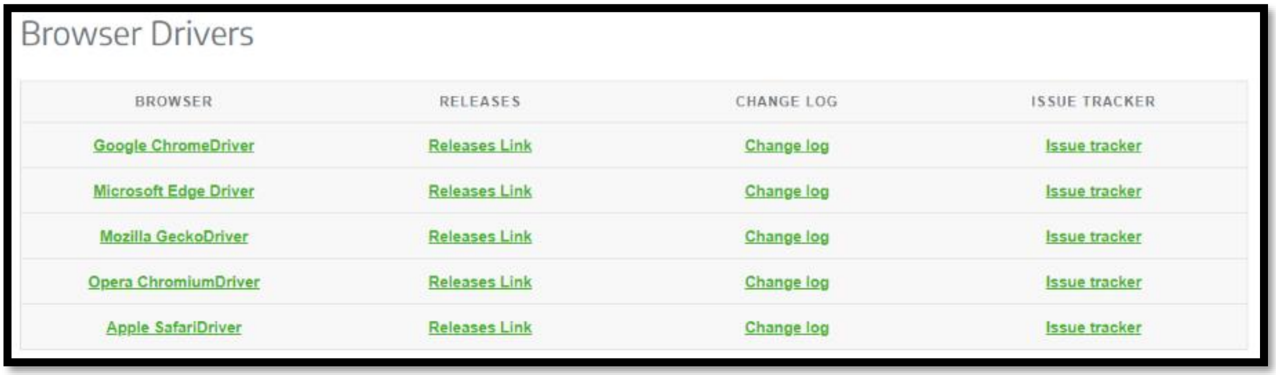
Rex
Jones II

(Transcript) WebDriverManager Bypass Drivers

In this video, I am going to speak about WebDriverManager. WebDriverManager is an API that allows us to skip downloading a driver such as chromedriver, edgedriver, and geckodriver. For example, you know how we can go to Change Log for the driver and we download the driver depending on our browser version. We can bypass downloading the driver and also bypass writing System.setProperty.

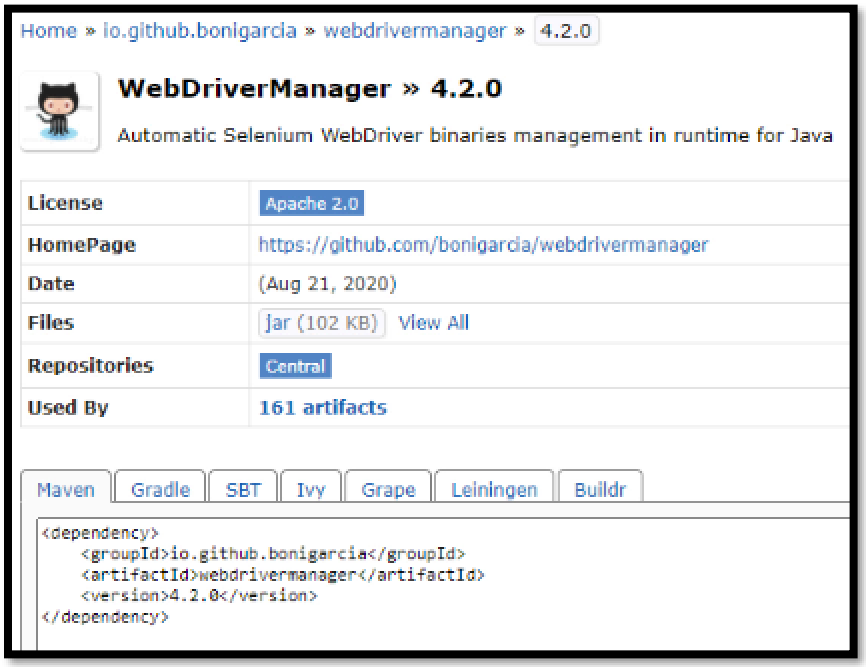


Browser Drivers

BROWSER	RELEASES	CHANGE LOG	ISSUE TRACKER
Google ChromeDriver	Releases Link	Change log	Issue tracker
Microsoft Edge Driver	Releases Link	Change log	Issue tracker
Mozilla GeckoDriver	Releases Link	Change log	Issue tracker
Opera ChromiumDriver	Releases Link	Change log	Issue tracker
Apple SafariDriver	Releases Link	Change log	Issue tracker

If you are new to my YouTube videos, consider subscribing to my channel and clicking the bell icon. You can also follow me on Twitter, connect with me on LinkedIn and Facebook. After this video, I am going to release the code and create a transcript and place them on GitHub.

Let's go to Maven's Repository, search for WebDriverManager, select WebDriverManager and we see different versions. Version 4.2 is the most recent version and it has a dependency. Copy the dependency and go to



WebDriverManager » 4.2.0

Automatic Selenium WebDriver binaries management in runtime for Java

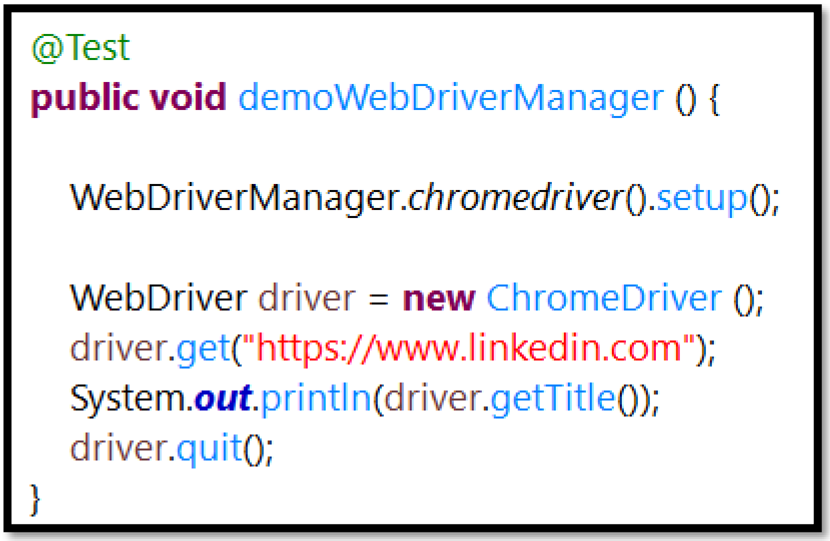
License	Apache 2.0
HomePage	https://github.com/bonigarcia/webdrivermanager
Date	(Aug 21, 2020)
Files	jar (102 KB) View All
Repositories	Central
Used By	161 artifacts

```
<dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>4.2.0</version>
</dependency>
```

the pom.xml file, we paste the dependency and we enter CTRL + SHIFT + F to format the file. Also save the file. Now, let's write our Test Script. WebDriverManager dot and we see different drivers. Let's select chromedriver() then setup() the driver. Here's how the process of WebDriverManager works: First, it checks the browser version installed on our machine then it checks the driver version. If the driver version is unknown. It will use the most recent driver version.

Selenium WebDriver still requires a driver in order to use Firefox, Chrome, or any other browser. It requires a driver because that's how Selenium controls the browser. Therefore, we write

WebDriver driver = new ChromeDriver(); then import the drivers.
The driver will load LinkedIn by writing driver.get("https://www.linkedin.com");
Let's also print the page title because Selenium will execute so fast that we may not see the page load.
sysout(driver.getTitle); That will verify we actually loaded the page.
then we quit the driver driver.quit;



```java
@Test
public void demoWebDriverManager () {

   WebDriverManager.chromedriver().setup();

   WebDriver driver = new ChromeDriver ();
   driver.get("https://www.linkedin.com");
   System.out.println(driver.getTitle());
   driver.quit();
}
```

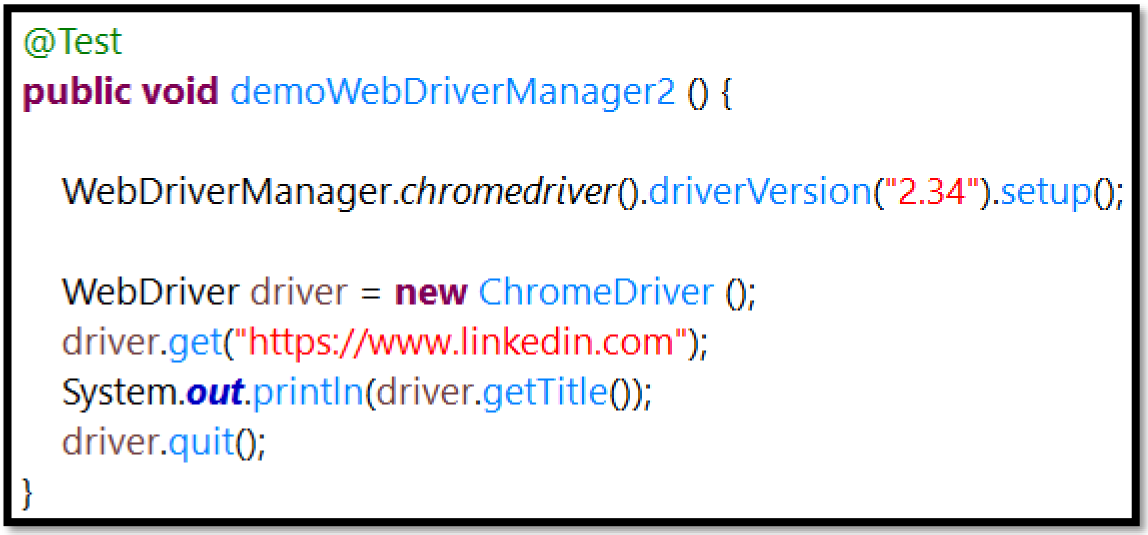
Let’s Run. LinkedIn loaded and it closed. The Console shows LinkedIn: Log In or Sign Up.

We can go to the HomePage of Boni Garcia, who is the creator of WebDriverManager to see the dependencies and view some of his notes. We see WebDriverManager is a library that allows us to automate the management of the drivers required by Selenium WebDriver.

Here is the Table of Contents: Motivation shows the System.setProperty for each driver. It talks about how it’s annoying to manually check for a new version of the driver. WebDriverManager is automated. Have you ever experienced a situation where you execute your Test Scripts and they pass then the same Test Scripts start failing due to your browser? It’s possible your browser updated automatically to the next version which makes your browser and driver not compatible. The WebDriverManager API verifies your browser version then it uses the correct driver version.

We can use WebDriverManager different ways: as a Java Dependency, as a Command Line Interface, as a Server, as an Agent, and as a Container. The Test Script I executed used the Java Dependency. We can also use WebDriverManager for Gradle.

Now, you may work at a company that disables the browsers from updating automatically to the next version. WebDriverManager takes care of that scenario if we need our driver to remain at the same version. Let me show you by making this method 1 then copying and pasting new method. Change 1 to 2 and we are going to replace this line by writing WebDriverManager.chromedriver().driverVersion(). Let’s say it’s 2.34 we want it to remain at then we set up that driver.

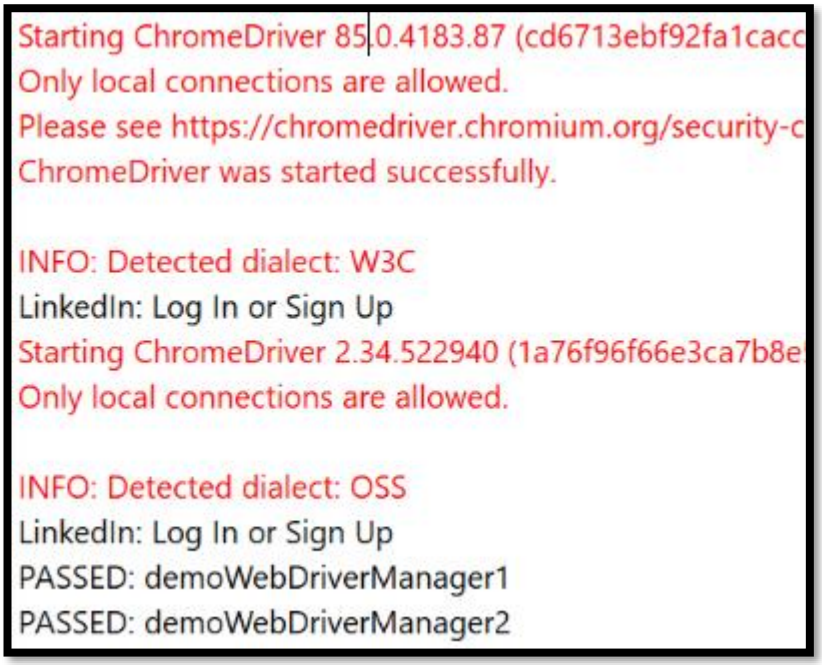


```
@Test
public void demoWebDriverManager2 () {

    WebDriverManager.chromedriver().driverVersion("2.34").setup();

    WebDriver driver = new ChromeDriver ();
    driver.get("https://www.linkedin.com");
    System.out.println(driver.getTitle());
    driver.quit();
}
```

Let’s run. LinkedIn loaded and it loaded again. Let’s look at the results. We see both methods Passed. With the first method loading ChromeDriver 85 and the second driver is ChromeDriver 2.34. We see the page title for both methods.



```
Starting ChromeDriver 85.0.4183.87 (cd6713ebf92fa1cacc
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-c
ChromeDriver was started successfully.

INFO: Detected dialect: W3C
LinkedIn: Log In or Sign Up
Starting ChromeDriver 2.34.522940 (1a76f96f66e3ca7b8e
Only local connections are allowed.

INFO: Detected dialect: OSS
LinkedIn: Log In or Sign Up
PASSED: demoWebDriverManager1
PASSED: demoWebDriverManager2
```

That’s it for WebDriverManager and Thanks for watching. I’ll see you in the next video.

Resources

- WebDriverManager GitHub https://github.com/bonigarcia/webdrivermanager

- Raghav Pal https://www.youtube.com/watch?v=8vWTgyoG0nc&t=1s
- Mukesh Otwani https://www.youtube.com/watch?v=Pb8gsLeSOFo
- Environment Variables C:\Users\Rex Allen Jones II\Downloads\Drivers