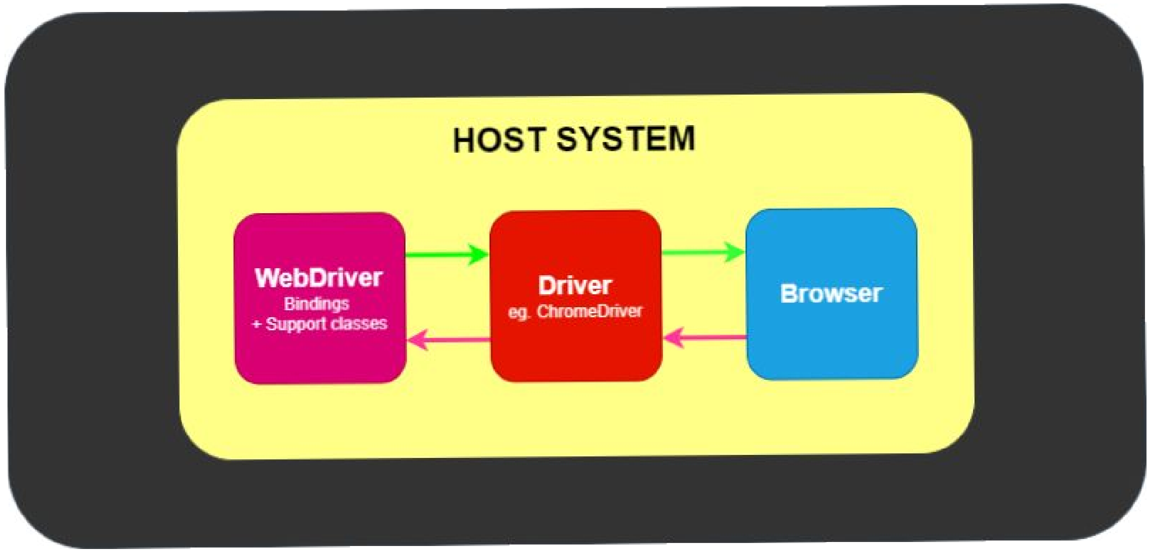
Selenium browser automation

- 1. Selenium **Webdriver** with a language library, e.g. Java

  (Xpath very useful)

- 2. Selenium **IDE** for GUI lovers (and siderunners)

- 1. Selenium Webdriver with a language library, e.g. Java



Why use Selenium?

- OSS (Apache license) and free for web testing
- Fast and reliable, once you know its quirks
- Front end tests, giving you the assurance that changes aren’t breaking things
- Can be put in a CI pipeline (e.g Gitlab)
- Works with different browsers
- Works with different languages
- For creating bulk data, even performance testing

- *Show video - “create job”*

Distro - Ubuntu 20.03

Browser - Firefox (v.83 - 93)

Java version: Installed Oracle (JDK 8.14)

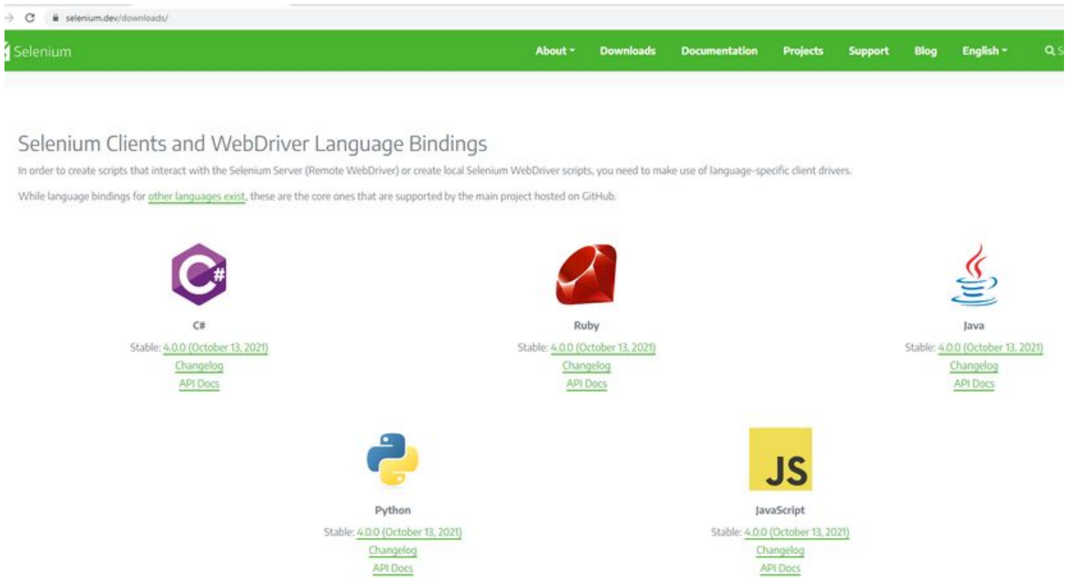
IDE - Installed IntelliJ Idea IDE (v.20)

We need:
Java language bindings for Selenium
A webdriver library, i.e. for Firefox (versions very important!)
Any other packages, e.g. JUnit testing framework

Install the webdriver language bindings

Went here - https://www.selenium.dev/downloads/

- under Selenium Clients and WebDriver Language Bindings
- downloaded for Java 4.0
- unzipped download & put it in /opt/selenium, set permissions



Selenium Clients and WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.

While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on GitHub.

C#
Stable: 4.0.0 (October 13, 2021)
Changelog
API Docs

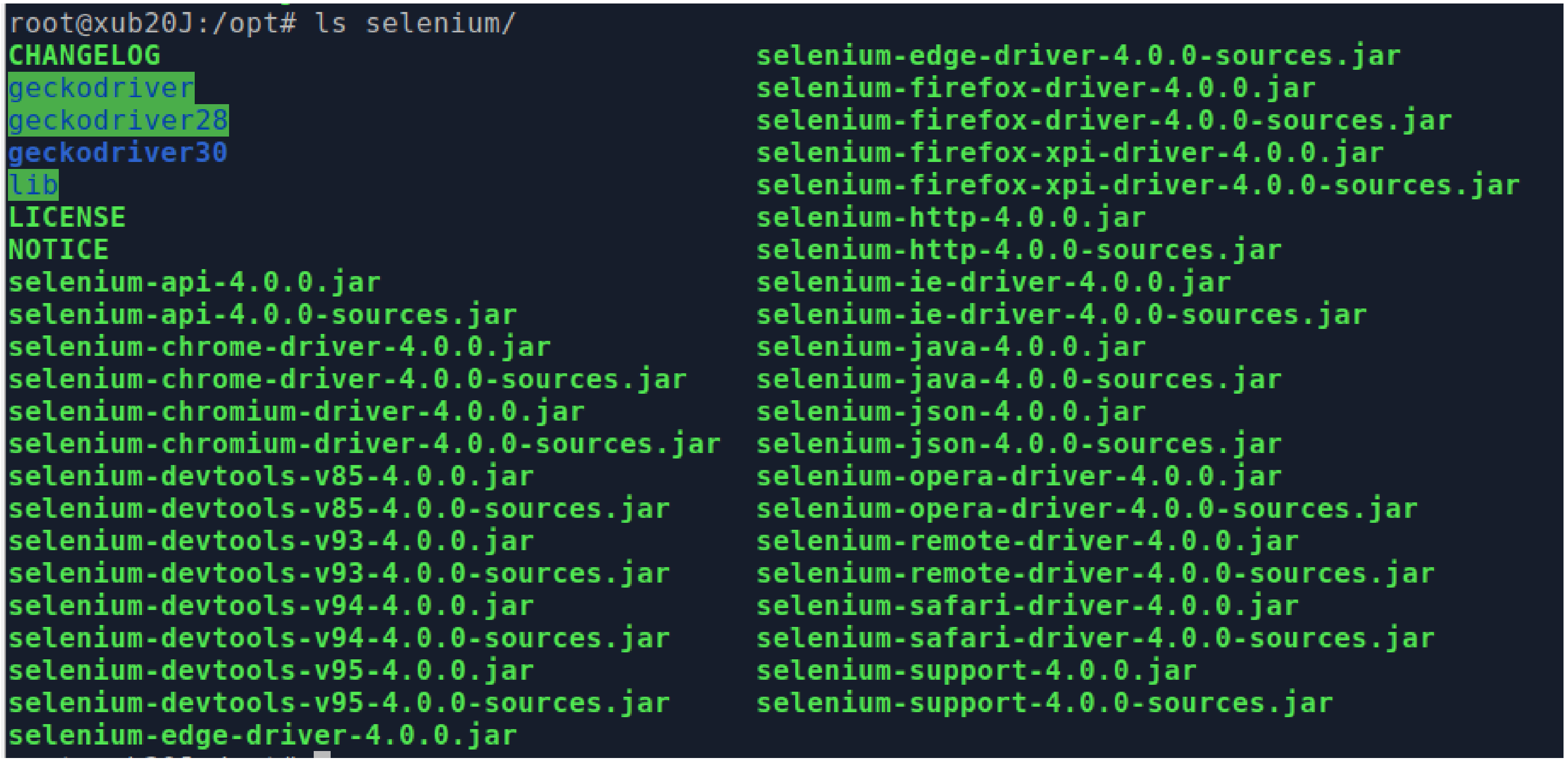
Ruby
Stable: 4.0.0 (October 13, 2021)
Changelog
API Docs

Java
Stable: 4.0.0 (October 13, 2021)
Changelog
API Docs

Python
Stable: 4.0.0 (October 13, 2021)
Changelog
API Docs

JavaScript
Stable: 4.0.0 (October 13, 2021)
Changelog
API Docs

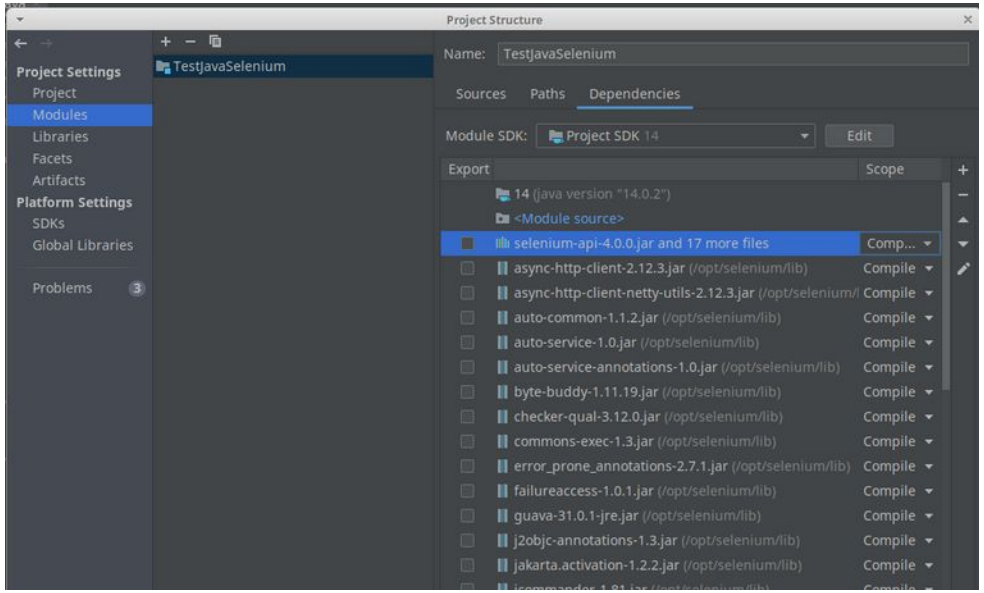
In /opt/selenium



```
root@xub20J:/opt# ls selenium/
CHANGELOG                                   selenium-edge-driver-4.0.0-sources.jar
geckodriver                                 selenium-firefox-driver-4.0.0.jar
geckodriver28                               selenium-firefox-driver-4.0.0-sources.jar
geckodriver30                               selenium-firefox-xpi-driver-4.0.0.jar
lib                                         selenium-firefox-xpi-driver-4.0.0-sources.jar
LICENSE                                     selenium-http-4.0.0.jar
NOTICE                                      selenium-http-4.0.0-sources.jar
selenium-api-4.0.0.jar                      selenium-ie-driver-4.0.0.jar
selenium-api-4.0.0-sources.jar              selenium-ie-driver-4.0.0-sources.jar
selenium-chrome-driver-4.0.0.jar            selenium-java-4.0.0.jar
selenium-chrome-driver-4.0.0-sources.jar    selenium-java-4.0.0-sources.jar
selenium-chromium-driver-4.0.0.jar          selenium-json-4.0.0.jar
selenium-chromium-driver-4.0.0-sources.jar  selenium-json-4.0.0-sources.jar
selenium-devtools-v85-4.0.0.jar             selenium-opera-driver-4.0.0.jar
selenium-devtools-v85-4.0.0-sources.jar     selenium-opera-driver-4.0.0-sources.jar
selenium-devtools-v93-4.0.0.jar             selenium-remote-driver-4.0.0.jar
selenium-devtools-v93-4.0.0-sources.jar     selenium-remote-driver-4.0.0-sources.jar
selenium-devtools-v94-4.0.0.jar             selenium-safari-driver-4.0.0.jar
selenium-devtools-v94-4.0.0-sources.jar     selenium-safari-driver-4.0.0-sources.jar
selenium-devtools-v95-4.0.0.jar             selenium-support-4.0.0.jar
selenium-devtools-v95-4.0.0-sources.jar     selenium-support-4.0.0-sources.jar
selenium-edge-driver-4.0.0.jar
```

Add the library to the project

- Made a new empty project
- made a new module
- Went to File >Project Structure >Modules >Dependencies, pushed +
- Selected the .Jar files in main selenium d/l folder, and in the libs folder > OK


Project Structure

Project Settings
- Project
- Modules
- Libraries
- Facets
- Artifacts

Platform Settings
- SDKs
- Global Libraries

Problems 3

TestJavaSelenium

Name: TestJavaSelenium

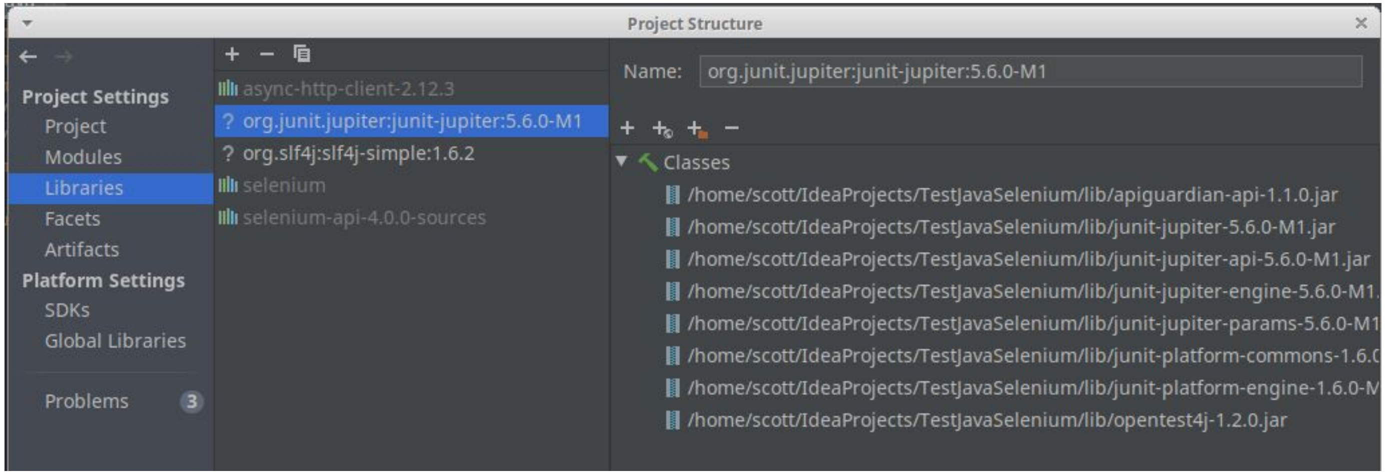
Sources | Paths | Dependencies

Module SDK: Project SDK 14 | Edit

Export		Scope
	14 (java version "14.0.2")	
	<Module source>	
	selenium-api-4.0.0.jar and 17 more files	Comp...
	async-http-client-2.12.3.jar (/opt/selenium/lib)	Compile
	async-http-client-netty-utils-2.12.3.jar (/opt/selenium/l	Compile
	auto-common-1.1.2.jar (/opt/selenium/lib)	Compile
	auto-service-1.0.jar (/opt/selenium/lib)	Compile
	auto-service-annotations-1.0.jar (/opt/selenium/lib)	Compile
	byte-buddy-1.11.19.jar (/opt/selenium/lib)	Compile
	checker-qual-3.12.0.jar (/opt/selenium/lib)	Compile
	commons-exec-1.3.jar (/opt/selenium/lib)	Compile
	error_prone_annotations-2.7.1.jar (/opt/selenium/lib)	Compile
	failureaccess-1.0.1.jar (/opt/selenium/lib)	Compile
	guava-31.0.1-jre.jar (/opt/selenium/lib)	Compile
	j2objc-annotations-1.3.jar (/opt/selenium/lib)	Compile
	jakarta.activation-1.2.2.jar (/opt/selenium/lib)	Compile
	jcommander-1.81.jar (/opt/selenium/lib)	Compile

Add any packages needed

- Needed to add some libraries using Maven (package manager)
- from the main menu, select File | Project Structure
- Under Project Settings, select Libraries, click +
- Select From Maven to download a library from Maven, e.g. org.**junit**.jupiter (5.6)
- (push magnifying glass, then down chevron in select box)
- - also added org.slf4j (**slf4j**-simple 1.6)



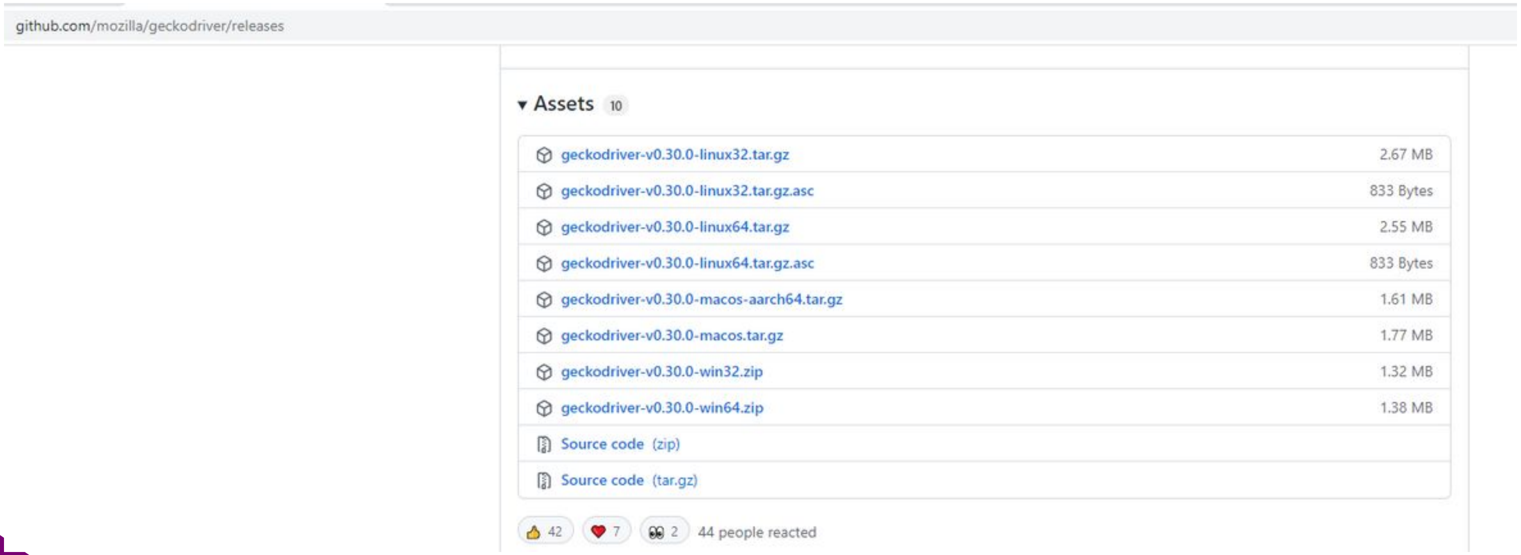
async-http-client-2.12.3
? org.junit.jupiter:junit-jupiter:5.6.0-M1
? org.slf4j:slf4j-simple:1.6.2
selenium
selenium-api-4.0.0-sources

Name: org.junit.jupiter:junit-jupiter:5.6.0-M1

Classes
/home/scott/IdeaProjects/TestJavaSelenium/lib/apiguardian-api-1.1.0.jar
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-jupiter-5.6.0-M1.jar
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-jupiter-api-5.6.0-M1.jar
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-jupiter-engine-5.6.0-M1.
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-jupiter-params-5.6.0-M1
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-platform-commons-1.6.0
/home/scott/IdeaProjects/TestJavaSelenium/lib/junit-platform-engine-1.6.0-M
/home/scott/IdeaProjects/TestJavaSelenium/lib/opentest4j-1.2.0.jar

Add the driver library

- Downloaded geckodriver (for Firefox) - went here
- https://github.com/mozilla/geckodriver/releases
- Downloaded version 30
- Unzipped the .tar.gz file & put it in /opt/selenium/geckodriver
- set permissions



github.com/mozilla/geckodriver/releases

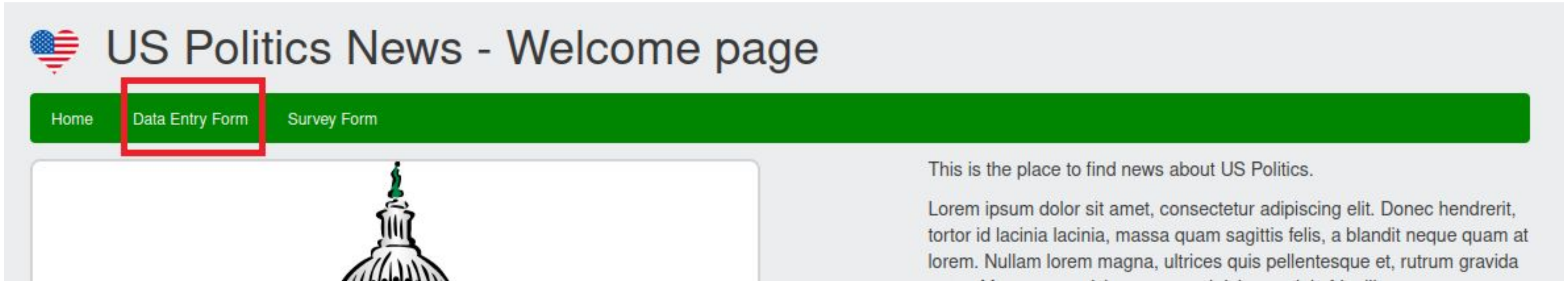
▾ Assets 10

geckodriver-v0.30.0-linux32.tar.gz	2.67 MB
geckodriver-v0.30.0-linux32.tar.gz.asc	833 Bytes
geckodriver-v0.30.0-linux64.tar.gz	2.55 MB
geckodriver-v0.30.0-linux64.tar.gz.asc	833 Bytes
geckodriver-v0.30.0-macos-aarch64.tar.gz	1.61 MB
geckodriver-v0.30.0-macos.tar.gz	1.77 MB
geckodriver-v0.30.0-win32.zip	1.32 MB
geckodriver-v0.30.0-win64.zip	1.38 MB
Source code (zip)	
Source code (tar.gz)	

42 7 2 44 people reacted

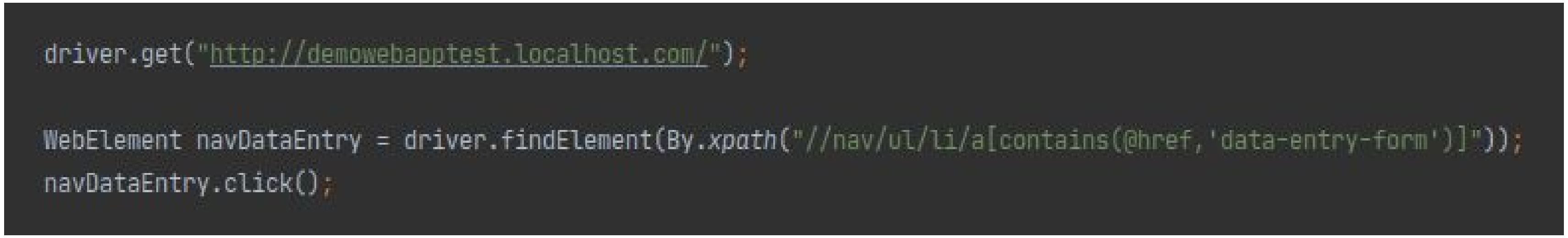
A Selenium event

- Demowebapptest.localhost.com



This is the place to find news about US Politics.

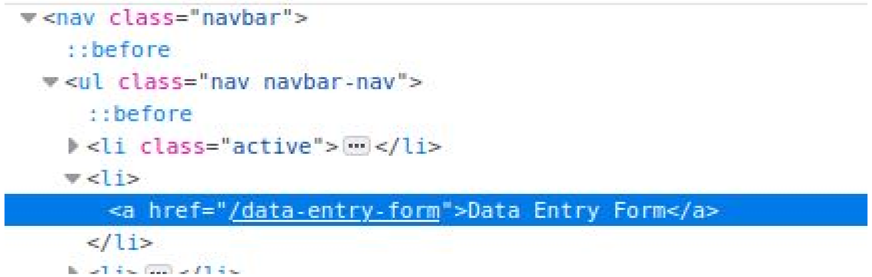
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec hendrerit, tortor id lacinia lacinia, massa quam sagittis felis, a blandit neque quam at lorem. Nullam lorem magna, ultrices quis pellentesque et, rutrum gravida



```
driver.get("http://demowebapptest.localhost.com/");

WebElement navDataEntry = driver.findElement(By.xpath("//nav/ul/li/a[contains(@href,'data-entry-form')]"));
navDataEntry.click();
```

How to locate our element

- HTML structure / DOM (XML dialect)



```
▼<nav class="navbar">
   ::before
  ▼<ul class="nav navbar-nav">
     ::before
    ▶<li class="active">…</li>
    ▼<li>
       <a href="/data-entry-form">Data Entry Form</a>
     </li>
```

- Recommend using **XPath**: a query language for selecting nodes from an XML document
  e.g. path expression for our element:
  //nav/ul/li/a[@href=’data-entry-form’]

  (You can use CSS selectors but not as precise or efficient)

XPath Basic Outline

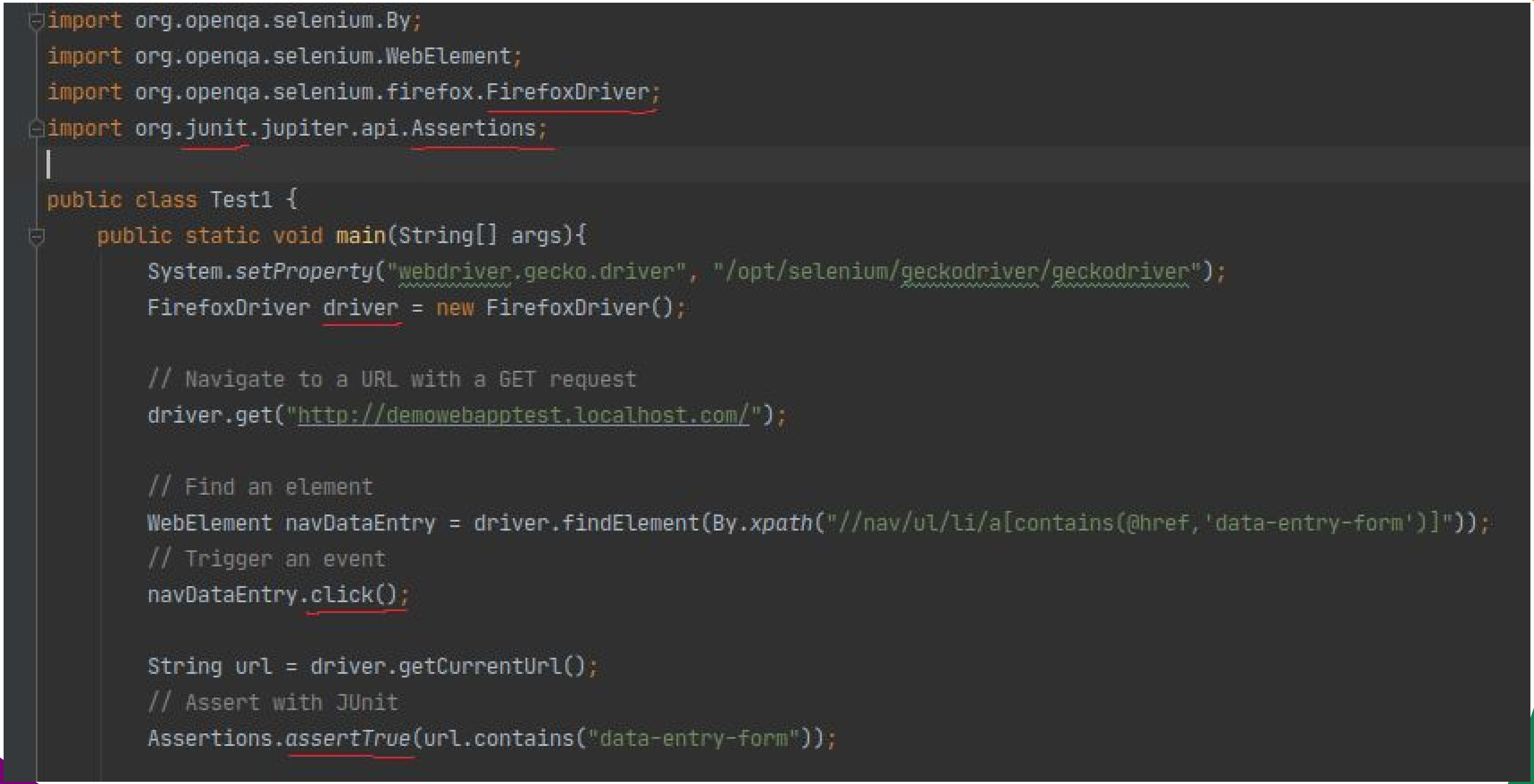
Expression	**Description**
/	Selects from the root node
//	Selects nodes in the document that match, no matter where they are
..	Selects the parent of the current node
@	Selects attributes
*	Matches any element node

e.g.

(absolute) /html/body/div/nav/ul/li/a[@href=’/data-entry-form’]

(relative) //nav/ul/li/a[**contains**(@href=’data-entry-form’)]

Set up driver, trigger event, assert

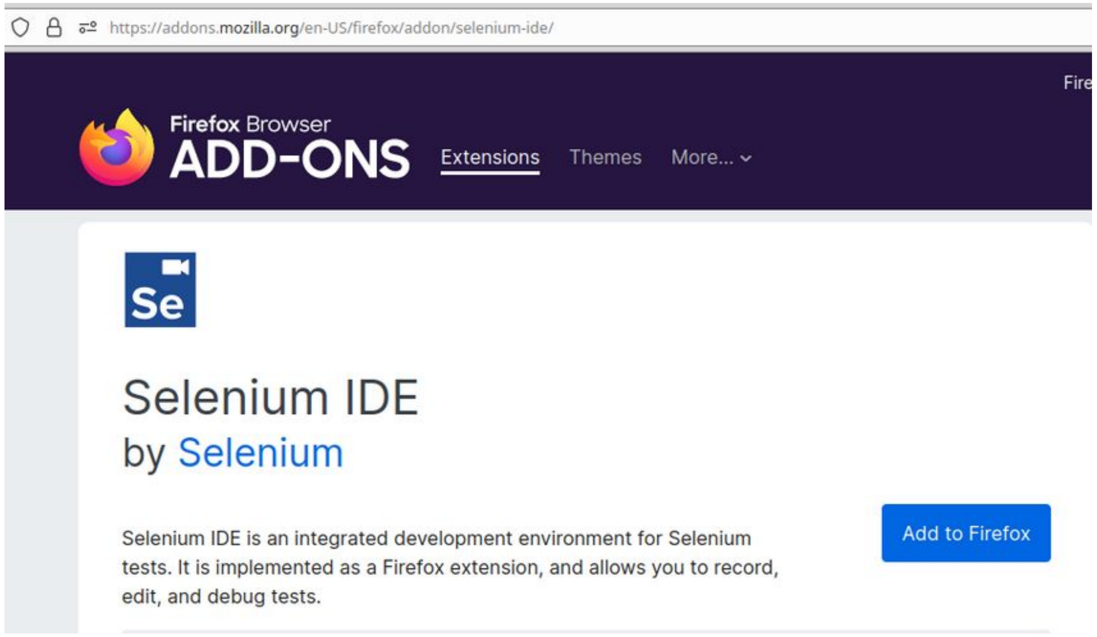


```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.junit.jupiter.api.Assertions;

public class Test1 {
    public static void main(String[] args){
        System.setProperty("webdriver.gecko.driver", "/opt/selenium/geckodriver/geckodriver");
        FirefoxDriver driver = new FirefoxDriver();

        // Navigate to a URL with a GET request
        driver.get("http://demowebapptest.localhost.com/");

        // Find an element
        WebElement navDataEntry = driver.findElement(By.xpath("//nav/ul/li/a[contains(@href,'data-entry-form')]"));
        // Trigger an event
        navDataEntry.click();

        String url = driver.getCurrentUrl();
        // Assert with JUnit
        Assertions.assertTrue(url.contains("data-entry-form"));
```

- *Show basic Intellij Java project*
- *Run it again with isSlowDemo = false*
- *Deliberately throw an error*

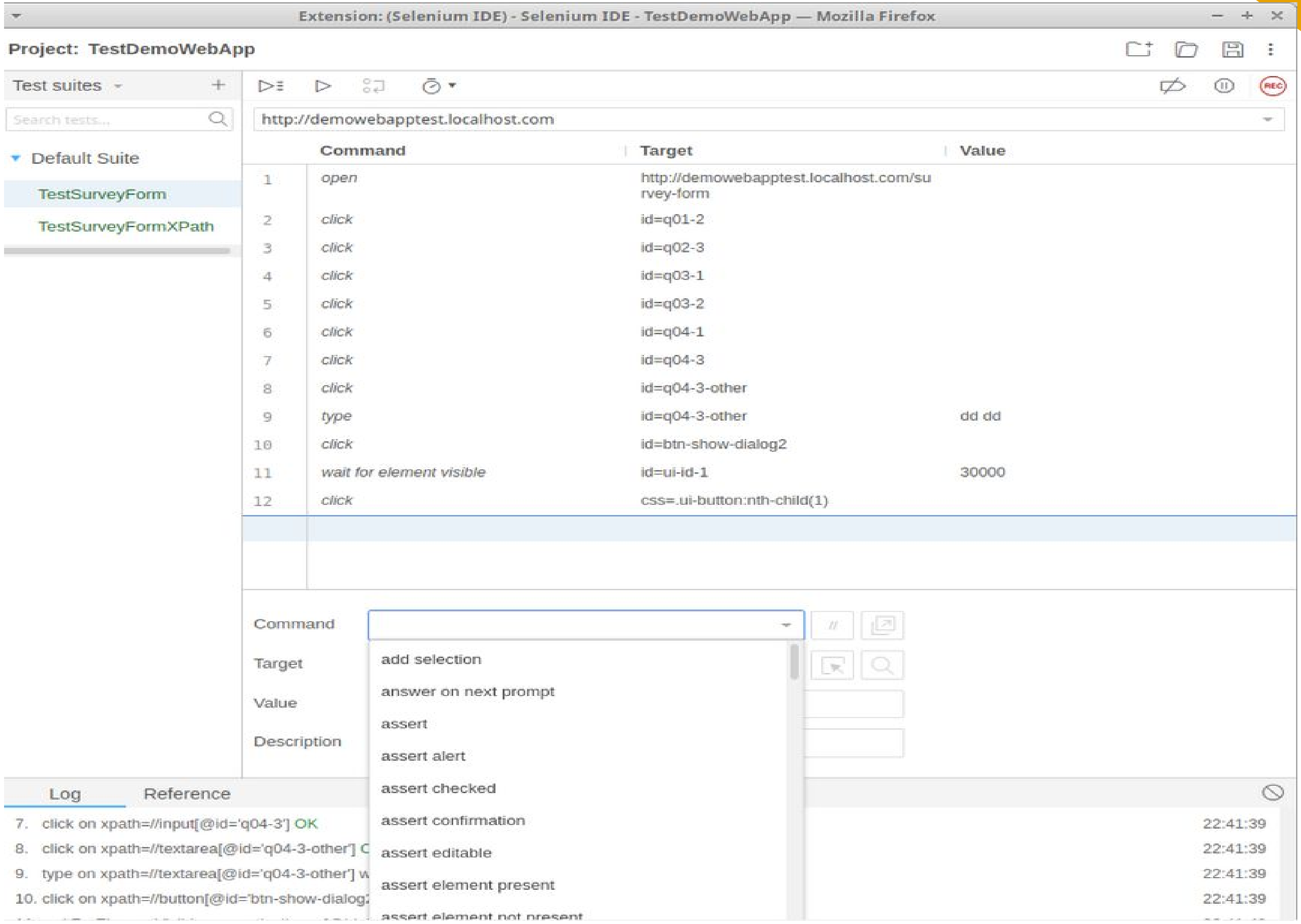
Selenium IDE

- A GUI playback tool that is in fact full-featured
- Usually installed as a browser extension / add-on
- Uses projects, tests and test suites



Selenium IDE
by Selenium

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.



	Command	Target	Value
1	*open*	http://demowebapptest.localhost.com/survey-form	
2	*click*	id=q01-2	
3	*click*	id=q02-3	
4	*click*	id=q03-1	
5	*click*	id=q03-2	
6	*click*	id=q04-1	
7	*click*	id=q04-3	
8	*click*	id=q04-3-other	
9	*type*	id=q04-3-other	dd dd
10	*click*	id=btn-show-dialog2	
11	*wait for element visible*	id=ui-id-1	30000
12	*click*	css=.ui-button:nth-child(1)	

```
7.  click on xpath=//input[@id='q04-3'] OK                    22:41:39
8.  click on xpath=//textarea[@id='q04-3-other'] C            22:41:39
9.  type on xpath=//textarea[@id='q04-3-other'] w             22:41:39
10. click on xpath=//button[@id='btn-show-dialog2             22:41:39
```

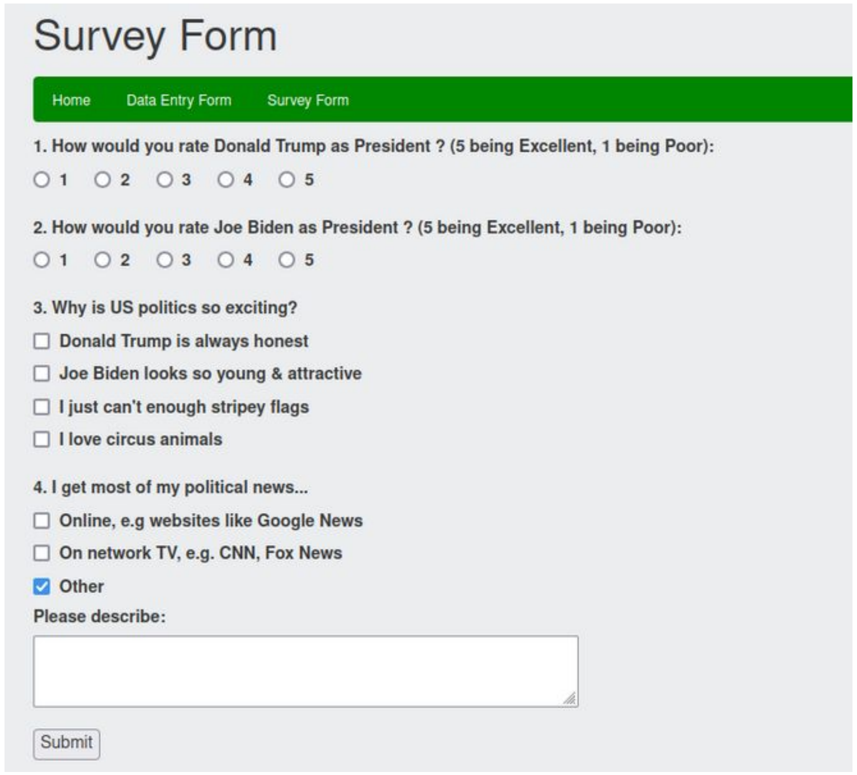
IDE features

- Allows tests to be **embedded** in other tests
- Allows you to run small **scripts**, i.e. Javascript
- Provides **assert** statements (and others that can be used like asserts)
- Provides **loops and conditions** functionality
- **Debugging** and stepping through is available
- Stores its output as **“.side” files** in JSON format
- The .side files can be manually edited
- The .side files can be manually executed on the CLI by a side-runner
- Similar to Webdriver, but has a lot of important **differences**
- Good for anyone who doesn’t like to deal with lots of code

Try out - webappdemo

Try the Survey Form

- record some events
- Use “Select target in page” button
- Add an assert
- Rework again with xpath selectors



Survey Form

1. How would you rate Donald Trump as President ? (5 being Excellent, 1 being Poor):

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

2. How would you rate Joe Biden as President ? (5 being Excellent, 1 being Poor):

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

3. Why is US politics so exciting?

- [ ] Donald Trump is always honest
- [ ] Joe Biden looks so young & attractive
- [ ] I just can't enough stripey flags
- [ ] I love circus animals

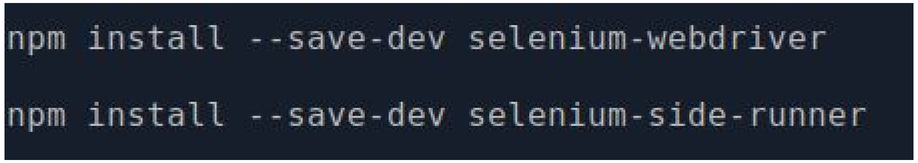
4. I get most of my political news...

- [ ] Online, e.g websites like Google News
- [ ] On network TV, e.g. CNN, Fox News
- [x] Other

Please describe:

Running a .side file with NodeJs

- Install nodejs
- Install npm (package manager)
- (install a webdriver if you haven’t already, e.g. geckodriver and add it to the PATH)
- Make a new project directory
- Install packages for selenium in the project



```
npm install --save-dev selenium-webdriver

npm install --save-dev selenium-side-runner
```

- Find the path to the side-runner index.js file, e.g. node_modules/selenium-side-runner/dist/index.js
- Try a command on the CLI like:
- node [path-to-side-runner] -c “browserName=’firefox’” [path-to-side-file]

“Gotchas”

- A lot of **documentation** online for Webdriver and the IDE is out of date
- Some of the official Selenium documentation for the IDE is wrong
- You need to match the browser version with the Webdriver version, **especially** for Firefox
- Don’t go for specific matches on page elements. Use **contains()** to be futureproof
- Sometimes **wait** or pause statements are necessary for testing on slow servers (unavoidable)
- **XPath patterns** that work for one **browser** may not work in another
- XPath patterns that work in the **IDE** may not work in the side-runner
- Sometimes Selenium just gets confused, e.g. for the iframe or window to look for elements in
- Sometimes in the IDE asserts have to be done in a Javascript snippet

- Show videos – “bad” and “good”