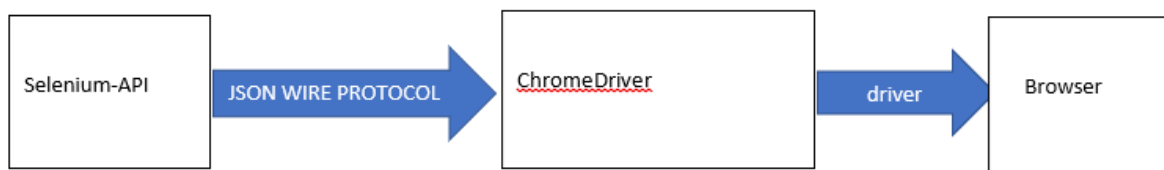


SELENIUM-4 ENHANCEMENTS

Until Selenium -3 era, JSON wire protocol is being used to talk to browser, this protocol helps for seamless communication between the selenium and browser even when the selenium script is written in any programming language (Java, c#, python, perl, ruby) as it supports. JSON wire protocol no longer be used with selenium 4.x(when its fully developed). Selenium 4.x will use “W3C/Http wire protocol” to talk to browser, however this is not supported by all the browsers yet (Firefox to give total support for now). Selenium 4.x will continue to use JSON wire protocol until the HTTP protocol is totally implemented and all browser vendors are ready to support.

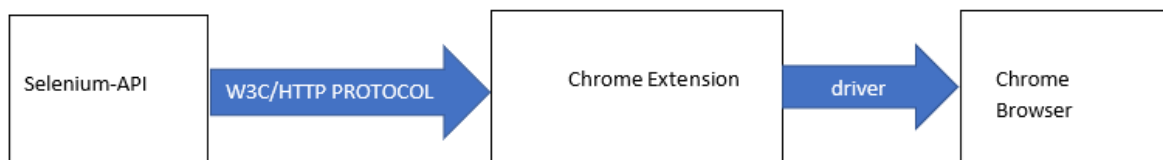
3.x architecture:



The new protocol brings in some enhancements in executing /sending the browser commands

Selenium 4.x is going to have a browser extension as an amidst layer before the browser window. This extension is liable for sending /receiving the responses from and to between the browser and selenium -API

4.x (alpha) architecture: -



There is a new class file written which is DevTools which implements the interface closable.

The class helps dealing with the page developer tools (console/network/security/DOM) in a better way than it was ever in the past.

Here below, some key features: -

- **Attaching the new driver object to an already opened window**

This is one key enhancement I should say; the main purpose of this feature is during debug process. The features claim to be important though it's not much used in conventional execution.

Eg: This is used mostly in failure cases where a script fails somewhere in the middle of 10th page and it's a great deal of difficulty to debug right from the 1st page, in this case we just attach our new driver object to 10th page as it becomes easy to debug from there.

How to Do it: - Well, all we need to do is get the "debugger-address" of the page from the DevTools class method and parse it to the new driver object before we start navigating to the URL. The below code is getting the chromeBrowser's capability, getting the debugger address then setting the experimental option. Since the debugger address changes everytime we create a new session it's a good practise to write that value into a properties file dynamically and read the same.

```
FileInputStream fis=new FileInputStream("./resources.properties");
```

```
Properties prop=new Properties();
```

```
Prop.load(fis);
```

```
Object cap=Driver.getCapabilities().getCapability("goog:chromeOptions");
```

```
Map<String, String> chromeOptions= (Map<String, String>)cap;
```

```
String address=ChromeOptions.get("debugger:Address");
```

```
Prop.setProperty("debugger",address);
```

```
ChromeOptions.setExperimentalOptions("debugger Address", prop.getProperty("debugger"));
```

- **Minimizing the window: -**

So far in selenium we have the option to maximize the browser but Se-4.x helps us minimizing as well, the meaning of minimize is pretty clear that we are just hiding the tab, not resizing it.

How to do it:-

```
RemoteWebDriver driver=new ChromeDriver();
```

```
Driver.manage().window().minimize();
```

- **Switching to a new tab: -**

With the use of Selenium 3.14 we can only switch to a new window, the control automatically transfers to the new window that's opened. But for the first time in selenium 4.x we have the ability to switch to a tab within the same window.

How to do it:- The choice (tab/window) is facilitated through an enum and we will be switched accordingly .

//command to switch to a new tab

```
Driver.switchTo().newWindow(WindowType.TAB);
```

//command to switch to a new window

```
Driver.switchTo().newWindow(WindowType.WINDOW);
```

- **Taking DOM/Fullsize screenshots with the help of Developer tools**

<https://developers.google.com/web/tools/chrome-devtools/>

As much as I know,

1. There are changes to the GRID, but not so stable and recommended to use.
2. In near future we may have some extra functions for xpath such as (nearby, around. Etc)

Note: All the functionalities those were documented above have not been implemented fully by all browser vendors. I think Firefox is bit ahead of rest other browsers.