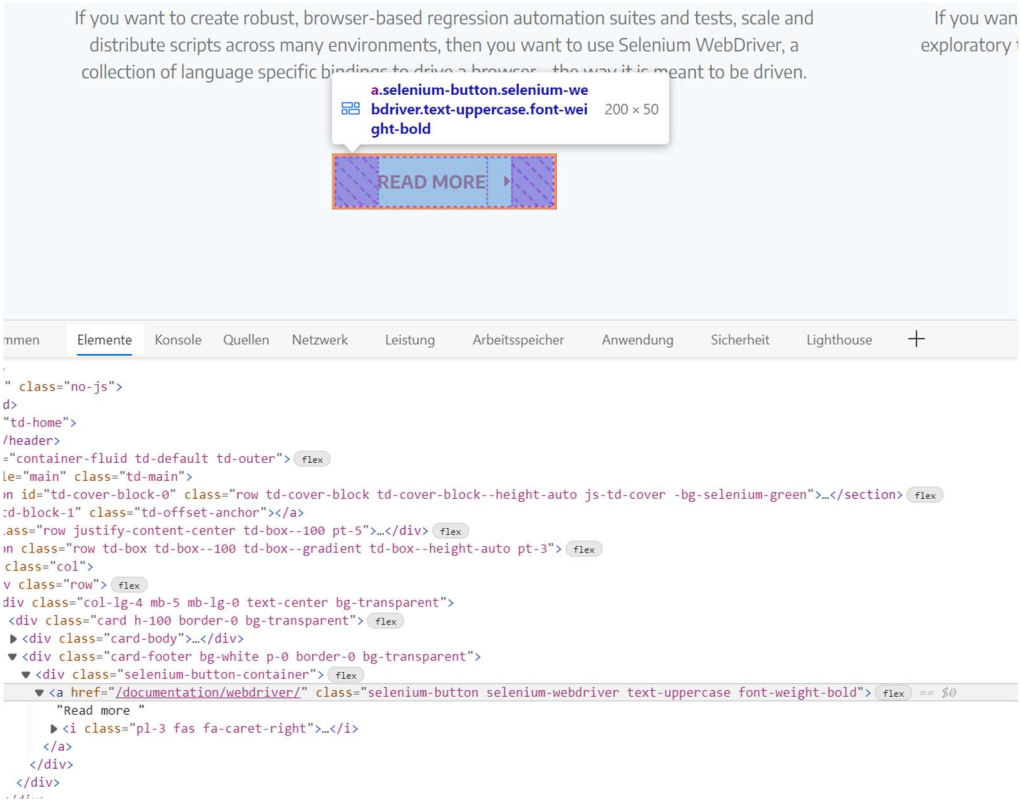
Implementation

Find Elements

Open developer tools in your browser by pressing „f12“

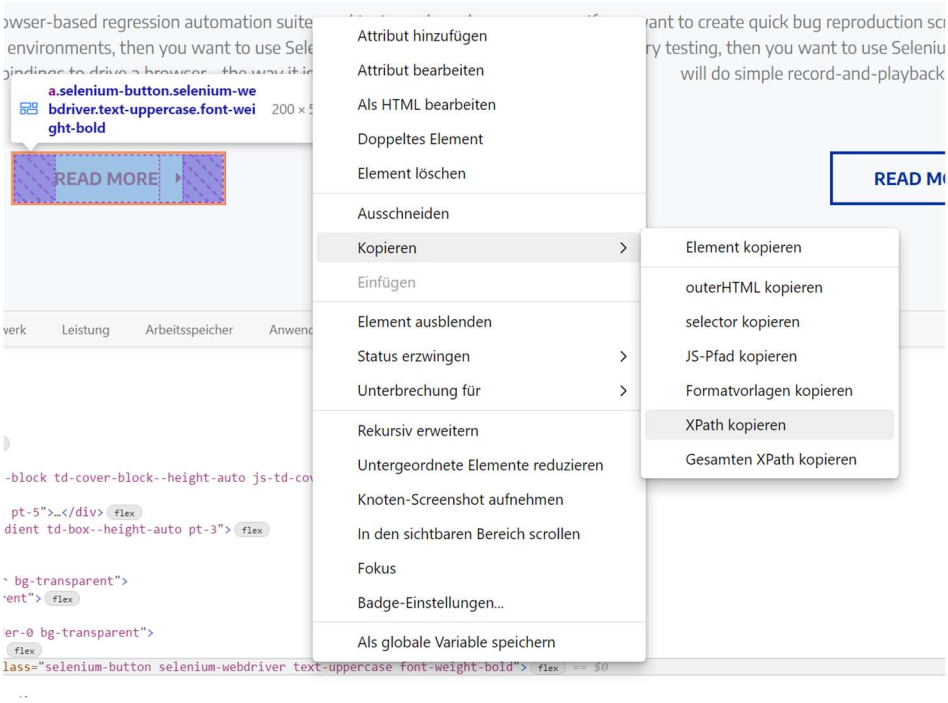


Click the symbol in the upper left corner of the developer tool window (blue marked in the above displayed picture). You can now select elements on the page in the browser window and get the according html tag displayed in the developer tool window.

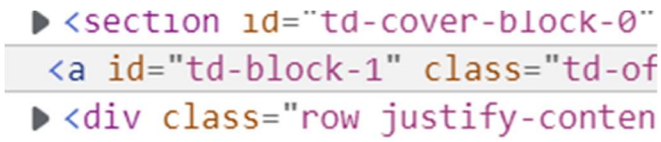


Button element "READ MORE" selected

You can copy the XPath of an element by right clicking on the element’s tag’s entry in the developer tool window, selecting “Copy” and then “Copy XPath”. Or you can use the elements id, if this attribute is set.



Copy XPath



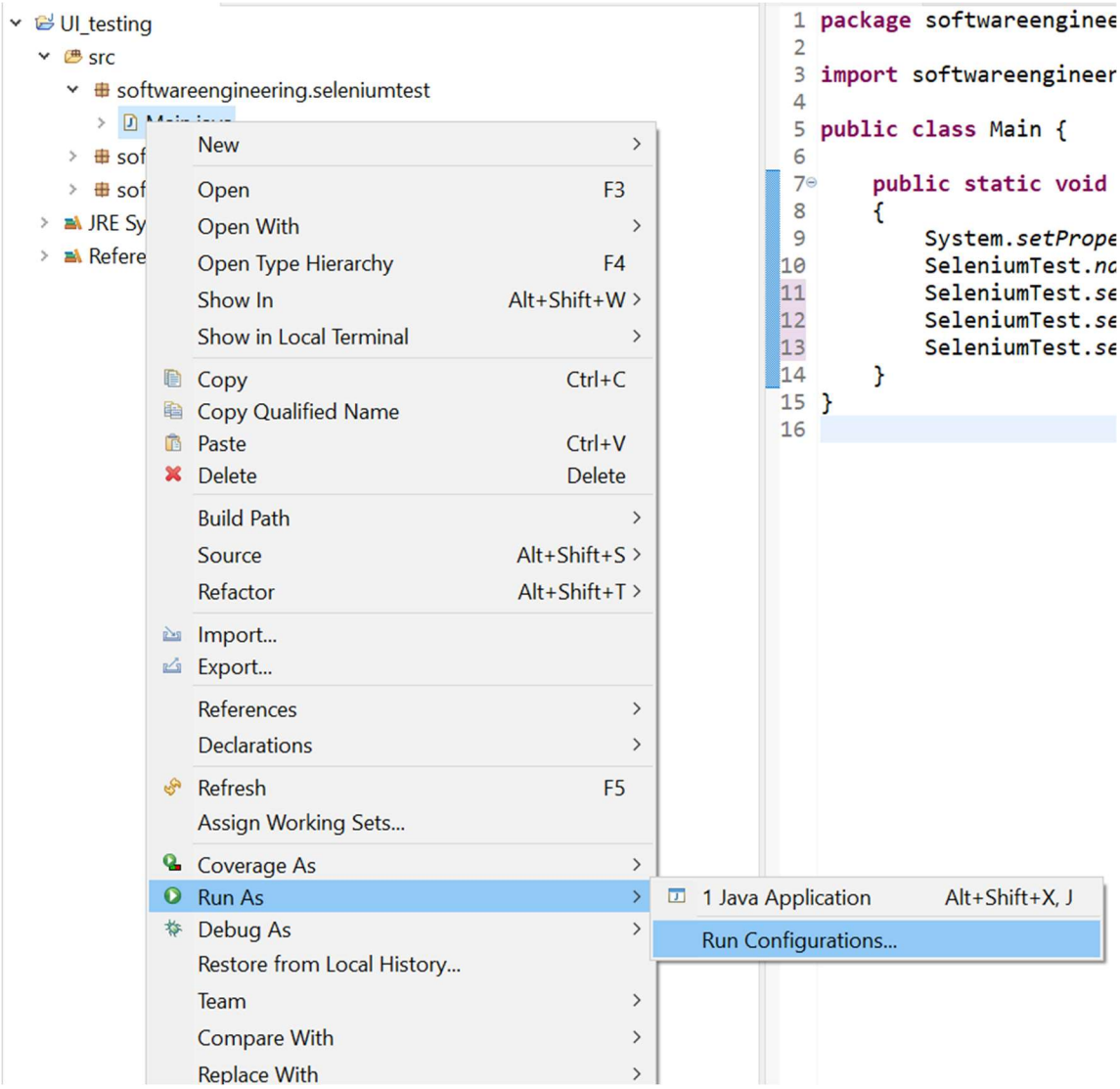
Find id

Execution

In order to start the application, run the main method. A browser window will then open. In it the actions implemented in each test case are executed. You can see whether the implementation is correct by whether the actions are executed as desired. If no assertion error occurs during execution the tests run successfully.

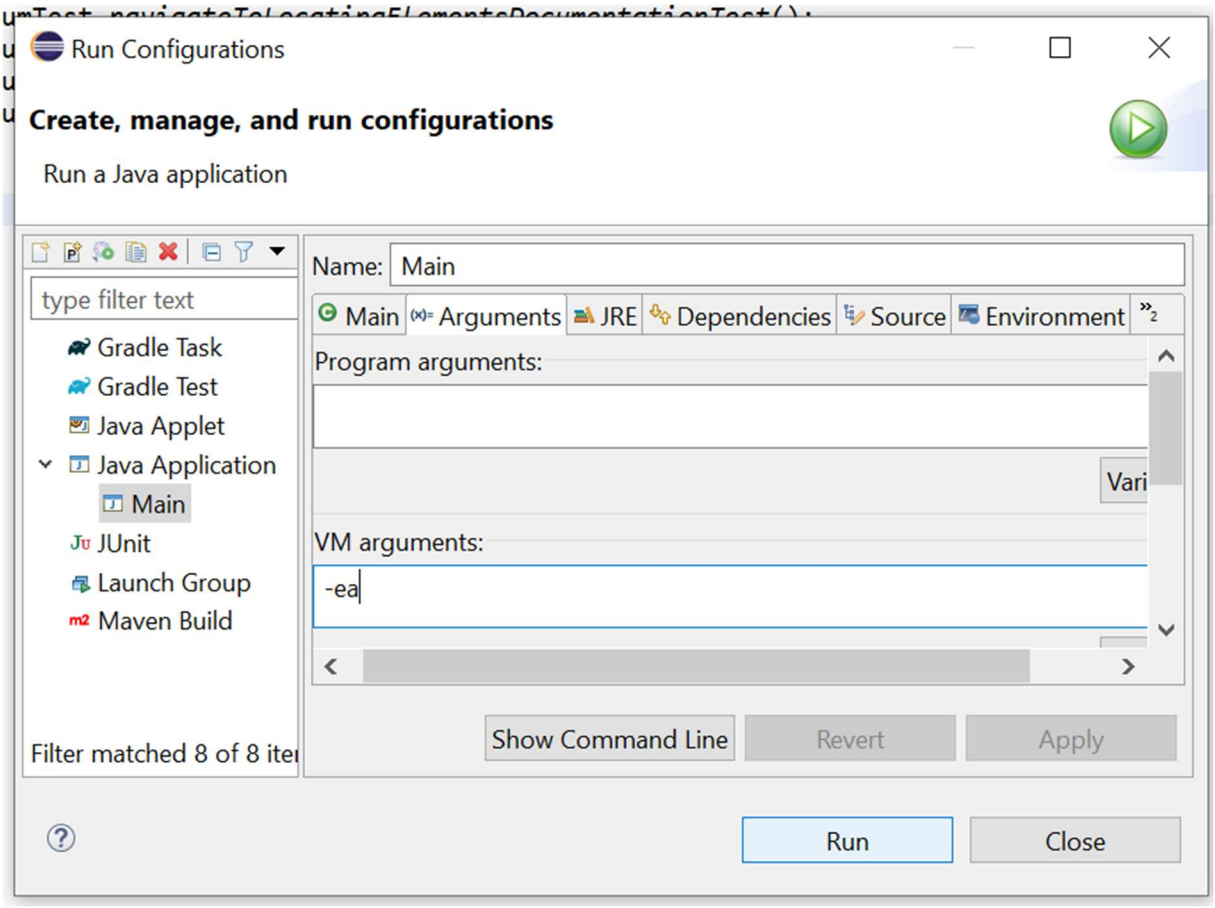
Use Assertions in Java

In order to get assertion errors displayed in the console, you have to set the parameter "-ea" in your Run configurations. Follow the steps underneath for further instructions.



Enable Java Assertions in Eclipse: Step 1

Right click on your Main class > Select "*Run As*" > "*Run Configurations…*"



Enable Java Assertions in Eclipse: Step 2

Select the "*Arguments*" tab and enter "-ea" into the VM arguments field. Click "*Run*" to run the Main class.