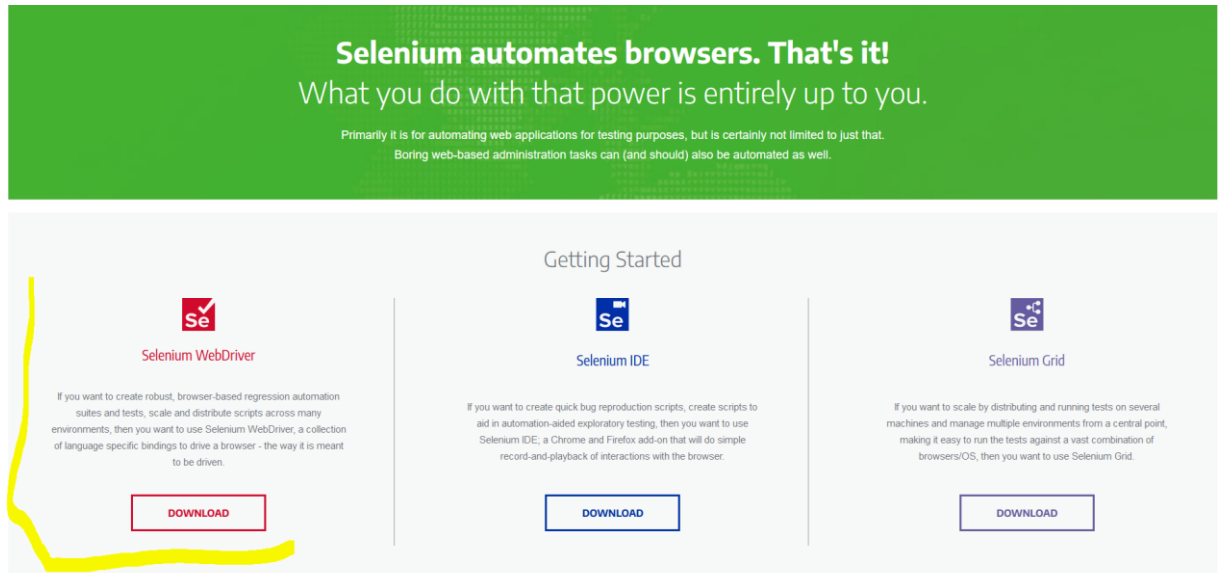


Anleitung zur Nutzung von Selenium

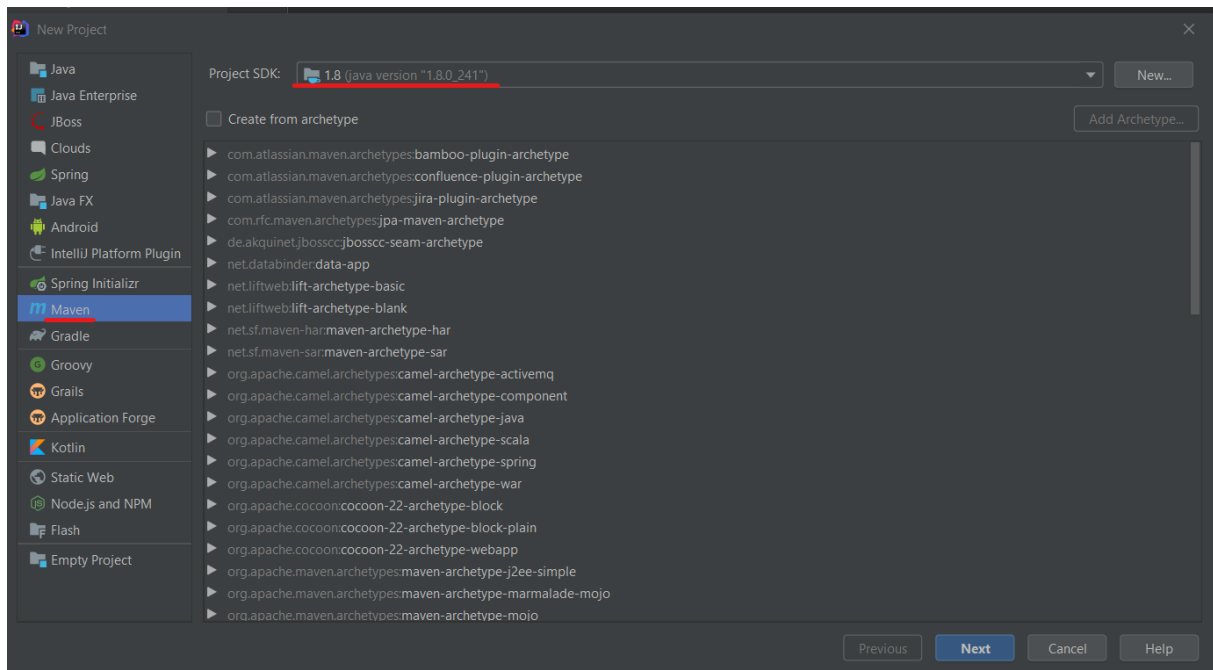
Das Ganze geschieht m Beispiel der DIE IntelliJ.

Alle Screenshots entstammen diesem Programm.

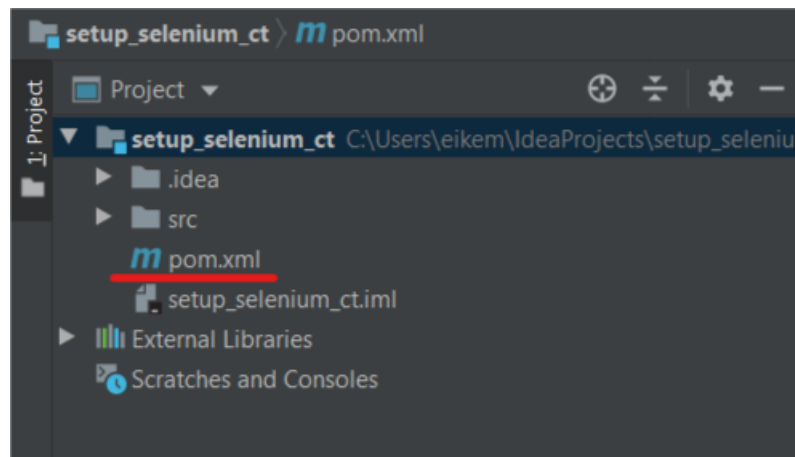
1. Den Selenium Webdriver runterladen



2. Neues Java 8 Projekt erstellen. Maven als Build-Tool auswählen!



3. In die Maven Pom wird die aktuelle Selenium Version eingetragen: Im Projekt nach der pom.xml suchen.



Diese Datei öffnen und folgendes eintragen:

(<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.141.59>)

<dependencies>

...

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

...

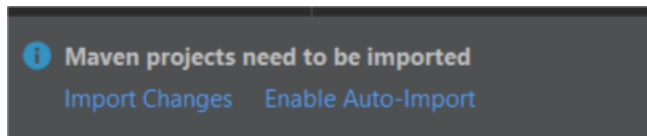
</dependencies>

Oder statt der Version 3.141.59 eine andere aktuelle Version aus dem Maven Repository.

Die Punkte stehen dort, weil zur Zeit keine weiteren Dependencies benötigt werden.

```
m pom.xml x
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <dependencies>
8          <dependency>
9              <groupId>org.seleniumhq.selenium</groupId>
10             <artifactId>selenium-java</artifactId>
11             <version>3.141.59</version>
12         </dependency>
13     </dependencies>
14
15     <groupId>org.example</groupId>
16     <artifactId>setup_selenium_ct</artifactId>
17     <version>1.0-SNAPSHOT</version>
18
19
20 </project>
```

Unten rechts sollte nun ein kleines Fenster auftauchen:



Klickt nun einfach auf „import changes“.

Sollte IntelliJ behaupten, es gäbe kein Artefakt mit dem Namen „org.seleniumhq.selenium“ klickt auch einfach auf „Import changes“.

Sollte das nicht klappen, schaut auf der Seite der Maven Repositories nach einer anderen Version!

4. Geckodriver für Firefox oder Chromedriver für Chrome herunterladen

Je nach Browser den ihr zum Testen verwenden wollt, benötigt ihr noch einen Driver, damit der Browser automatisiert bedient werden kann.

Für Firefox den „Geckodriver“:

<https://github.com/mozilla/geckodriver/releases>

Sucht euch hier eine passende Version für euer Betriebssystem aus, ladet sie herunter und entpackt sie.

Für Chrome den „Chromedriver“

<https://chromedriver.chromium.org/downloads>

Sucht euch hier eine passende Version für euer Betriebssystem aus, ladet sie herunter und entpackt sie.

5. Den Driver einbinden

Nachdem einer der beiden Driver heruntergeladen und entpackt wurde, merkt euch den Pfad dahin!

C:\Users\eikem\OneDrive\Documents\HAW\geckodriver					
	Name	Status	Änderungsdatum	Typ	Größe
riff	geckodriver.exe	✓	27.03.2020 10:46	Anwendung	3.483 KB

Tragt diesen in euerem Java Projekt ein:

```
System.setProperty("webdriver.gecko.driver", "C:\\Users\\eikem\\OneDrive\\Documents\\HAW\\geckodriver\\geckodriver.exe");
```

```
m pom.xml x main.java x
1 import org.openqa.selenium.WebDriver;
2 import org.openqa.selenium.firefox.FirefoxDriver;
3
4 public class main {
5
6     public static void main(String[] args){
7         System.setProperty("webdriver.gecko.driver", "C:\\Users\\eikem\\OneDrive\\Documents\\HAW\\geckodriver\\geckodriver.exe");
8         WebDriver driver=new FirefoxDriver();
9         driver.get("https://tessa.haw-hamburg.de/");
10        driver.close();
11    }
12
13 }
```

Dann nur noch loslegen!

QUELLEN:

<https://github.com/mozilla/geckodriver/releases>

<https://chromedriver.chromium.org/downloads>

https://www.selenium.dev/documentation/en/getting_started_with_webdriver/locating_elements/

https://www.selenium.dev/documentation/en/getting_started_with_webdriver/performing_actions_on_the_aut/

<https://www.selenium.dev/documentation/en/>